TOWARDS NEXT GENERATION WLANS: EXPLOITING COORDINATION AND COOPERATION

BY

ZHENG ZENG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Professor P. R. Kumar, Chair & Director of Research
Professor Klara Nahrstedt
Professor Yih-Chun Hu
Professor Matthew Caesar
Professor Romit Roy Choudhury, Duke University

Abstract

Wireless Local Area Networks (WLANs) operating in the industrial, scientific and medical (ISM) radio bands have gained great popularity and increasing usage over the past few years. The corresponding MAC/PHY specification, the IEEE 802.11 standard, has also evolved to adapt to such development. However, as the number of WLAN mobile users increases, and as their needs evolve in the face of new applications, there is an ongoing need for the further evolution of the IEEE 802.11 standard. In this thesis we propose several MAC/PHY layer protocols and schemes that will provide more system throughput, lower packet delivery delay and lessen the power consumption of mobile devices. Our work investigates three approaches that lead to improved WLAN performance: 1) cross-layer design of the PHY and MAC layers for larger system throughput, 2) exploring the use of implicit coordination among clients to increase the efficiency of random media access, and 3) improved packets dispatching by the access points (APs) to preserve the battery of mobile devices. Each proposed solution is supported by theoretical proofs and extensively studied by simulations or experiments on testbeds.

To Father and Mother.

Acknowledgments

Foremost, I am sincerely thankful to my thesis advisor, Professor P. R. Kumar, for his support and guidance on my Ph.D. study and thesis. I have learnt a lot from him in many ways. Besides the professional knowledge, it is his dedication to work and rigorous attitude toward research that I look up to and will try my best to achieve in my future career. This thesis would not have been possible without him.

My great gratitude goes to Professor Matthew Caesar, Professor Romit Roy Choudhury, Professor Yih-Chun Hu and Professor Klara Nahrstedt, for serving on my doctoral committee and for their insightful comments on this thesis.

I want to express my special thank to my former advisor, late Professor Jennifer C. Hou. With her passion and immense knowledge, she had led me to the area of wireless communication in my early Ph.D. years, and enhanced my interests in research. I will remember her for my life.

I would like to extend my gratitude to the fellow labmates in the UIUC System and Networking group, especially Yan Gao, I-Hong Hou, Lu-chuan Kung, Vinod M Prabhakaran, Lu Su and Yong Yang, for their great help in our collaboration. Also, I would like to thank my coauthors, including but not limited to Professor Carl Gunter, Professor Xue Liu, Professor Lui Sha, Professor Qixin Wang, Professor Rong Zheng, Dr. Ying Huang, Dr. Kun Tan, Dr. Wook Shin, Yufei Wang and Guanbo Zheng.

I thank NSF for supporting my attendance to the prestigious conference SIGCOMM and ICDCS, which have inspired me in many aspects of my research. I also thank the Q-award scholarship by Qualcomm for supporting my doctoral study.

Lastly, and most importantly, I wish to thank my parents, Ping Zeng and Jian Liu, for their love and support. To them I dedicate this thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation and Research Challenges

WLANs have become very popular and have been widely deployed in recent years, especially enterprise WLANs consisting of a set of APs that provide wireless service to client devices over a relatively large area, such as a business building or a university campus. Large wireless network projects are being put up in many major cities: New York City, for instance, has begun a pilot program to cover all five boroughs of the city with wireless Internet access. Usually WLANs serve as the last hop of the internet connection, and 802.11 Distributed Coordination Function (DCF) is the most popular, if not the only, MAC protocol used in WLANs nowadays.

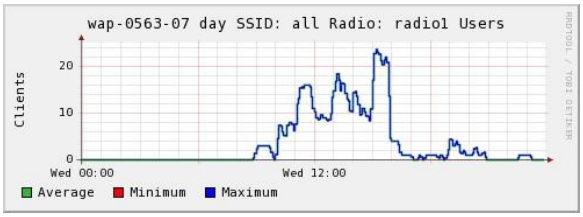
Looking back at the evolution of WLANs during the past fifteen years, there are three trends that are worth noticing. The first one is the significant increase of link data rate. While in 1997 the date rate was only 1Mbps, it increased to 11Mbps in 1999, 54Mbps in 2003, and 170mbps in 2009 with MIMO technology. The second trend is greater congestion along with greater usage. One can consider the campus WLAN of the University of Illinois at Urbana-Champaign for example. Figure 1.1 shows the number of client hosts associated with four APs over 24 hours. AP1 and AP2 are spatially close enough that most clients of the two APs are within each other's transmission range. So are AP3 and AP4. It can be seen that during peak hours, each AP serves over 20 clients, and there are more than 40 clients within the same transmission range. Last but not lease, nowadays users ask for more from WLANs. In addition to larger bandwidth, users also desire quality of service (QoS), less power consumption of their mobile devices, security, and a larger environment where they have wireless access. Facing such situations and challenges, the 802.11 standard has also evolved to incorporate more functionalities, such as 802.11e for QoS, 802.11i for security, and 802.11p which standardized the wireless access for the vehicular environment.

However, despite these efforts, the current MAC and PHY specification in the IEEE 802.11 standard still has both a need for as well as room for improvement to better serve the increasing number of users, providing more system throughput or lower packet delivery delay, and at lower energy consumption. In this thesis, we

AP1 in the University Dormitory

AP2 in the University Dormitory

AP3 in the Dept. of Computer Science Building



AP4 in the Dept. of Computer Science Building

Figure 1.1: The number of wireless clients associated to one access point

identify three possible approaches that can be used to improve WLAN performance: cross-layer design of the PHY and MAC layers, mechanisms for implicit coordination among clients, and packets dispatching on the access points (APs).

1.1.1 MAC-PHY Cross-layer Design

In the 802.11 Standard, MAC and PHY are two separate layers that share as little information as possible between them. This is because the Open Systems Interconnection Basic Reference Model (OSI model) [70], which is used to guide the design of most network communication protocols, is a strictly layered architecture with each layer providing service to its upper layer but hiding as many details as possible to maintain isolation between layers. In the 802.11 specifications, the MAC layer deals with addressing and multiplexing on multi-access media, while the physical layer handles only the issue of how to transmit/receive over the medium. Current physical layer receivers only provide a packet to the MAC layer if it is successfully received, or discard it otherwise. Very little additional information is passed by the physical layer to the upper layers in commodity wireless cards.

Besides existing efforts, there is one additional MAC-PHY cross-layer design topic that needs further study: How can we help the MAC layer make more intelligent decisions when encountering corruptions, by identifying the *cause* of packet corruption, and exporting it to upper layers. Wireless communication is inherently vulnerable in nature and packets can be corrupted due to various reasons. The MAC layer cannot diagnose the reason for the loss of packets when the PHY layer does not provide it any information, as happens when the IEEE 802.11 specification is strictly followed. Hence it handles all packet losses in the

same way. However, the best reaction that the MAC layer should take is different for different causes of packet corruption. For example, if the cause is a weak signal, then the sender need not backoff but could use a lower data rate in the retransmission. On the other hand, if the cause of packet corruption is interference from transmitters within the same wireless system, then the sender could backoff but retransmit at the same rate; Finally, if it is interference from other systems, say, ultra-wide-band devices, whose spectrum overlaps with the transmitter-receiver pair, but the interference is only over a few sub-carriers, then the retransmission would better be done without backoff, and at the same rate, but avoiding using those interfered sub-carriers. It can be seen, therefore, that by allowing the link to respond to packet corruption in differentiated ways, each individually optimized, and therefore always operates at the "best" setting in different situations, the network system throughput can potentially be improved.

1.1.2 Implicit Coordination among Clients

The 802.11 DCF is a completely distributed design, with no coordination, even among APs, let alone clients. In the recent past, however, enterprise WLANs have made a dramatic shift towards centralized architectures. In these architectures, a central control element observes the entire network, centrally configures the parameters of each AP, and centrally assigns each client to its AP. Studies have shown that in many scenarios such central scheduling can achieve higher performance than 802.11 DCF by significantly reducing collisions and making better use of the bandwidth. In particular, there have been many efforts in both academia [54, 62] and industry [4] to schedule downlink traffic (traffic from the AP to clients) to outperform DCF under certain circumstances.

However, despite all the efforts aimed at enhancing coordination and centralized control among APs, there has not been much work in exploring the possibility of cooperation *between* clients, mostly due to practical concerns. Coordination between clients is usually achieved by centralized scheduling, which may suffer both from the likely huge communication overhead as well as vulnerability to missing control messages, especially for uplink traffic. We therefore address the following question: Is it possible to let clients loosely cooperate, without necessitating any extra control overhead merely by making use of the broadcast nature of the wireless medium and clients overhearing each other?

1.1.3 Packet Dispatching on Access Points

There is increasing use of battery-powered portable devices, such as smart phones and PDAs, to access the Internet through Wi-Fi, and reducing Wi-Fi power consumption has become crucial for extending the battery life of portable devices. The IEEE 802.11 Power Saving Mode (PSM) has been proposed to reduce

such power drain. Unlike the original Wi-Fi Constantly Awake Mode (CAM) which draws power over 1W all the time, a mobile device running PSM can go to sleep that incurs power consumption of only around 50mW [43]. However, when a PSM device is sleeping, it cannot transmit or receive any packets; hence, PSM clients conserve energy at the cost of larger packet delivery delay.

As pointed out by several researchers [50][23][8], the power consumption of a PSM client depends not only on the traffic load destined for it, but also on the traffic loads destined for other clients. The reason is that when the AP serves more than one client or there is background traffic, the packet transmission from the AP to the PSM client may be susceptible to interruption by data transmission from other hosts or from the AP to other hosts. This results in the prolongation of the time for the PSM client to receive the buffered data client, and thus more power consumption.

Motivated by this observation, we ask the following question: *What is the optimal strategy to deliver the downlink packets so that the PSM clients waste the least time staying awake and overhearing downlink traffic being transmitted to other clients?* Clearly, the absolutely sleep-optimal solution is for the AP to wake up one client, send all and only its packets to it, put it back to sleep, then wake up the next client, and repeat the process. However, this solution is not practical as the AP cannot wake up a client anytime it wants to, since the client is not capable of receiving any information from the AP while it is sleeping.

Other possible improvement also exists in the current 802.11 MAC/PHY specification. However, we will focus on the above ones in this thesis. As a matter of fact, the above challenges are motivated by the real needs originating from the campus WLANs in the University of Illinois at Urbana-Champaign.

1.2 Contribution

In this thesis, we present a suite of our work to address the above challenges. Specifically, our contributions are listed as follows.

Packet Corruption Classifier (PCC): We design a classifier called *Packet Corruption Classifier (PCC)*, which immediately identifies the cause of a packet corruption, by analyzing the pattern of certain physical layer information in the OFDM system. The analysis can be done in parallel with the normal packet de-modulation process at the receiver side. We study and implement three algorithms in *PCC* to separate the causes of corruption, and compare their accuracy based on experiments conducted in an electromagnetic anechoic chamber using a GNU radio test bed. We presents the details in Chapter 2.

Modeling the Effect of Carrier Sensing and Back-off on Network Capacity: In order to better understand the efficiency of the802.11 DCF protocol, we propose an analytical model that extends Calí's

model [15] to multi-hop wireless networks and incorporates the effects of physical carrier sensing, SINR, and collision caused by cumulative interference. In our proposed model, we follow Calí's methodology of characterizing the time interval between two consecutive successful transmissions as the *virtual transmission time*, t_v, and derive its expected value. We faithfully incorporate all the aforementioned effects that the carrier sense threshold CS_{th} makes in multi-hop wireless networks, and derive $E(t_v)$ *as perceived by nodes in one interference range*. We validate the derived model via simulation, and obtain several important implications from the analytical model.

CHAIN: Inspired by the above modeling and analysis, we propose CHAIN, a Coordinated Heavy-traffic efficient Access scheme that is derived from 802.11 DCF and which can co-exist with it. CHAIN is designed mainly to improves the efficiency of uplink traffic. By improving the uplink efficiency, it also indirectly improves the down-link throughput. CHAIN significantly improves the uplink channel access efficiency without increasing the channel access delay, under any traffic pattern. Moreover, it introduces negligible control and coordination overhead. CHAIN is also robust, yet simple and practical. CHAIN adapts its behavior automatically to the dynamic traffic load. When the network is idle, it operates in a way similar to DCF. But when the network becomes congested, the clients automatically adapt their behavior to reduce the contention, without explicit control messages from AP or a scheduler. As a consequence, the network throughput is significantly improved.

SOFA: Targeting at preserving the battery of mobile devices, we propose SOFA, a downlink traffic scheduler for the AP that achieves *system sleep-optimality*, and *energy fairness*, while meeting an *attention fairness constraint*, and *without any deferral of transmissions beyond the current beacon period.*. The terminologies in italics will be formally introduced and explained in Chapter 5. Further, SOFA is analytically proved to be stable.

In each of the next four chapters, we cover one of the above topics.

Chapter 2

Packet Corruption Classifier

2.1 Introduction

The Open Systems Interconnection Basic Reference Model (OSI model) [70], which is used to guide the design of most network communication protocols, is a strictly layered architecture with each layer providing service to its upper layer but hiding as many details as possible to maintain isolation between layers. In wireless networks, for example, the MAC layer deals with addressing and multiplexing on multi-access media, while the physical layer handles only the issue of how to transmit/receive over the medium, i.e., how to communicate with a single device. Current physical layer receivers only provide a packet to the MAC layer if it is successfully received, or discard it otherwise. There is very little additional information provided by the physical layer to the upper layers in commodity wireless cards. Lately, however, the wireless networking community has evinced increasing interest in exporting information gathered by the physical layer to solve certain MAC level problems. Most current work is aimed at answering the question: Is it possible to obtain and export physical layer information from a packet that is *not* successfully received? If so, how do we do so?

This chapter is addressed at answering to what extent available physical layer information can be exploited. If a packet is not received correctly, there are two most common reasons. The first cause is due to a weak received signal. This happens when the signal strength of the sender at the receiver side is not strong enough to overcome the noise caused by fading or processing circuits of the hardwire. The second cause is interference. Interference results from one or more concurrent transmissions from other devices, and it may vary in time even within the duration of a single packet transmission. Note that interference need not just come from other devices within the same wireless system, or even not spread over the same frequency spectrum. Here by "interference" we will refer to transmissions over the same spectral band, unless specifically suggested otherwise. Separating this "same-band" interference from a weak signal is our primary interest. We will also study and discuss "overlapping-band" interference in Section 2.6 to complete our work. The specific question we want to address is: *By only analyzing certain available physical layer information in re-*

ceived data, can we separate the cause of packet corruption between weak signal and interference? Moreover, can we do so in a way that introduces no communication overhead and minimal computational overhead? The answer to this question would depend on the specific physical layer design, while in this chapter we study the OFDM system.

Being able to identify the cause of packet corruption and exporting it to upper layers is of significant help in many ways. We identify two of them. First, the MAC layer can make more intelligent decisions when encountering corruptions. The best reaction that the MAC layer should take to different kinds of corruption differs. For example, if the cause is a weak signal, then the sender could use a lower data rate in the retransmission without backoff; on the other hand, if the cause of packet corruption is interference from transmitters within the same wireless system, then the sender could backoff but retransmit at the same rate; Finally, if it is interference from other systems (say, ultra-wide-band devices) whose spectrum overlaps with the transmitter-receiver pair, but only interferes over a few sub-carriers, then the retransmission would better be done without backoff, at the same rate, but avoiding using those interfered sub-carriers. Therefore, by allowing the link to always respond to packet corruption in the best way, and always operate at the "best" setting in different situations, the network performance can potentially be improved.

A second major benefit of such packet classification is that it can be used to provide a more accurate time-varying network topology/conflict graph on the fly. Scheduling problems in wireless network [12][19] have been studied for a long time. Many of the solutions are based on a *given* conflict graph which defines the relation between any two links, whether they interfere with each other or not, and in practice such conflict graphs are usually gained through one-time measurement only, as suggested in [46]. However, the conflict graph may be time-varying since the channel condition between nodes is time-varying. It is possible that two links that do not interfere in the morning do interfere with each other in the afternoon in a medical environment (which is one of the motivations for our study). Therefore a static conflict graph does not suffice. However, with proper coordination, feedback and diagnosis on when and which packets get corrupted due to interference, we can modify the interference graph in real-time, while introducing no overhead traffic when the conflict graph remains static. This can potentially result in better performance.

My key contribution is the design of a classifier called *Packet Corruption Classifier (PCC)*, which immediately identifies the cause of a packet corruption, by analyzing the pattern of certain physical layer information in OFDM system. The analysis can be done in parallel with the normal packet de-modulation process at the receiver side. We study and implement three algorithms in *PCC* to separate the causes of corruption, and compare their accuracy based on experiments held in an electromagnetic anechoic chamber using a GNU radio test bed.

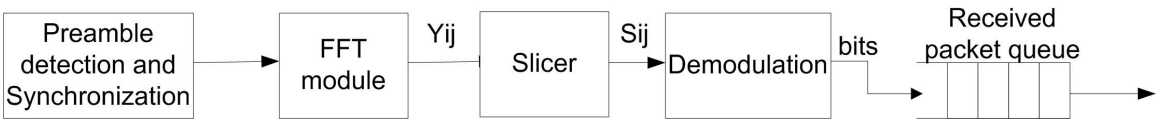


Figure 2.1: Block diagram of an OFDM receiver

The rest of this chapter is organized as follows. After briefly introducing OFDM in Section 2.2, we formulate the corruption diagnosis problem as a change point problem, and present *PCC* in Section 2.3. Section 2.4 describes the set up of the experiment. The performance of *PCC* is evaluated in Section 2.5, followed by a discussion on how to choose the proper classification algorithm. Section 2.6 expands *PCC* to identify "overlapping-band" interference. In Section 2.7 we discuss the related work, and finally conclude in Section 2.8.

2.2 Background Material

In this section, we first present a brief description of the OFDM physical layer, and define notation that will be used throughout the chapter. Then we identify and explain the key observation that leads to the design of PCC.

2.2.1 An Overview of OFDM Physical Layer

OFDM, short for *Orthogonal Frequency Division Multiplexing*[17], is an FDM modulation technique for transmitting large amounts of digital data over radio. OFDM works by splitting the radio signal into multiple smaller sub-signals that are then transmitted simultaneously at different frequencies, which are defined as *sub-carriers*, to the receiver.

Figure 2.1 gives a (partial) block diagram of an OFDM receiver. The input is a sequence of sampled signals after the signal has been converted to baseband, while the output is a packet queue, or, say, a bit stream. One key element in the receiver is the *slicer*, which quantizes the sampled signal to one of a few symbols $\{\hat{a}_k\}$ in the form of complex numbers. The set $\{\hat{a}_k\}$ is determined by modulation, and usually we have $|\hat{a}_k| = 1$. In current off-the-shelf devices, the physical layer exports only the final bit stream, which results in the isolation of the upper layers from implementation details of the physical layer. However, we seek to identify additional information related the physical layer implementation that breaks this isolation, but helps estimate channel condition and interference. It turns out that such infomation can be gained from the input and output stream of the *slicer*.

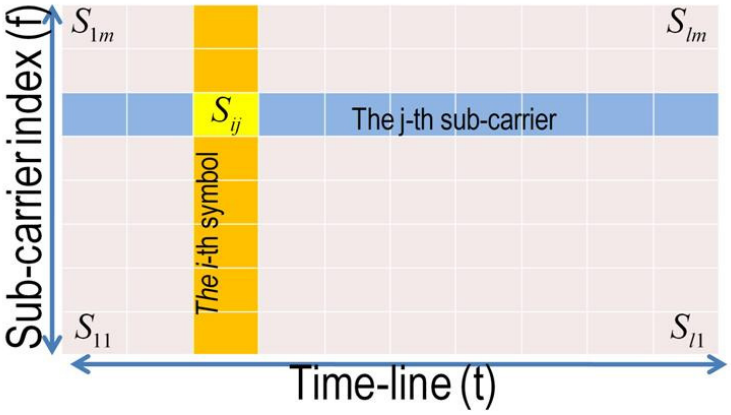


Figure 2.2: An OFDM packet composition

Figure 2.2 illustrates the composition of a packet in an OFDM receiver. The x-axis is the time-line while the y-axis is the *sub-carrier* index. We use m to denote the number of *sub-carriers*. A *symbol* refers to a collection of signals modulated in single unit of time synchronously across all m *sub-carriers*, i.e., a column in Figure 2.2. We use S_{ij} to represent the content at the j-th sub-carrier in the i-th symbol of a packet before it enters the *slicer*, and Y_{ij} after that. It follows that $S_{ij} \in R^2$ and $Y_{ij} \in \{\hat{a}_k\}$. The distance between them is defined as

$$X_{ij} := Y_{ij} - S_{ij}. \tag{2.1}$$

Note that $1/|X_{ij}|$ is a good approximation of SINR (Signal to Interference plus Noise Ratio).

2.2.2 Hint from Constellation Graph

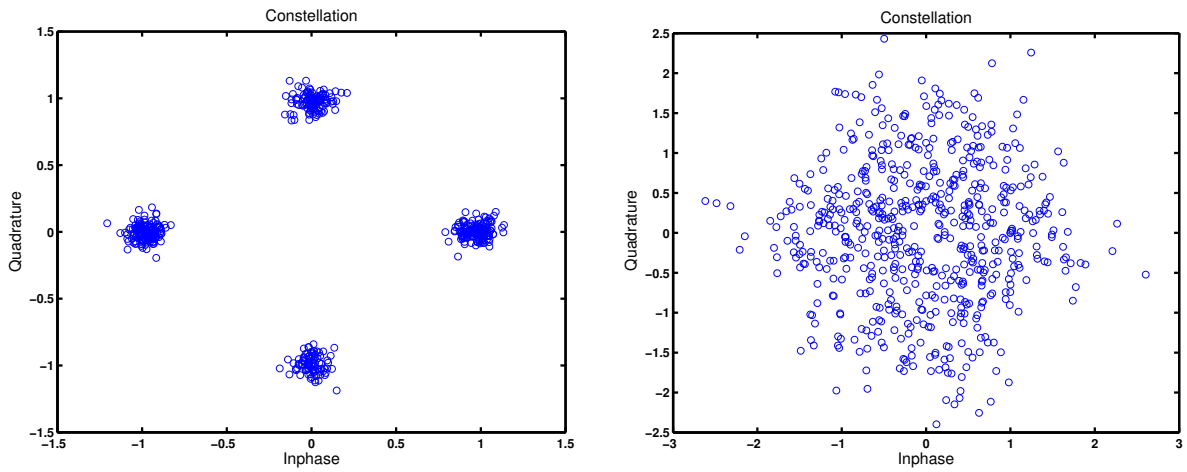
The following three assumptions guide the design of PCC.

(A1) When there is no interference, the channel condition remains invariant throughout a packet reception.

(A2) All sub-carriers share the same channel condition.

(A3) When the cause of corruption is interference, we only consider the situations when interference starts or ends somewhere during the packet reception.

(A1) implies that we don't consider fast-fading, which is reasonable since our target scenario is an indoor environment without high speed mobility. **(A2)** implies that we don't consider the corruptions caused by interference whose duration entirely *covers* the duration of packet reception. **(A2)** does not hold when de-modulation requires higher SINR than the OFDM synchronization procedure, and we are considering relaxing **(A2)** in future work.



Before Interference During Interference

Figure 2.3: Consellation graph of an interfered packet

Given the three assumptions, we hypothesize the following conjecture. If a packet corruption is caused by a weak signal, $\{X_{ij}\}$ conforms to one single distribution. On the other hand, if a packet corruption is caused by interference, then $\{X_{ij}\}$ from the non-interfered part conforms to one distribution, while $\{X_{ij}\}$ from the interfered part conforms to another. We have obtained experimental results from the GNU radio test bed that support this conjecture. Figure 2.3 shows the constellation graph of a packet that is ten symbols long transmitting on 128 sub-carriers. The interference starts in the middle of the reception. The left figure plots $\{S_{ij}|0 < i < 3\}$ while the right one plots $\{S_{ij}|7 < i < 11\}$ and the difference can be seen to be dramatic. The important conclusion is that by following this conjecture, the original problem can be formulated as a *change point problem*.

2.3 Problem Formulation and PCC Design

2.3.1 The Change Point Problem

We start with the formal statement of a general change point problem. Let $S = \{x_1, x_2, ..., x_n\}$ be a sequence of independent random variables, with probability distributions $F_1, F_2, ..., F_n$, respectively. Then the problem is to test the following null hypothesis:

$$H_0 : F_1 = F_2 = ... = F_n, \tag{2.2}$$

versus the alternative

$$H_q : F_1 = ... = F_{k_1} \neq F_{k_1+1} = ... = F_{k_q} \neq F_{k_q+1} ... = F_n, \tag{2.3}$$

where $1 < k_1 < k_2 < ... < k_q < n$, q is the unknown number of change points, and $k_1, k_2, ..., k_q$ are the respective unknown positions that have to be estimated.

For our problem, S consist of $|X_{ij}|$, where $n = m \times l$, and $x_{(i-1)\times m+j} = |X_{ij}|$. There are three hypothesis to be tested. First, if the packet is corrupted due to weak signal, H_0 defined in (2.2) holds. Second, if the packet is corrupted due to interference that either (i) starts before the packet reception and ends during the packet reception or (ii) starts during the packet reception and lasts till the end of reception, then there is only one change point and H_1 holds, with

$$H_1 : F_1 = ... = F_{k_1} \neq F_{k_1+1} = ... = F_n. \tag{2.4}$$

Last, if the packet is corrupted due to interference that both starts and ends during the packet reception, then there are two change points and H_2 holds, with

$$H_2 : F_1 = ... = F_{k_1} \neq F_{k_1+1} = ... = F_{k_2} \neq F_{k_2+1} = ... = F_n. \tag{2.5}$$

The goal is to test H_0 against H_1 plus H_2. We note that it is not needed to separate H_1 from H_2, i.e., we only need to test *whether* there exists a change point(s) or not.

2.3.2 The Packet Corruption Classifier (PCC) Algorithm

The traditional change point solution is to go through the whole sequence S and test each variable x_i to determine whether it is the change point. This requires a lot of computation. Since our goal is to only search for the existence of one change point, we have designed the following simple sliding window algorithm to reduce the computational complexity. Let SW_1 and SW_2 represent two adjacent subsequences of S, each of which has l_w variables, i.e.,

$$SW_1 = \{x_{i+1}, x_{i+2}, ..., x_{i+l_w}\}, \tag{2.6}$$

$$SW_2 = \{x_{i+l_w+1}, x_{i+l_w+2}, ..., x_{i+2\times l_w}\}. \tag{2.7}$$

Then the hypotheses become

$$H_0' \quad : \quad F_1 = ... = F_{i+2\times l_w}, \tag{2.8}$$

$$H_1' \quad : \quad F_1 = ... = F_{i+l_w} \neq F_{i+l_w+1} = ... = F_{i+2\times l_w}. \tag{2.9}$$

We move SW_1 and SW_2 throughout S with step size t, and test H_0' against H_1' for each move. If H_0' holds for every move, then we accept that the packet corruption is due to a weak signal. Otherwise, we say it is due to interference. The following pseudocode shows the details with return value true if it is interference, and false otherwise.

procedure $PCC(t, l_w)$
Interference=False
$SW_1 = x_{i+1}, x_{i+2}, ..., x_{i+l_w}$
$SW_2 = x_{i+l_w+1}, x_{i+l_w+2}, ..., x_{i+2\times l_w}$
while SW_2 **stays in** S **do**
  test H_0' **against** H_1'
  if H_1' **holds then**
    Interference=True
    Jump out of the loop
  end if
  Move SW_1 **and** SW_2 t **variables toward the end of** S
end while
Return Interference

Note that signals in one symbol are modulated in a single unit of time synchronously, so that we can assume the variables in one symbol conform to one distribution. Therefore the change point can only be located between symbols, i.e., between $X_{i,m}$ and $X_{i+1,1}$ for some i, so that it is safe to set the step size t to be m instead of 1, and set the window size l_w to be m or a multiple of m.

2.3.3 Evaluation Metrics

In the next section we discuss the tests that will be used to separate H_0' from H_1'. Before that, it is necessary to first define the metrics that evaluate the test algorithms. If our goal is to detect interfered packets from among all corrupted ones, then there are two metrics:

False positive rate: This is the percentage of corrupted packets which the test identifies as interfered,

from among all the packets whose real corruption reason is weak signal.

Miss Rate: This is the percentage of corrupted packets which the test identifies as weak signal, from among all the packets whose real corruption reason is interference.

2.3.4 Hypothesis Tests

We apply and study three Hypothesis tests: (1) Change of Mean Test (CMT); (2) Chi-square Test of Independence (CST); and (3) Mann-Whitney-Wilcoxon (MWW). Although using different techniques, they share the same structure. The tests output an indicator result $I \in R$ for each sliding window operation, and compare it to a pre-determined threshold I^{th}. If $I \leq I^{th}$, then H_0' is accepted; otherwise H_1' is accepted. We describe the three tests by defining their indicators I_{cmt}, I_{cst}, I_{mww}, respectively.

Change of Mean Test (CMT)

In the following discussion, F_i refers to the distribution of X_{ij} with interference which has mean μ_i and variance σ_i^2; while F_n refers to the distribution of X_{ij} without interference with mean μ_n and variance σ_n^2. Similarly, F_a refers to distribution of SW_1, while F_b refers to distribution of SW_2. *CMT* is motivated by the observation that $\mu_i \neq \mu_i$ when one window is interfered with while the other is not. Let $\bar{x}_a$ denote the mean value of variables in SW_1, and $\bar{x}_b$ denote the mean value of variables in SW_b. The indicator is defined as

$$I_{cmt} := |\bar{x}_a - \bar{x}_b|. \tag{2.10}$$

It is clear that I_{cmt} tends to be smaller when $F_a = F_b$, and larger when $F_a \neq F_b$. We further discuss *CMT* performance in Section 2.3.5.

The Chi-square Test of Independence (CST)

The chi-square test of independence is used to test whether two categories (each with many cells or groups) are related or not related (independent). In order to qualify for the test, we first need to constructed a *contingency table* based on SW_1 and SW_2, as in Table 2.1. A *cell* is a continuous range on R, f_{ij} is the number of variables from sliding window i whose value drops in *cell* j, and $Col_j := f_{1j} + f_{2j}$. k is a tuneable value whose setting determines the number of *cells*. In our experiment we set $k = 6$ where $cell j := [(j-1)/5, j/5)$, for $j = 1, 2, ..., 5$ and $cell 6 := [1, \infty)$. The indicator is defined as the χ^2 value

$$I_{cst} := \Sigma_{j=1}^{k} \frac{(f_{1j} - E_j)^2}{E_j} + \Sigma_{j=1}^{k} \frac{(f_{2j} - E_j)^2}{E_j}. \tag{2.11}$$

Table 2.1: Contingency table

Window Index	Cell 1	Cell 2	...	Cell k Total)	Row
1	f_{11}	f_{12}	...	f_{1k}	l_w
2	f_{21}	f_{22}	...	f_{2k}	l_w
Column Average	E_1	E_2	...	E_k	

Again, the smaller I_{cst}, the more likely that $F_a = F_b$.

Mann-Whitney-Wilcoxon Test (MWW)

The Mann-Whitney-Wilcoxon test is a non-parametric test for assessing whether two samples of observations come from the same distribution. It is one of the best-known non-parametric significance tests. It works as follows.

First, we arrange all the variables from SW_1 and SW_2 into a single ranked series. That is, we rank all the variables, regardless of which sliding window they are in. Then we add up the ranks for the variables which come from SW_1 and denote their sum by R_1; similarly, we add up the ranks for the variables which came from SW_2, and denote their sum by R_2. Define the indicator as

$$I_{mww} := \max(R_1, R_2). \tag{2.12}$$

It is known that if two populations share the same distribution, the probability of an observation from one population exceeding an observation from the second population is 0.5. Based on this fact, R_1 and R_2 tend to be closer, which means I_{mmw} is smaller when $F_a = F_b$.

Deciding the Threshold I^{th}

The value of I^{th} directly affects the performance of PCC. Instead of choosing I^{th} from statistical tables, we set I^{th} through training sets obtained from controlled experiments. Given a group of corrupted packets for which we already know the reason of corruption, we specify a false positive rate threshold β, and then choose the corresponding threshold I^{th}_{β} to be the smallest real number such that the false positive rate of the training set is less than β. In the wireless protocol context, this design leaves the control to the actual link manager, so that it can choose the balance between false positive rate and hit rate, based on the application on its own preference.

2.3.5 Discussion on Performance of the Tests

The performance of all three statistical tests is affected by F_i and F_n, the parameters of PCC (such as sliding window size l_w), and even packet length. It will be helpful if we can analytically determine how well PCC works under certain scenarios. In other words: Given a certain threshold I_β^{th}, is it possible to approximately estimate the corresponding false positive rate and the miss rate mathematically? In this section, we will answer this question when PCC uses *CMT*. The analysis of *CST* and *MWW* can be done similarly.

The False Positive Rate

To facilitate the derivation of parameters of interest, for all corrupted packets due to weak signal, we assume that (i) the ground noise (relative to signal strength) for every packet conforms to Ray-leigh distribution $Rayleigh(\sigma)$, and (ii) every such packet is l symbols long. Instead of determining the false positive rate for a given I^{th}, it is equivalent to find out what is the probability that PCC identifies one "weak signal" packet as interfered. Note that X_{ij} can be regarded as $N_{ij} + I_{ij}$, where N stands for noise and I stands for interference, with $I_{ij} = 0$ when there there is no interference. Therefore $F_n = Rayleigh(\sigma)$. According to the central limit theorem (CLT) which states that the sum of a sufficiently large number of identically distributed independent random variables each with finite mean and variance will be approximately normally distributed, we may approximate

$$\bar{x}_a, \bar{x}_b \sim N(\sigma\sqrt{\frac{\pi}{2}}, \frac{(4-\pi)\sigma^2}{2l_w}). \tag{2.13}$$

Then, because the difference between two normally distributed random variables is also normally distributed, it follows that

$$(\bar{x}_a - \bar{x}_b) \sim N(0, \frac{(4-\pi)\sigma^2}{l_w}). \tag{2.14}$$

For each run of CMT, the probability that H_0' is accepted is

$$p_1 = p(H_0') = p(|\bar{x}_a - \bar{x}_b| < I^{th}) = 2 \times \Phi(\frac{I_\beta^{th}}{\sqrt{(4-\pi)/l_w}\sigma}) - 1. \tag{2.15}$$

Since the step size of the sliding window is m, CMT is run $(l-1)$ times for each packet. Therefore the probability that PCC mistakes a "weak signal" packet as interfered is

$$p_f = 1 - p_1^{l-1}, \tag{2.16}$$

which is therefore the false positive rate. It easily follows from the derivation that given I^{th}, p_f is smaller with a lower ground noise level (smaller σ), shorter packets (smaller l), and larger sliding window size l_w.

The Miss Rate

Again, to facilitate the derivation of parameters of interest, for all corrupted packets due to interference, we assume that (i) the ground noise (relative to signal strength) for every packet conforms to Ray-leigh distribution $Rayleigh(\sigma_n)$, (ii) every packet is l symbols long, and (iii) $|X_{ij}|$ from the interfered part conforms to distribution F_i with mean μ_i and variance σ_i^2. When the sliding window moves to the change point, and assuming that SW_2 is interfered with, similar to the derivation in previous section, we obtain

$$\bar{x}_a \sim N(\sigma\sqrt{\frac{\pi}{2}}, \frac{(4-\pi)\sigma^2}{2l_w}), \tag{2.17}$$

$$\bar{x}_b \sim N(\mu_i, \frac{\sigma_i^2}{l_w}), \tag{2.18}$$

$$(\bar{x}_a - \bar{x}_b) \sim N(\mu_d, \sigma_d^2), \tag{2.19}$$

where $\mu_d = \mu_i + \sigma\sqrt{\frac{\pi}{2}}$ and $\sigma_d^2 = \frac{2\sigma_i^2+(4-\pi)\sigma^2}{2l_w}$. Therefore, the probability that PCC mistakes the packet as having a weak signal is

$$\begin{aligned} p_m &= 1 - p(H_1') = 1 - p(|\bar{x_a} - \bar{x_b}| < I^{th}) \\ &= 1 - \Phi(\frac{I_\beta^{th} - \mu_d}{\sigma_d}) + \Phi(\frac{-I_\beta^{th} - \mu_d}{\sigma_d}). \end{aligned} \tag{2.20}$$

It easily follows from the derivation that given I^{th}, p_m is smaller with lower ground noise level (smaller σ_n), stronger interference (larger μ_i), and larger sliding window size l_w.

2.4 Experimental Setup

2.4.1 GNU Testbed

We have evaluated PCC in a three-node GNURadio testbed. Each node is a commodity PC connected to a USRP GNU radio [1].

(a) Hardware and Software Environment: We use the Universal Software Radio Peripheral (USRP) for our RF sender/receiver, and the RFX2400 daughterboards which operate in the 2.4 GHz range. The OFDM software implementation for the signal processing blocks is from the open source GNU Radio project [2].

(b) Modulation. The OFDM implementation uses the modulation/demodulation module as a black-box, and works with a variety of modulation schemes. In our experiment, however, we only use differential quadrature phase-shift keying (DQPSK). The reason is that the PN synchronization implementation

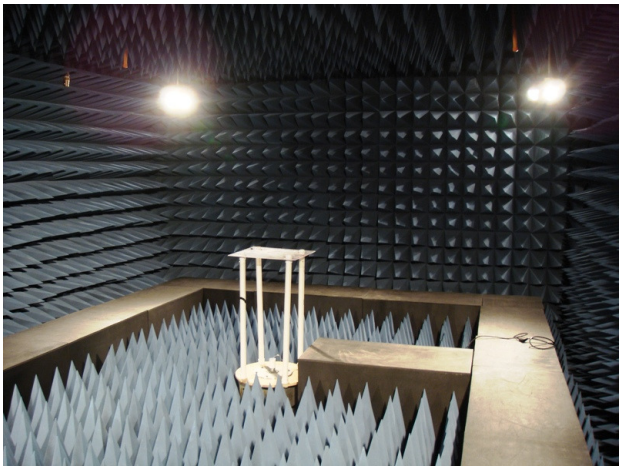


Figure 2.4: The inside of an anechoic chamber

[52] of our software has no fine-synchronization stage. DPSK requires less accurate synchronization than non-differential modulation/demodulation schemes; therefore we choose DQPSK to minimize the effect of inaccurate synchronization.

(c) Configuration Parameters. We use the default GNU Radio configuration, i.e., on the transmitter side the DAC rate is 128e6 samples/ s, the interpolation rate is 128. On the receiver side, the ADC rate is 64e6 samples/s, and the decimation rate is 64. The number of sub-carriers is 102. Each packet consists of a PN preamble, a 300-byte (or 600-byte) payload, and 32-bit CRC. This implies that the body of each packet is 12 (or 24) symbols long.

2.4.2 Experimental Environment

We have run the experiments in the Illinois Wireless Wind Tunnel (iWWT) [61], an electromagnetic anechoic chamber (Figure 2.4 shows the inside view of iWWT). An anechoic chamber is a shielded structure, with two important properties: (i) shielding prevents sources external to the chamber from interfering with reception at hosts within the chamber; and (ii) the anechoic chamber is lined internally with absorbing foam panels, which reflect minimal energy. Due to the second property, the walls of the chamber become essentially "invisible" to the devices inside the chamber. We have chosen iWWT as our experimental environment for property (i), because we must have full control of interference sources in order to establish the "ground truth" about the reason for a packet corruption. In iWWT, there are no interference sources except those that we deliberately input.

2.4.3 Scenario Design

All the experiments were conducted in iWWT, and the collected results are used as training sets for PCC. T_1 and T_2 are two transmitters and R is the receiver. T_1 and T_2 transmit over the same spectral band, but append different preambles to the packets they send, so that R can only detect the packets from T_1. We create two scenarios, with the second scenario divided into four categories:

Scenario A: Weak signal. T_2 is shut down in this scenario. We fix the distance between T_1 and R, and change the received signal power at R by tuning T_1's transmission power P_1. We gradually decrease P_1 to find (i) the threshold level (denoted by P_{no}) when R detects no packets from T_1, and (ii) the level (denoted by P_c) when almost all packets that R received are corrupted. We say a packet is corrupted if its bit error rate (BER) is higher than 1%. We assume a packet with (BER) lower than 1% can be recovered by forward error correction that is used by almost all current wireless protocols. Then we set P_1 to some value between P_{no} and P_c, and let T_1 transmit until R receives enough corrupted packets. All the packets are logged for further processing. We ran the experiments several times with different P_1 values within $[P_{no}, P_c]$. For each power level, we run the experiment twice, sending packets of size 300 bytes and 600 bytes respectively, and call them scenarios A_1 and A_2 respectively. In this way we obtain two training sets of packets corrupted due to weak signals: S_{A_1} and S_{A_2}.

Scenario B: Interference. In this scenario, T_1-R is the sender-receiver pair, while T_2 is the interference source. Both T_1 and T_2 send packets periodically, with the packet rate of T_2 being lower. By looking into the synchronized packet logs at the transmitters, we obtain the "ground truth" about which packets from T_1 have overlapping with packets from T_2, and those packets are then candidates for the training set of packets corrupted due to interference. T_1 and T_2 are placed the same distance away from R, so that we can assume that the ratio of their signal strengths at R is the same as the ratio of transmission powers they use. We repeat experiments by changing P_1 and P_2, and the interference scenario can be further divided into four sub-scenarios based on the following combinations:

B_1**: Strong signal/strong interference** - We can determine the threshold level of P_1 (denoted by P_s) such that over 98% of received packets have zero BER. When $P_1 \geq P_s$, we say it is a *strong signal scenario*. When $P_2 \geq P_1$, we say the *interference is strong*. To summarize, we label the scenario as B_1 when $P_2 \geq P_1 \geq P_s$. Again, we repeat experiments several times using different power values, log packets received at R and label them with the appropriate (P_1, P_2) pair. Packets having the same label form a *group*.

B_2**: Strong signal/weak interference**. Interference is called *weak* when $P_2 \leq P_1$. The scenario is labeled as B_2 when $P_1 \geq P_s$ and $P_2 \leq P_1$.

B_3**: Borderline signal/strong interference** - We can determine the borderline value of P_1 so that for

$P_1 \geq P_b$, 98% of the received packets are not corrupted. The scenario is labeled as B_3 when $P_2 \geq P_1 = P_b$.

B_4**: Borderline signal/weak interference**. The scenario is labeled B_4 when $P_1 = P_b$ and $P_2 \leq P_1$.

It is clear that $P_s > P_b > P_c > P_{no}$. When P_1 decreases from above P_s to below P_{no}, the link condition between T_1 and R downgrades from good, to fair, to noisy, and finally to no link at all.

2.5 Performance

We assess the performance of PCC and the three classification algorithms by evaluating how accurately they can separate corruptions due to weak signals from those due to interference in scenarios B_1, B_2, B_3, and B_4.

2.5.1 Performance Comparison of Algorithms

In order to compare the classification accuracies of CMT, CST and MWW, we first run PCC on S_{A_1}, pick I_β^{th} so that the false positive rate in S_{A_1} is β. Then we run PCC (with *indicator threshold* being I_β^{th}) on packets collected in scenarios B_1 to B_4, and compare the miss rate of each *group* between CMT, CST and MWW . Both the sliding window size and step size are 102, i.e., the number of *sub-carriers*. Figures 2.7, 2.8, 2.9 show the *classification* of packet corruptions gained by CMT, CST and MWW, respectively. Each histogram is from one *group* of the scenario. The (P_1, P_2) setting is (P_s, P_s) for Figures 2.7.b, 2.8.b, 2.9.b and $(P_b, 0.45 \times P_b)$ for Figures 2.7.c, 2.8.c, 2.9.c. Collectively the nine figures illustrate how well the classification is done by different algorithms under different scenarios. Note that the less two histograms overlap, the better the classification.

In scenarios B_1 and B_2, all three algorithms identify the cause of corruption quite accurately. For CST and MWW, the miss rate is less than 0.5%, with false positive rate 0.5% for any *group* of packets. For CST and MWW, the miss rate is less than 0.5% with false positive rate 3%. In scenarios B_3 and B_4, the accuracy of all three algorithms drops (especially in B_4), but CST drops faster than the other two. Figure 2.6 compares the performance of the algorithms under scenario B_4, and we explain the reason in the next section.

2.5.2 The Impact of Signal/Interference Strength

First we explain why classification accuracy drops in scenario B_4 so dramatically for CMT, as shown in Figure 2.6. Recall that as discussed in Section 2.3.5, given I^{th} in CMT, the miss rate decreases when $\mu_d = \mu_i + \sigma\sqrt{\frac{\pi}{2}}$ increases. A strong signal from the sender implies lower ground noise (smaller σ_n); interference stronger than the signal pulls S_{ij} far away from constellation points $\bar{a}_k$, hence yields larger μ_i. When either the signal

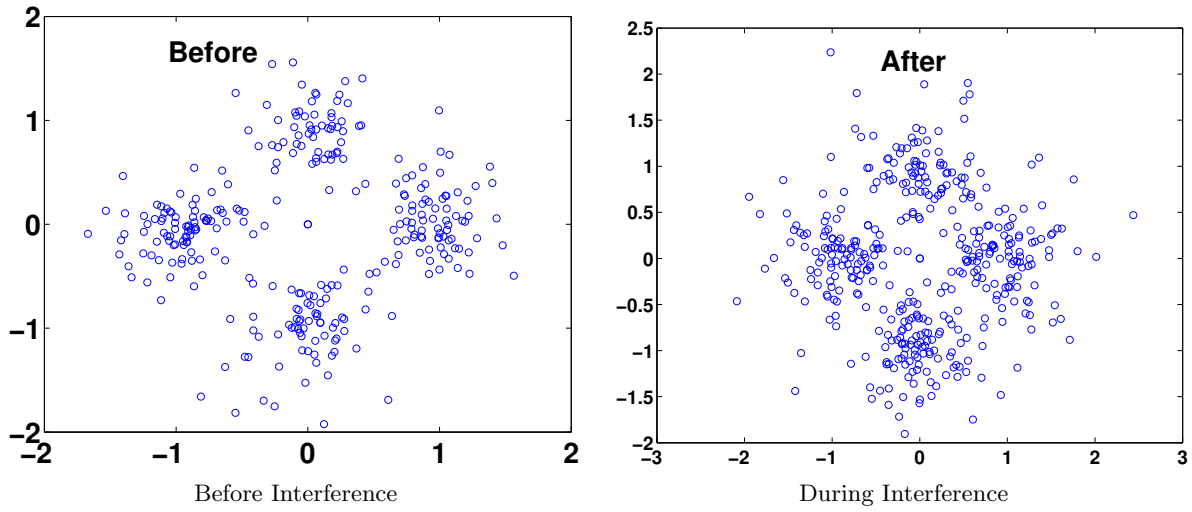


Figure 2.5: Consellation graph of an interfered packet with $P_1 = P_b$ and $P_2 = 0.45 \times P_1$

or the interference is strong, we have μ_d large enough for CMT to perform well. In scenario B_4, however, when σ_n is not small enough and μ_i is not large enough, CMT barely works. Figure 2.5 illustrates this by showing the constellation graph of a weakly interfered packet with sender's signal strength at borderline. On the other hand, the classification accuracy of CST and WMM does not rely on μ_d but on how different the two distributions F_n and F_i are. Experimental results show that as long as (i) the sender's signal strength is at least fair (i.e., $P_1 \geq P_b$); and (ii) the interference is strong enough to cause packet corruption, then even in the worst case, F_n and F_i are different enough for CST and WMM to attain a miss rate lower than 10%, with the false positive rate threshold set to 5%. WMM has a slightly better performance than CST by achieving miss rate 6%, with the false positive rate threshold set to 5%.

2.5.3 The Impact of Packet Size

When I^{th} is fixed, changing the packet size l does not affect miss rate very much. But a larger packet size l may increase the false positive rate for all three tests. This is because as suggested in (2.16), the longer the packet, the more likely that one of the $l-1$ tests will return an indictor larger than I^{th}. Table 2.2 illustrates the impact of packet size and verify this. Here we choose the threshold I^{th} by fixing the miss rate, then we compare the false positive rate in S_{A_1} and S_{A_2}.

2.5.4 The Impact of Sliding Window Size

The sliding window size l_w is the size of samples to run the tests. Statistically, the larger l_w is, the better performance is, and experimental results support this. Table 2.3 assesses how well PCC-($l_w = 102$) and

Table 2.2: The comparison of the false positive rate between A_1 and A_2. The packets corrupted due to interference are those collected with $P_1 = P_b = 2.2 \times P_2$. The miss rate is 8%.

	CMT	CST	MWW
A_1	0.2874	0.0784	0.0327
A_2	0.5014	0.1851	0.0399

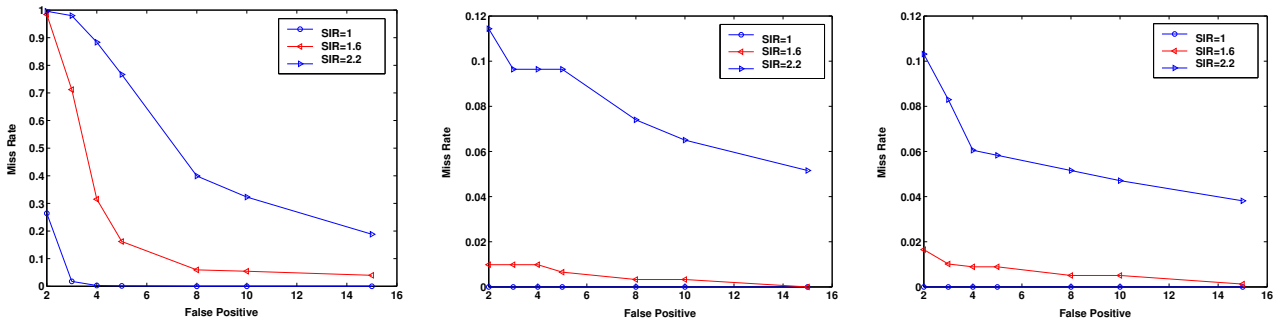
Table 2.3: The change of the false positive rate by changing the sliding window size l_w. The miss rate is 8%.

	CMT	CST	MWW
l_w=102	0.2874	0.0784	0.0327
l_w=50	0.4686	0.1251	0.0349

PCC-($l_w = 50$) separate packet corruptions in scenario A_1, from the corruptions in scenario B_4 with $P_1 = P_b = 2.2 \times P_2$. However, larger l_w costs more in computation. Moreover, if l_w is so large that it covers $w > 1$ symbols, then the interference that starts/terminates among the first (or last) w symbols may not be detected. Therefore, l_w must be tuned carefully.

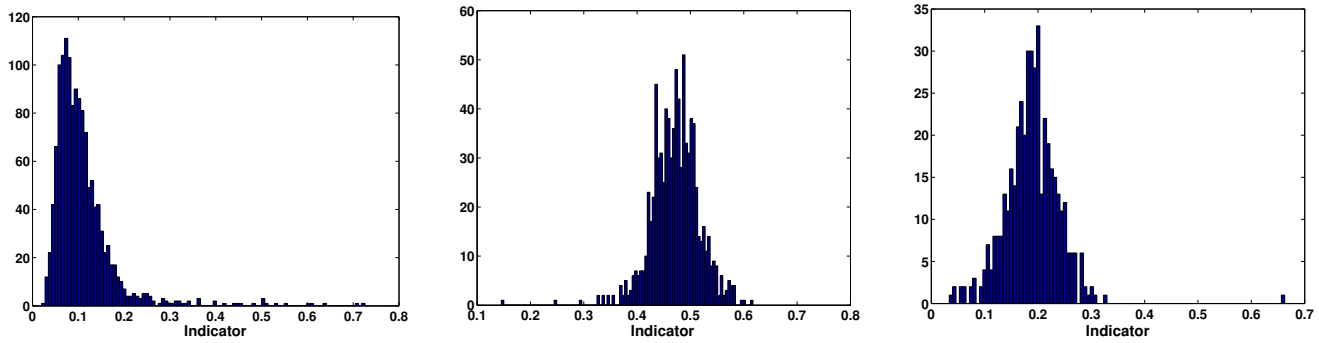
2.6 Interference from an Adjacent Band

This section discusses how PCC can be extended to detect interference from wireless devices transmitting on an adjacent spectral band, but "crossing the line". Recently there has been much cognitive radio/ultra-wideband usage aim at making use of as much of the unused spectrum as possible. Some aggressive protocols have been designed. For example, [47] proposes a wideband network protocol that keeps grabbing more and more spectrum till it learns from observed backoff activities of other narrowband devices that it hurts them, and then releases some of the spectrum. We expect that in the future when systems running different



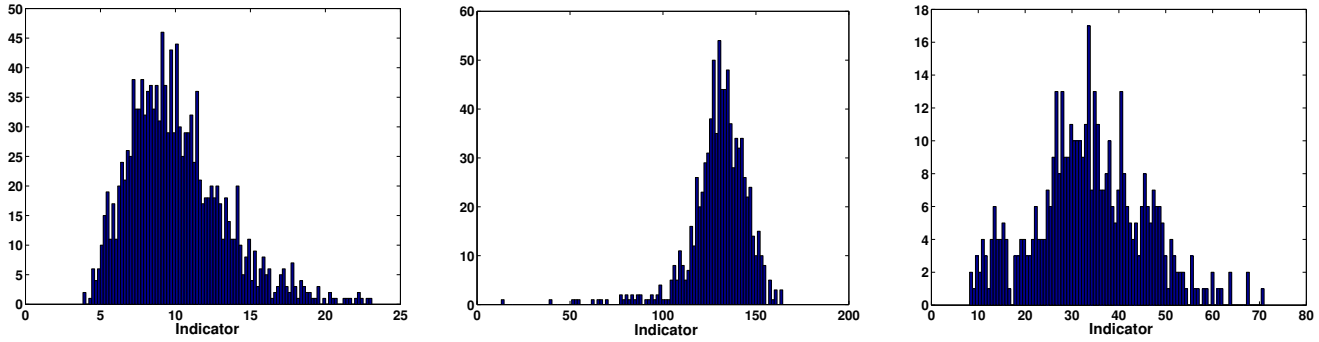
(a) mean value change (b) chi-square test (c) mann whitney W test

Figure 2.6: Miss rate comparison of three tests, under the scenario that sender's signal strength is around -80dbm at the receiver's side



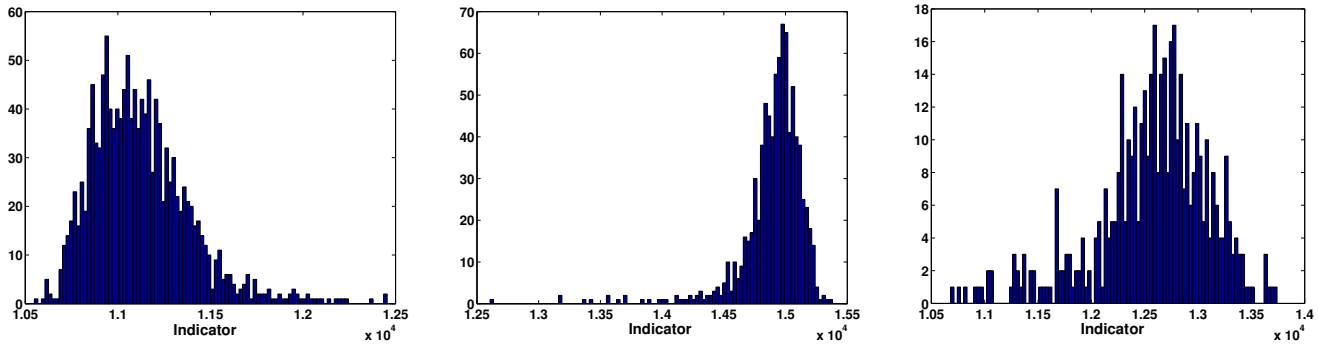
(a) Weak Signal (b) Strong Signal/Strong Interference (c) Weak Signal/Weak Interference

Figure 2.7: Histogram of Indicator output from CMT



(a) Weak signal (b) Strong Signal/Strong Interference (c) Weak Signal/Weak Interference

Figure 2.8: Histogram of Indicator output from CST



(a) Weak Signal (b) Strong Signal/Strong Interference (c) Weak Signal/Weak Interference

Figure 2.9: Histogram of Indicator output from MWW

protocols operate together, interference from adjacent bands may be quite common.

When interference is from an adjacent band, only the *sub-carriers* on the edge where two bands overlap are affected. Therefore when such interference happens, we observe changes not only between symbols, but also between *sub-carriers*. However, if the overlapped band is too narrow, the interference cannot be well detected by PCC. The reason is that in order for PCC to work properly, the sliding window size cannot be too small. If the number of interfered *sub-carriers* is too small, PCC cannot get enough sample points to run the statistical tests.

Therefore we make the following assumption: we want to detect interference that affects at least k *sub-carriers*. Without loss of generality, in the following we only consider interference overlapping with the first to k-th *sub-carriers*. Then we execute the following steps to detect the interference: (i) Choose the sequence S to be $x_{(i-1)\times k+j} = |X_{ij}|$, for $1 \leq i \leq l$. This means that instead of processing the whole packet, we only process the contents of the first k *sub-carriers*; (ii) run PCC(k, l_w) on S to detect change point(s), noting that because k is small, one window may cover several symbols; (iii) if a change point is detected, then locate the duration of interference in the time-line. Suppose the interference starts at the s-th symbol and ends at the d-th, which covers l_i symbols; (iv) process the chunk composed by the symbols interfered. Choose S to be $x_{(i-s+1)+j\times l_s} = |X_{ij}|$, for $s \leq i \leq d$. Run PCC(l_w, l_w) to detect the change point along *sub-carriers*, but fix SW_1. We accept the hypothesis that there is adjacent band interference only when PCCs in (ii) and (iv) both find change points.

We have also conducted experiments in iWWT to evaluate the above solution. The parameter settings are $l_w = 40$ and $k = 20$. In a borderline signal/strong interference scenario, when around 25 out 102 *sub-carriers* are interfered with, PCC-CST has a hit rate over 99%, with false positive rate less than 3%.

2.7 Related Work

Paper [48] presents the first empirical study based on bit error patterns of received data for loss diagnosis in 802.11. The authors proposed several novel symbol level metrics, such as symbol error rate, error per symbol and S-score, to separate collisions (caused by interference from the same system) from weak signal. Those metrics motivated our design of PCC to some extent. However, their scheme, called COLLIE, requires the receiver to send back the whole corrupted packets back to the sender, which introduces a large reverse traffic overhead. Our PCC, on the other hand, only requires the receiver to send a few bits to the sender, but provides more accurate classification than COLLIE. Besides [48], there have been other research efforts aimed at diagnosing the packet loss problem. Yun and Seo [68] propose to detect collisions in 802.11 links by

measuring the RF energy and its changes. However, this work was done through simulation only. The rate adaption mechanism CARA of [18] tries to detect collisions by using RTS-CTS. Their scheme fails in the presence of hidden terminals, which are actually major sources of collisions. None of the above approaches make use of *signal-level* physical layer information, but rather use bit level information, or the MAC layer logical information exported by commodity wireless devices.

In recent years, there has been a growing interest in the wireless network community at making use of more information from physical layer. We note a few works. Jamieson *et al.* [27] develop a partial packet recovery (PPR) system. PPR has a key component called *softPHY*, which is an expanded physical layer interface that provides hints to higher layers about the physical layer's confidence in each bit it decodes. The hint is based on the the Hamming distance in block decoding, which is a different approach from ours. Gollakota *et al.* [20] proposes Zigzag decoding to solve the hidden terminal problem by locating the start of the second packet when two packets overlap (thus collide). Zigzag's success is built upon the capability of the DSSS physical layer to detect preambles of packets at very low SINR, which does not apply in the OFDM system. Ziptx from Lin *et al.* [42] can improve system throughput by using pilot bits to detect the per-symbol BER, and using forward error correction to correct symbols with low BER while retransmitting symbols with high BER. Ziptx explores new ways of recovering from packet corruptions and also suggests different solutions to different corruptions.

2.8 Conclusions

In this chapter, we have tried to determine if one can identify the cause of packet corruptions in the OFDM system. Unlike most of the previous approaches, our proposed scheme, PCC, looks directly into available physical layer information at *signal-level*, and uses three *statistical* tests to classify packets corrupted as being due to interference or weak signals. Through evaluations conducted on a GNU radio test bed in iWWT over a wide range of experiments, we identify the working ranges of the *statistical* tests, and compare their performances under different channel conditions. We also provide analytical explanations to the experimental results. When PCC uses the most accurate test WMM, it can achieve a low miss rate (of interference) of 6%, with the false positive rate threshold set to 5%, even under the scenario where interference is hardest to detect. Since all experimental results and conclusions given in this paper are based on experiments conducted on a real test bed, we expect that the implications of our results can be very useful in upper-layer problem domains, such as link scheduling, channel management, packet recovery design, etc., where a better understanding of the link behavior can help a lot.

Chapter 3

Modeling the Effect of Carrier Sensing and Back-off on Network Capacity

3.1 Introduction

Because the medium in wireless networks is by nature shared among nodes in the spatial domain, media access control (MAC) plays an important role of coordinating medium access among nodes. The IEEE 802.11 [6] distributed coordination function (DCF) protocol is a CSMA/CA MAC protocol that has been widely studied and used in wireless networks because of its distributed nature and ease of implementation. Essentially DCF arbitrates medium access with two mechanisms: *carrier sense multiple access* for detecting simultaneous transmissions and for mitigating interference and *binary exponential back-off mechanism* for resolving contention.

DCF carrier sense can be categorized into *virtual* carrier sense and *physical* carrier sense, and we focus on the latter. Before attempting for transmission, a node senses the medium and defers its transmission if the channel is sensed busy, i.e., the strength of the received signal exceeds a certain threshold CS_{th}. Carrier sense reduces the likelihood of collision by preventing nodes in the vicinity of each other from transmitting simultaneously, while allowing nodes that are separated by a safe margin (termed as the carrier sense range, R_{cs}) to engage in concurrent transmissions. The latter effect is referred to as *spatial reuse*. In multi-hop wireless networks, the choice of the carrier sense range depends on CS_{th} and the minimum Signal-to-Interference-and-Noise-Ratio (that gives the minimal SINR at which the received signal can be correctly decoded). If the transmit power of all the nodes is the same, a large value of CS_{th} implies a small value of R_{cs} (i.e., better degree of spatial reuse), but the interference to be tolerated by the transmission may be also high. A small value of CS_{th}, on the other hand, implies a larger value of R_{cs}, but the resulting SINR will be comparatively higher.

Several research efforts have been made to understand the impact of carrier sense and spatial reuse on system performance. Zhu *et al.* [29] attempt to identify the optimal carrier sense threshold that maximizes spatial reuse in a regular topology. Jamieson *et al.* [28] carry out an empirical study to understand the limitation of carrier sense. Yang and Vaidya [67] show that the MAC layer overhead has a great impact on

the choice of the carrier sensing range and the data rate. Zhai and Feng [69] point out that the carrier sense threshold that maximizes the network capacity does not vary significantly with the channel data rates.

The binary exponential back-off mechanism, on the other hand, aims to resolve collision. If nodes that are spatially close to each other sense the medium to be idle and transmit simultaneously, collisions occur. Alternatively, the accumulative interference contributed by the concurrent transmissions of multiple nodes outside the carrier sense range could be so significant that it corrupts the transmission. The binary exponential back-off mechanism is designed to deal with these situations. Each node must wait for a time interval specified by the contention window, before it starts to transmit (after sensing the channel idle for DIFS). An adequate contention window size reduces the collision probability, while not wasting the bandwidth by having all the nodes busy backing off.

There have been several models that characterize the transmission activities governed by IEEE 802.11 DCF and study how to tune the contention window size in single-cell WLANs. (See, for example, [34] for an excellent survey.) Calí *et al.* [14] and Bianchi [10] devise, respectively, two analytical models to calculate the system capacity in WLANs. In particular, Bianchi [10] models the behavior of the binary backoff counter at one tagged station as a discrete Markov chain model. Calí *et al.* [14] derive a theoretical throughput bound by approximating IEEE 802.11 with a p-persistent model of IEEE 802.11. Both observe that the system throughput only relies on the contention window size and the number of active stations. They also show that with the current parameter settings of IEEE 802.11, the maximal achievable system throughput falls far beneath the theoretical capacity bound. Kumar *et al.* [36] present a fixed point analysis of Bianchi's model in the asymptotic regime of a large number of stations, and give explicit expressions for the collision probability, the aggregate attempt rate, and the aggregate throughput. All these studies focus on single-cell wireless LANs.

Medepalli and Tobagi [44] extend Bianchi's model to accommodate the effect of hidden/exposed nodes in *multi-hop* wireless networks. They show that the minimum contention window size used in the exponential back-off mechanism has a more profound effect on mitigating flow starvation than the maximum contention window size. What has not been exclusively addressed in the study is (i) the impact of carrier sense threshold (that determine the sharing range of the wireless medium and hence the extent of spatial reuse) on the system performance and (ii) the interplay between the carrier sense threshold and the contention window size.

In this chapter, we propose an analytical model that extends Calí's model to multi-hop wireless networks and incorporates the effects of physical carrier sense, SINR, and collision caused by accumulative interference. The major difficulty in modeling multi-hop wireless networks lies in that the carrier sense threshold CS_{th} plays an important role in several aspects. First, when node i attempts for transmission, it has to ensure

that the received signal strength does not exceed CS_{th}. In WLANs, this means all the other nodes do not transmit at the time of physical carrier sense. However, in multi-hop wireless networks, this merely means that all the nodes within the carrier sense range R_{cs} of node i do not transmit.[1] Second, the condition for successful transmission in WLANs is that no more than one nodes attempt to transmit *simultaneously*. Once the transmission of a node starts, all the other nodes will sense a busy medium and be silenced. However, in multi-hop wireless networks, after a node senses an idle medium and starts transmission, nodes outside R_{cs} may still engage in new transmissions, adding to the level of interference. Whether or not node i's transmission succeeds then depends on the *Signal-to-Interference-and-Noise-Ratio* (SINR) — if the SINR perceived by node i is smaller than a minimum SINR threshold β, the transmission cannot be correctly decoded and is thus failed. CS_{th} is a tunable parameter that controls spatial reuse and transmission quality (as determined by interference among concurrent transmissions). A larger value of CS_{th} allows better spatial reuse at the expense of increased interference (and hence the likelihood of frames being corrupted because of accumulative interference). The impact of CS_{th} on the systems throughput has not been extensively studied (at the level as detailed as that in Calí's model).

In our proposed model, we follow Calí's methodology of characterizing the time interval of two consecutive successful transmissions as the *virtual transmission time*, t_v, and derive its expected value. We faithfully incorporate all the aforementioned effects that CS_{th} makes in multi-hop wireless networks, and derive $E(t_v)$ *as perceived by nodes in one interference range*. (The reason for choosing the "cell" as nodes within an interference range will be elaborated on in Section 3.3.) We validate the derived model via simulation, and make several important implications from the analytical model.

The rest of this chapter is organized as follows. In Section 3.2 we summarize Calí's model and the Hexagon interference model [40] that will be used throughout the paper. In Section 3.3, we present the analytical model that characterizes the transmission activities governed by IEEE 802.11 DCF in multi-hop wireless networks. In Section 3.4, we discuss several hints we can get from the derived model. In Section 3.5, we validate the proposed model via simulation and make several important observations from both the analytical/simulation results. Finally, we conclude the paper in Section 3.6 with a list of future research agendas.

3.2 Background Material

In this section, we summarize 1) the p-persistent model of IEEE 802.11 DCF proposed by Calí *et al.* [14][15]; 2) the hexagon interference model which describes the worst-case interference which a wireless (receiver) node

[1] Note that we do not consider channel errors in this paper.

Table 3.1: Systems parameters

Parameter	Description
M	Number of active hosts
p	Probability that each node attempts for transmission when the medium is sensed idle
q	Parameter for the geometric distribution of the packet size, i.e., $\Pr\{\text{packet\_length} = \text{i slot}\} = q^{i-1}(1-q)$.
$\bar{m}$	Average transmission time, i.e., $\bar{m} = t_{slot}/(1-q)$
τ	Maximum propagation time
DIFS	Distributed interframe spacing
SIFS	Short interframe spacing
EIFS	Extended interframe spacing
ACK	Time required to transmit the ACK
N_c	Number of collisions in a virtual transmission time
$E(T_c)$	Average length of a collision period
$E(I)$	Average duration of consecutive idle slots before a successful transmission or a collision
$E[S]$	Time required to complete a successful transmission (including all the protocol overheads), i.e., $E(S) = \bar{m} + SIFS + ACK + DIFS$.

may experience with the use of physical carrier sense. These models will be leveraged throughout the paper.

3.2.1 *p*-persistent Model That Characterizes IEEE 802.11 DCF in WLANs

For analytical tractability, Calí *et al.* [15, 14] consider a *p-persistent* version of IEEE 802.11 DCF, which differs from the standard protocol only in the selection of the backoff interval. Instead of using the binary exponential backoff timer values, the *p-persistent* version determines its backoff interval by sampling from a geometric distribution with parameter p. Due to the memoryless property of this geometric-distributed backoff algorithm, it is more tractable to analyze the *p-persistent* IEEE 802.11 protocol.

The analytic model is derived under the assumption that all the stations always have packets ready for transmission (which is termed the *asymptotic* condition in [15, 14]). Under the geometrically-distributed backoff assumption, the process that characterizes the occupancy behavior of the channel (idle slots, collisions, and successful transmission) till the end of each successful transmission is regenerative, with the sequence of time instants corresponding to the completion of successful transmission being the regenerative points. Calí *et al.* exploit this regenerative property and define the jth virtual transmission time as the time interval between the jth and *(j+1)*th successful transmissions. As shown in Figure 3.1, idle periods and collisions precede a successful transmission, where an idle period is a time interval in which the channel is idle due to the fact that all the back-logged stations are in the back-off mode, and a collision is the interval in which two or more stations attempt for transmission and their packets collide with one another.

Let t_v be defined as the average virtual transmission time, I_i and $T_{c,i}$ as the length of the ith idle period

and the length of the ith collision in a virtual transmission time respectively. Given the major system parameters in Table 4.1, the protocol capacity ρ can be expressed as

$$\rho = \frac{\bar{m}}{t_v}, \tag{3.1}$$

where

$$\begin{aligned} t_v &= E\left(\sum_{i=1}^{N_c}(DIFS + I_i + T_{c,i} + SIFS)\right) + (N_c + 1)\cdot E(I) + E(S) \\ &= E(N_c)\cdot(E(T_c) + DIFS + SIFS) + (E(N_c) + 1)\cdot E(I) + \bar{m} + SIFS + ACK + DIFS. \end{aligned} \tag{3.2}$$

Note that $SIFS$ and ACK in the first term on the right hand side of Eq.(3.2) is due to the extra waiting period in EIFS after detection of an incorrectly-received frame (i.e., frame collision). Note that in Calí's model, it is assumed that each station waits for an interval of DIFS after a frame collision, while we assume the use of EIFS here.

The expressions of $E(N_c)$, $E(T_c)$, and $E(I)$ have been derived for WLANs [15, 14]:

$$E(N_c) = \frac{1-(1-p)^M}{Mp(1-p)^{M-1}} - 1, \tag{3.3}$$

$$E(T_c) = \frac{t_{slot}}{1-(1-p)^M - Mp(1-p)^{M-1}} \times [\sum_{h=0}^{\infty}\{h \times [(1-pq^h)^M - (1-pq^{h-1})^M]\} - \frac{Mp(1-p)^{M-1}}{1-q}], \tag{3.4}$$

$$E(I) = \frac{(1-p)^M}{1-(1-p)^M} \times t_{slot}. \tag{3.5}$$

3.2.2 Hexagon Interference Model

The hexagon interference model has been used to to calculate the worst-case SINR given that every node senses the medium before attempting for transmission. Specifically, let P denote the transmission power used by a sender, P_r the received power at the corresponding receiver, r is the distance between the sender and the receiver, and θ is the path loss exponent (which usually ranges from 2 (free space) to 4 (two-ray ground)). Then we have

$$P_r = \frac{P}{r^{\theta}}. \tag{3.6}$$

With this radio propagation model, we can derive the relation between the *carrier sense range*, R_{cs}, and the carrier sense threshold, CS_{th}:

$$CS_{th} = \frac{P}{R_{cs}{}^{\theta}}. \tag{3.7}$$

$$
\begin{aligned}
SINR &= \frac{\frac{P}{R_{tx}^{\theta}}}{\frac{2P}{(R_{cs}-R_{tx})^{\theta}} + \frac{P}{(R_{cs}-R_{tx}/2)^{\theta}} + \frac{P}{R_{cs}^{\theta}} + \frac{P}{(R_{cs}+R_{tx}/2)^{\theta}} + \frac{P}{(R_{cs}+R_{tx})^{\theta}}} \\
&= \frac{1}{\frac{2}{(X-1)^{\theta}} + \frac{1}{(X-1/2)^{\theta}} + \frac{1}{(X)^{\theta}} + \frac{1}{(X+1/2)^{\theta}} + \frac{1}{(X+1)^{\theta}}}
\end{aligned} \tag{3.9}
$$

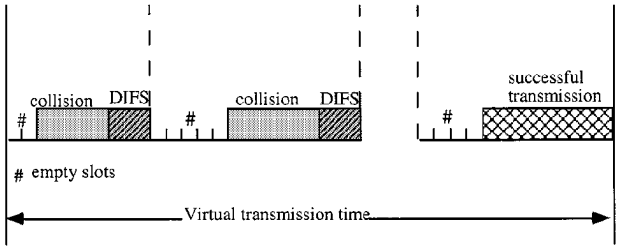


Figure 3.1: Structure of virtual transmission time.

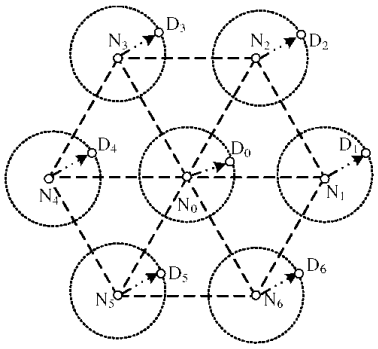


Figure 3.2: Interference Model. D_i is the intended receiver of sender N_i $(1 \leq i \leq 6)$.

In order to decode the received signal correctly, the received signal is required to exceed a threshold called the *receive sensitivity* (RX_{th}). By Eq. (3.6), the maximum transmission range R_{tx} can be calculated as

$$
R_{tx} = (\frac{P}{RX_{th}})^{\frac{1}{\theta}}. \tag{3.8}
$$

Let X be defined as the ratio between R_{cs} and the transmission range R_{tx}, i.e., $X \triangleq \frac{R_{cs}}{R_{tx}}$.

Figure 3.2 shows the scenario in which the receiver D_0 incurs the worst-case interference. Note that D_i is the intended receiver of sender N_i $(1 \leq i \leq 6)$, and the distance between N_i and D_i is R_{tx}. By the definition of CS_{th}, the distance between any two adjacent senders is at least R_{cs}. N_1–N_6 constitute the six 1^{st} tier interference nodes that are located at the closest possible locations to D_0. Let P_r denote the power received at D_0 from N_0 and $P_{r,i}$ the power received at D_0 from N_i. It has been shown in [40] that the worst case interference (and hence the smallest SINR at receiver D_0) is incurred when D_0 is so positioned that the six 1^{st} tier interference nodes are, respectively, of distance $R_{cs} - R_{tx}$, $R_{cs} - R_{tx}$, $R_{cs} - R_{tx}/2$, R_{cs}, $R_{cs} + R_{tx}/2$, and $R_{cs} + R_{tx}$ away from it, as illustrated in Figure 3.2. The corresponding worst-case $SINR$ at receiver D_0 is expressed expressed in Eq. (3.9).[2]

Note that the hexagon model errs on the pessimistic side for the following reasons. First, the worst-case interference only occurs when there are six nodes located exactly at the desired locations, which rarely

[2]Note that we ignore the background noise in the expression.

happens in a random topology. Second, use of physical carrier sense usually prevents such an extreme case from taking place. As shown in Figure 3.2, if N_0 starts transmission first, then N_i may initiate a concurrent transmission because it is outside the carrier sense range of N_0. Now the accumulative (interference) signal strength at N_{i-1} (or N_{i+1}) is $2\times\text{CS}_{th}$, which prevents N_2 from initiating a concurrent transmission until N_0 or N_i completes its transmission. In general, although the number of 1^{st} tier interference nodes is six, not all of them can locates at the worst case location.

Given that the hexagon model errs on the pessimistic side and is too conservative, one may consider only the interference from one closest interference node which is R_{cs}-R_{tx} away from receiver D_0. In this case, the SINR at receiver D_0 is

$$SINR = (X-1)^{\theta}. \tag{3.10}$$

Given the minimal SINR threshold β and the transmit power P (which in turn determines R_{tx} by Eq. (3.8)), one can determine the value of CS_{th} by the following steps: (1) obtaining the value of X using either Eq. (3.9) $= \beta$ or Eq. (3.10) $= \beta$; (2) obtaining the value of R_{cs} by $X = \frac{R_{cs}}{R_{tx}}$, and the value of CS_{th} by Eq. (3.7). Let $CS_{th,1}$ and $CS_{th,2}$ denote the CS_{th} derived using Eq. (3.9) and (3.10), respectively. We have $CS_{th,1} < CS_{th,2}$. However, the optimal value of CS_{th} is even larger than $CS_{th,2}$. This will be corroborated in our simulation study in Section 3.5. Moreover, both our proposed analytical model and simulation results indicate that, when CS_{th} is slightly smaller than $CS_{th,2}$, although it cannot completely prevent collision caused by multiple concurrent transmissions from taking place, it can still keep the collision probability low, while achieving better spatial reuse.

3.3 Analytical Model

In this section, we present our analytical model that extends Calí's work to multi-hop wireless networks, with consideration of the effects of physical carrier sense, SINR, and collision caused by accumulative interference (as a result of concurrent transmissions). Recall that Calí *et al.* use their model to derive the optimal value of the attempt probability p (and hence the contention window size) that maximizes the protocol capacity. Similarly, the systems throughput derived in our model will be a function of p, the carrier sense threshold CS_{th}, the minimum SINR threshold β, and the node density, λ, in the network. Given the minimum SINR threshold β and the node density λ (which are considered part of the network configuration), we can (numerically) obtain the optimal combination of (p, CS_{th}) that maximizes the systems throughput in a multi-hop wireless network.

3.3.1 Assumptions and Notations

The following assumptions have been made to devise the analytical model:

(A1) Nodes are distributed on a plane according to a Poisson point process with node density λ.

(A2) Every node uses the same power P. The maximum transmission range R_{tx} can then be determined from Eq. (3.8) given the receive sensitivity RX_{th}.

(A3) The radio propagation model is given in Eq. (3.6). For a transmission to be successful, the SINR at the receiver must exceed the minimum SINR threshold β.

(A4) If the carrier sense threshold CS_{th} is determined, the corresponding *carrier sense radius* R_{cs} is determined by Eq. (3.7).

(A5) Whether a node decides to access the media is independent of others. This is referred to as the independent access assumption.

(A6) To derive the accumulative interference contributed by nodes that are outside R_{cs} of an intended receiver, we only consider the six 1^{st}-tier interference nodes (e.g., N_1-N_6 in Fig. 3.2). As indicated in [22][13], the interference contributed by these six interference nodes dominate.

Note that while **(A1)**-**(A4)** are consistent with PHY/MAC operations in realistic settings, **(A5)**-**(A6)** are approximations.

The major system parameters are given in Table 4.1. To ensure that any on-going transmission will not be corrupted by its closest possible competing sender, we should have that Eq. (3.10) $\geq \beta$. This implies

$$X = \frac{R_{cs}}{R_{tx}} \geq \beta^{\frac{1}{\theta}} + 1. \tag{3.11}$$

Eq. (3.11) gives a lower bound on the value of "safe" R_{cs}. Also, given an on-going transmission, we define the interference range as a circle centered at the receiver with radius R_i. If any node within the interference range transmits, the on-going transmission is corrupted at the receiver. Specifically,

$$\frac{P/R_{tx}^{\theta}}{P/R_i^{\theta}} \leq \beta \Longrightarrow R_i \leq \beta^{\frac{1}{\theta}} \times R_{tx}. \tag{3.12}$$

3.3.2 Model Overview

We follow Calí's methodology of characterizing and deriving the *virtual transmission time*. However, we have to consider the following aspects in multi-hop wireless networks:

1. We *re-define* the number of active nodes (M). In a WLAN, M is the number of active nodes in the WLAN. However, in a multi-hop wireless network the notion of a "cell" (in which the system view is

applied) has to be defined. In this paper, we define the "cell" A to be a circle with the interference range, R_i^2, as the radius, i.e., $A = \pi R_i^2$, and M be all the nodes located in A. We will elaborate on the rationale in Section 3.3.3.

2. We consider other possibilities that may lead to an idle period before either a successful transmission or a collision (Figure 3.1). In a WLAN, an idle period is consecutive idle slots in which all the back-logged nodes are in the back-off mode. However, in a multi-hop network, an idle period may also occur in A when all the nodes in A are being silenced by senders outside A but within the carrier sense ranges of these nodes. We will discuss how we take into account of various possibilities in calculating $E[I]$ in Section 3.3.4.

3. We re-derive the attempt probability p. Specifically, let E denote the event that the "cell" is idle and E_1 the event that a node *senses* the medium to be idle. Also, let $p \triangleq \Pr(\text{a node attempts for transmission } |E)$ and $p_1 \triangleq \Pr(\text{a node attempts for transmission } |E_1)$. It has been shown in [14] that $p_1 = 2/(E(CW)+1)$ in a WLAN. Moreover, because $E \equiv E_1$ in a WLAN, we have $p = p_1$. However, in a multi-hop wireless network, with the accumulative interference outside the interference range taken into account, $E_1 \subset E$, and p_1 alone is not sufficient to characterize the access probability. We will discuss how we re-derive p in Section 3.3.5.

4. We enumerate and consider all possible causes of collisions in multi-hop wireless networks. In a WLAN, collisions are only caused by simultaneous transmissions within the WLAN. However, in a multi-hop wireless network, collisions also occur when the accumulative interference contributed by *simultaneous* transmissions outside the interference range and/or *concurrent* transmissions outside the carrier sense range. We will discuss how we take into account of all these factors into calculating the collision probability P_c and the collision period N_c in Section 3.3.6.

3.3.3 Definition of a Cell in Multi-hop Wireless Networks

Recall that Calí's model characterizes transmission activities governed by DCF in a WLAN with M active nodes *from the system view*. To adapt Calí's model to a multi-hop wireless network, the first step is to define the notion of a "cell." As such, we divide the plane into "cells" of area $A = \pi R_i^2$, and focus on the system view of a "cell." The rationale for using the interference range R_i as the radius of a "cell" is as follows. In a WLAN, an on-going transmission is successful if and only if no other transmissions overlap in time with the on-going transmission. To extend Calí's model to multi-hop wireless networks, the same property should hold. By the definition of the interference range R_i (Eq. (3.12)), we know that at most one transmission is allowed in the interference range of a receiver at any time in order for the transmission to be successful. If any node within the interference range of an intended receiver transmits, it corrupts the corresponding

transmission. Therefore we choose the "cell" A to be of area πR_i^2 and set M to be the number of active nodes within an interference range, i.e., $M = \lambda \pi {R_i}^2$.

With the new definition of a "cell," $packet\_size/t_v$ characterizes the system throughput within one "cell" A whose area depends on the minimum SINR threshold β (as R_i depends on β (Eq. (3.12))). Note that β determines the data rate at which a sender can sustain — the larger the β value, the larger the achievable data rate.

3.3.4 Derivation of the Idle Period

Recall that in a multi-hop network, an idle period may occur in A when all the nodes in A are being silenced by senders outside A but within the carrier sense ranges of these nodes. That is, the idle period may also be the time interval during which the "cell" is silenced by the transmissions in other "cells".

In order to incorporate this effect, we define the *silence area* A_S to be the area that is silenced by a transmission. Consider Figure 3.3 which depicts the best spatial reuse that can be achieved when N_0 and N_1-N_6 can simultaneously engage in successful transmissions. Let D_s denote the distance between any two concurrent transmissions. Then N_0–N_6 are located at a distance of D_s away from from N_0, which yields the best spatial reuse. (Note that D_s is not equal to R_{cs}, because as explained in Section 3.2.2, not all the first-tier interference nodes located at a distance of R_{cs} can transmit at the same time.) Without loss of generality, consider the interference N_0 experiences. In order for N_0 to succeed in its transmission, the aggregate interference contributed by N_1–N_6 should be less than or equal to CS_{th}, i.e.,

$$6 \cdot \frac{P}{D_s^\theta} \leq CS_{th} = \frac{P}{R_{cs}^\theta} \Longrightarrow D_s \geq R_{cs} \times 6^{\frac{1}{\theta}}. \tag{3.13}$$

The spatial reuse is maximized when the equality holds. In this case, every sender occupies an area that is composed of six small triangles, while each triangle is shared by three senders. As a result, the *silence area* of a transmission is the area of two triangles.

$$A_S = \frac{\sqrt{3}}{2} \times {D_s}^2 = \frac{\sqrt{3}}{2} \times (R_{cs} \times 6^{\frac{1}{\theta}})^2. \tag{3.14}$$

Let $M_s \triangleq \lambda A_S$ denote the number of nodes in a silence area and φ the ratio of M_s to M ($\varphi \triangleq M_s/M$). When a node transmits, it silences not only nodes within its interference range, but also nodes within its silence area. As a matter of fact, the transmission inside an interference range may silence nodes in its $\varphi - 1$ neighboring interference ranges.

Now let $E[I_1]$ and $E[I_2]$ denote, respectively, the average number of times the medium is idle due to

the event that all the nodes inside the "cell" are in the back-off mode and the event that all the nodes are silenced by transmissions outside the interference range. If an idle period is caused by transmission(s) in other interference ranges, it lasts for for the duration of a frame transmission, i.e. $\bar{m}$ = (t_{slot}/1-q). The idle period is thus redefined as

$$E[I]=E[I_1] \times \frac{t_{slot}}{1-q} + E[I_2] \times t_{slot}. \tag{3.15}$$

No matter which event causes an idle period, the probability that an idle period occurs is $(1-p)^M$ and the number of times idle periods occur follows a geometric distribution with $(1-p)^M$. Consequently,

$$E[I_1] + E[I_2] = \frac{(1-p)^M}{1-(1-p)^M}. \tag{3.16}$$

With probability $(1-(1-p)^{M(\varphi-1)})$, at least one sender in the other $(\varphi-1)$ interference ranges is transmitting, and hence the average number of times idle periods occur because of the second event is

$$E[I_2] = \frac{(1-p)^M}{1-(1-p)^M} \times [1-(1-p)^{M(\varphi-1)}]. \tag{3.17}$$

Finally $E[I]$ can be expressed as

$$E[I] = \frac{(1-p)^M \times t_{slot}}{1-(1-p)^M} \times \left[\frac{1-(1-p)^{M(\varphi-1)}}{1-q} + (1-p)^{M(\varphi-1)}\right]. \tag{3.18}$$

3.3.5 Derivation of Attempt Probability p

Let E denote the event that the "cell" is idle, and $p \triangleq$ Pr(a node attempts for transmission $|E$). Also, let E_1 denote the event that a node *senses* the medium to be idle, and $p_1 \triangleq$ Pr(a node attempts for transmission $|E_1$). It has been shown in [14] that $p_1 = 2/(E(CW)+1)$ in a WLAN. This, coupled with the fact that $E \equiv E_1$ (because every node hears every one else), leads to $p = p_1 = 2/(E(CW)+1)$. However, $E_1 \subset E$ in a multi-hop wireless network due to the accumulative interference outside the interference range. As a result, $p \neq p_1$.

To derive the attempt probability p, we define the following terms:

- $p_2 = \Pr\{E_1|$node i's carrier sense range is idle$\}$;
- $p_3 = \Pr\{$node i's carrier sense range is idle$|E\}$.

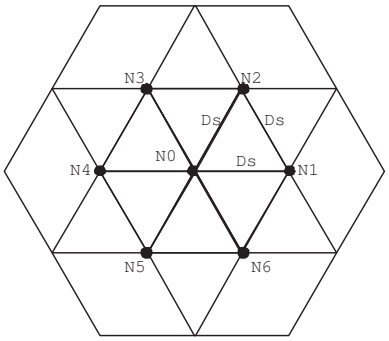


Figure 3.3: Definition of the silence range

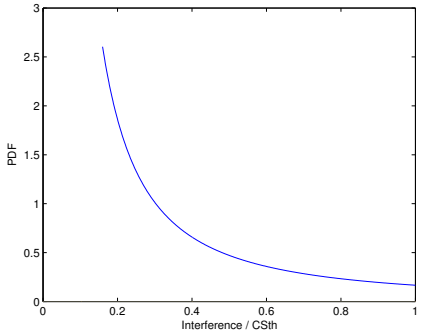


Figure 3.4: PDF of the interference given the existence of one interference node.

Table 3.2: Pr(Interference $< CS_{th}$).

# of nodes	P(I$< CS_{th}$)
2	0.918
3	0.762
4	0.560
5	0.357
6	0.192

Obviously,

$$p = p_1 \times p_2 \times p_3. \tag{3.19}$$

p_1 can still be expressed as $p_1 = 2/(E(CW)+1)$ in multi-hop wireless networks. In what follows, we derive p_2 and p_3.

Derivation of p_3

The probability p_3 can be straightforwardly derived, i.e.,

$$p_3 = \frac{(1-p)^{\lambda\pi {R_{cs}}^2}}{(1-p)^{\lambda\pi {R_i}^2}} = (1-p)^{\lambda\pi(R_{cs}^2 - R_i^2)}. \tag{3.20}$$

Derivation of p_2

By the definition of CS_{th} under the interference model, the probability p_2 is equivalent to Pr{ the aggregate interference contributed by transmissions outside the carrier sense range $\leq CS_{th}$ | node i's carrier sense range is idle }. It depends on the distribution of the interference nodes outside CS_{th}.

By **(A6)**, we consider only the accumulative interference contributed by six 1^{st} tier interference nodes. These nodes are located in the ring area that is centered at node i with inner radius R_{cs} and outer radius $2\times$

R_{cs}. Let I_{total} denote the total interference, I_j the interference received when there are j active interference nodes, and $\Pr(j)$ the probability that there exist j interference nodes, which can be calculated with p and the number of hosts in the ring area. Then the CDF of I_{total} can be expressed as

$$\Pr(I_{total} \leq x) = \sum_{j=0}^{6} \Pr(j) \times \Pr(I_j \leq x). \tag{3.21}$$

We consider first $\Pr(I_1 \leq x)$. If the interference node is located on the inner (outer) circle of the ring, then it contributes an interference level of CS_{th} ($CS_{th} \cdot (1/2)^{\theta}$). Moreover, $I_1 = k \cdot CS_{th}$ ($(1/2)^{\theta} \leq k \leq 1$) if and only if the interference node is located inside the ring area. Thus we have

$$\Pr(I_1 \leq k \cdot CS_{th}) = \frac{(k^{-\frac{1}{\theta}} + 2)(2 - k^{-\frac{1}{\theta}})}{3}, \tag{3.22}$$

and the PDF of I_1 can be expressed as

$$f(I_1) = \frac{2}{3} \times k^{-\frac{\theta+2}{\theta}}. \tag{3.23}$$

Because $I_{j+1} = I_j + I_1$ ($1 \leq j \leq 5$), we can obtain the PDF of I_{j+1} by performing the convolution of the PDF of I_j and that of I_1. With the PDF of I_j ($1 \leq j \leq 6$), p_2 can be expressed as

$$p_2 = \Pr(I_{total} < CS_{th}), \tag{3.24}$$

where the CDF of I_{total} is given in Eq. (3.21).

3.3.6 Derivation of Collision Probability and Collision Period

In a WLAN, collisions are only caused by *simultaneous* transmissions within the WLAN. (Note that simultaneous transmission cannot be avoided even with the use of physical carrier sense.) However, in a multi-hop wireless network, collisions can also occur when the accumulative interference contributed by *simultaneous* transmissions outside the interference range and/or *concurrent* transmissions outside the carrier sense range. To facilitate derivation of parameters of interest, $E(N_c)$ and $E(T_c)$, we classify collisions into three categories:

Type-1 : Collision caused by simultaneous transmissions *within* the interference range.

Type-2 : Collision caused by interference contributed by simultaneous transmissions *outside* the interference range but *within* the carrier sense range.

Type-3 : Collision caused by interference contributed by *concurrent* transmissions *outside* carrier sense

range.

Let $E(N_{c,1})$, $E(N_{c,2})$, and $E(N_{c,3})$ denote respectively the expected number of type-1, type-2, and type-3 collisions in a virtual transmission time, and $E(T_{c,1})$, $E(T_{c,2})$, and $E(T_{c,3})$ respectively the expected length of a type-1, type-2, and type-3 collision period in a virtual transmission time. In what follows, we derive these parameters.

Derivation of $E(N_{c,1}$ and $E(T_{c,1})$

Type-1 collisions are those considered in WLANs, and hence we can reuse the expressions given in Calí's model [14] directly

$$E(N_{c,1}) = \frac{1-(1-p)^M - Mp(1-p)^{M-1}}{P(\textit{successful transmission})}, \tag{3.25}$$

and

$$\begin{aligned} E(T_{c,1})=&\frac{t_{slot}}{1-(1-p)^M - Mp(1-p)^{M-1}} \times \\ &[\sum_{h=0}^{\infty}\{h \times [(1-pq^h)^M - (1-pq^{h-1})^M]\} - \frac{Mp(1-p)^{M-1}}{1-q}], \end{aligned} \tag{3.26}$$

except that the term $P(\textit{successful transmission})$ (i.e., the probability of a successful transmission within an interference range) in Eq. (3.25) has to be re-derived. We will derive this expression below.

Derivation of $E(N_{c,2})$ and $E(T_{c,2})$

Note that although a single transmission outside the interference range of a receiver will not corrupt the on-going transmission, accumulative interference from multiple *simultaneous* transmissions may. Let $I_{(R_i,R_{cs})}$ denote the accumulative interference contributed by simultaneous transmissions outside R_i but within R_{cs}. It is easy to see that when $\frac{P/R_{tx}^{\theta}}{I_{(R_i,R_{cs})}} \leq \beta$, collision occurs. Let p_4 be defined as the probability that accumulative interference contributed by simultaneous transmissions outside R_i but within R_{cs} leads to collision. Then

$$p_4 = \Pr(I_{(R_i,R_{cs})} \geq \frac{P}{\beta \cdot R_{tx}^{\theta}}). \tag{3.27}$$

The derivation of p_4 is similar to that of $1-p_2$ in Section 3.3.5, except that (i) the ring area is now with the inner radius R_i and the outer radius R_{cs}; and (ii) the number of simultaneously transmitting nodes could be more than 6, and hence the upper limit in the sum term of Eq. (3.21) can, in principle, go to infinity. (In our numerical examples, we set the upper limit to ℓ such that summing beyond ℓ does not further increase p_4.)

With the expression of p_4, $E(N_{c,2})$ and $E(T_{c,2})$ can now be expressed as

$$E(N_{c,2}) = \frac{p_4 \times Mp(1-p)^{M-1}}{P(\textit{successful transmission})}, \tag{3.28}$$

and

$$E(T_{c,2}) = \bar{m} = \frac{t_{slot}}{1-q}. \tag{3.29}$$

Note that $E(T_{c,2})$ is different from $E(T_{c,1})$. This is because $E(T_{c,1})$ is the same as $E[T_c]$ in a single-cell network, in which the collision period is the duration of the longest frame among multiple simultaneous transmissions that are involved in the collision. However, when a type-2 collision occurs, there is only one transmission within the interference range. Thus the collision period *from the perspective of nodes in this interference range* is one transmission duration (whose expected value is $\bar{m}$).

Derivation of $E(N_{c,3})$ and $E(T_{c,3})$

Consider an on-going transmission. If the transmission does not incur a collision due to simultaneous transmissions in the first slot, then in the course of transmission (the average duration of which is $\bar{m} = \frac{t_{slot}}{1-q}$), nodes within CS_{th} will keep silent, while the accumulative interference contributed by *concurrent* transmissions outside CS_{th} may still corrupt the transmission. By **(A6)**, we consider the accumulative interference, I_{total}, contributed by the six 1^{st}-tier interference nodes outside R_{cs}. Again, when $\frac{P/R_{tx}^{\theta}}{I_{total}} \leq \beta$, collision occurs. Thus the probability, p_5, that accumulative interference contributed by concurrent transmissions outside R_{cs} leads to collision can be expressed as

$$p_5 = \Pr(I_{total} \geq \frac{P}{\beta \cdot R_{tx}^{\theta}}). \tag{3.30}$$

Figure 3.5 illustrates how we calculate I_{total} as perceived by node N_0. The six 1_{st} tier interference nodes are located in the ring area centered at node N_0 and with inner (outer) radius of R_{cs} ($2R_{cs}$). The ring is divided into six sectors with equal proportion. In each sector there is one and only one interference node. Let I_i ($1 \leq i \leq 6$ denote the interference at the receiver D_0 contributed by the interference node in the ith sector. Then I_{total}=$\Sigma_{i=1}^{6} I_i$ denote the total interference. In practice, it is very difficult, if not impossible, to derive I_{total} because the six sectors are asymmetric irregular areas with respect to the receiver D_0. However, the interference, I_1, contributed by the interference node in the first sector dominates I_{total} since this interference node is closest to D_0. Hence, we approximate I_{total} by I_1, and only consider the shaded sector in Figure 3.5.

Let m denote the number of nodes in the shaded sector. Each of them transmits with a (likely different) attempt probability p. By Eq. (3.19), we need to derive p_1, p_2, and p_3. The probabilities p_1 and p_3 are the same for all nodes, while p_2 differs. This is because with respect to a node, say N_k, in the shaded sector, its 1^{st} tier interference node in the 4^{th} sector is fixed (i.e., N_0), and hence p_2 becomes the conditional probability that the total interference is less than CS_{th} given that N_0 transmits. Let $p_2^{(k)}$ denote p_2 with respect to node N_k in the shaded area, $I_0^{(k)}$ the interference contributed by N_0 and perceived at node N_k, and $I_{left}^{(k)}$ the remaining cumulative interference (perceived at node N_k) from the other five 1^{st}-tier interference nodes. Then $I_{left}^{(k)}$=I_{total}-$I_0^{(k)}$. Moreover we have

$$p_2^{(k)} = \Pr(I_{left}^{(k)} < (CS_{th} - I_0^{(k)})). \tag{3.31}$$

$I_{left}^{(k)}$ can be derived in the same manner as in Eq. (3.21), except that there are at most five 1^{st}-tier nodes and the entire ring area becomes $\frac{5}{6}$ of the ring. As shown in Eq. (3.31) if N_k is located at a position farther away from N_0, $p_2^{(k)}$ is larger. This implies that nodes which are more likely to induce collision at D_0 has a smaller attempt probability, thanks to the effect of physical carrier sense.

With I_{total} being approximated by I_1, p_5 is equal to the probability that the distance between the interference node, $N_{1,k}$ in the 1^{st} sector and D_0, denoted by $d_{N_{1,k},D_0}$, is smaller than the interference range R_i. Specifically, p_5 can be expressed as

$$\begin{aligned} p_5 = \Pr(d_{N_{1,k},D_0} \le R_i) &= \frac{\Sigma_{i=1}^{m} Ind(d_{N_{1,i},D_0} \le R_i) \times p^{(i)}}{\Sigma_{i=1}^{m} p^{(i)}} \\ &= \frac{\Sigma_{i=1}^{m} Ind(d_{N_{1,i},D_0} \le R_i) \times p_2^{(i)}}{\Sigma_{i=1}^{m} p_2^{(i)}} \end{aligned}$$

where $Ind(x)$ is an indicator function of x. When R_{cs} is smaller than $(1+\beta^{\frac{1}{\theta}}) \times R_{tx}$, p_5 is non-zero. Note that Eq. (3.32) is in descrete form and requires the knowledge of the number of nodes: m. When m is large we can use integral to calculate p_5.

With the expressions of p_4 (Eq. (3.27)) and p_5 (Eq. (3.32)), $E(N_{c,3})$ and $E(T_{c,3})$ can now be expressed as

$$E(N_{c,3}) = \frac{p_5 \times Mp(1-p)^{M-1} \times (1-p_4)}{P(\textit{successful transmission})}, \tag{3.32}$$

and

$$E(T_{c,3}) = \bar{m} = \frac{t_{slot}}{1-q}. \tag{3.33}$$

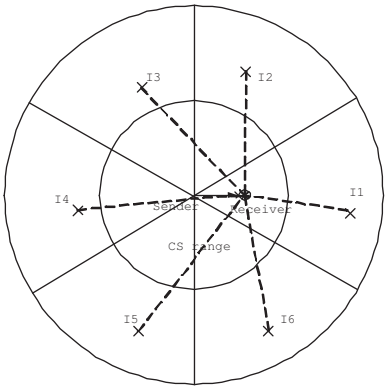


Figure 3.5: An example that illustrates how p_5 is calculated.

$$t_v = \sum_{i=1}^{3} E\left(N_{c,i}\right)\cdot\left(E\left(T_{c,i}\right) + DIFS + SIFS + ACK\right) + \sum_{1}^{3}\left(E\left(N_{c,i}\right)+1\right)\cdot E\left(I\right) + \bar{m} + SIFS + ACK + DIFS, \quad (3.35)$$

Derivation of P(*successful transmission*)

What is yet to be derived is P(*successful transmission*) A successful transmission occurs if and only if the following three conditions hold: 1) in the first slot of transmission, no type-1 collision occurs; 2) in the first slot, no type-2 collision occurs; 3) in the 2^{nd} to n^{th} slots, no type-3 collision occurs. P(*successful transmission*) is then expressed as

$$P(\textit{successful transmission}) = Mp(1-p)^{M-1}(1-p_4)(1-p_5). \quad (3.34)$$

By plugging Eqs. (3.15), (3.25) – (3.34) into Eq. (3.2), we obtain Eq. (3.35).

3.4 Discussion

In this section, we discuss several insights that are shed from the analytical model, which can be used as the guideline for setting MAC parameters.

The first observation is that while the average contention window size is set to $\frac{2}{p} - 1$ in WLANs, it can be set to a smaller value in multi-hop networks. This is because as revealed in Eq. (3.19), the attempt probability not only is a function of the contention window size (through p_1), but also takes into account of physical carrier sense (through p_2 and p_3). In order to ensure the attempt probability remains large enough to improve the system throughput, the contention window size should be set to a larger value than that in WLANs.

The second observation is that while the optimal throughput can be achieved with the use of a larger

value of p, it also leads to a larger collision probability. As a matter of fact, there exists a tradeoff between spatial reuse and collision caused by concurrent transmission. A larger value of p can increase the collision probability, but can also promote spatial reuse by by reducing the idle period in multi-hop networks. By comparing Eqs. (3.5) and (3.18), one can see that if the p value were set to the same value as that in WLANs, the idle period would be larger than its counterpart in WLANs. This is because an idle period may also occur in a "cell" when all the nodes in the cell are being silenced by sender nodes outside the cell but within their carrier sense ranges. This suggests that the p value should be set to a larger value in order to compensate this effect and promote spatial reuse.

3.5 Simulation Results

In this section, we carry out a simulation study to validate the analytical model, to verify the observation made on the model, and to study the impact of the contention window size and the carrier sensing range on the system performance. we first modify *J-sim* to incorporate that effect in order to improve the simulation fidelity.

3.5.1 Simulation Setup

The scenario used in the simulation study is as follows. There are a total of 480 nodes, half of which are senders and the other half are receivers. The 240 senders are uniformly distributed in a 900m×900m square area, and each receiver is located 80 meters away from its sendere, with R_{tx}=80m. Every source sends its packets directly to its intended receiver. Note that our primary focus is on how the various MAC parameters impact the system throughput. In order to eliminate the coupling effect of routing and MAC, we deliberately position the receiver one-hop away from its sender. We run simulation under two data rates: 2Mbps (SINR threshold = 4db) and 6Mbps (SINR threshold = 6db). For each data rate, the contention window size varies from 8 to 256 and R_{cs} varies from 140 meters to 220 meters. Every sender sends CBR packets of size 512 KB.

3.5.2 Model Validation

In section 3.3 we use p_2 and p_3 to characterize the effect of physical carrier sense on the attempt probability (p). The derivation of p_2 and p_3 is rigorous, except that we assume that the cumulative interference is mainly contributed by 1^{st} inteference nodes. However, when the contention window size becomes too small, p_2 and p_3 become smaller than they actually are in the simulation. As a result, p_2 and p_3 in the model

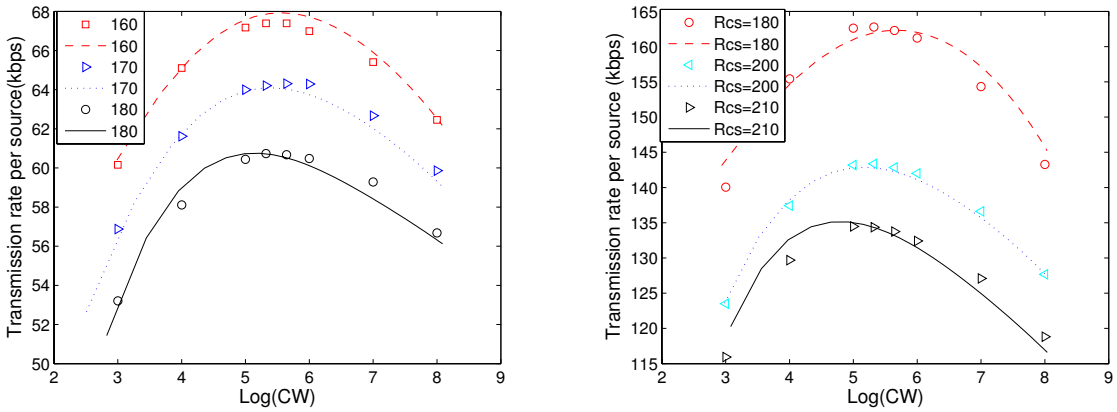


Figure 3.6: The throughput attained by each node. Noe that continuous curves represent theoretical calculations, while discrete dots represent simulation results.

exaggerate the effect of carrier sense and give a pessimistic value of p. To deal with this problem, we enforce the following rule: $p_2(p_3)=\phi$ if $p_2(p_3)< \phi$, where ϕ ranges between 0.1 and 0.5 according to the ratio of R_{cs} and R_i. Fig. 3.6 gives both the theoretical and simulation results of the per-node throughput for the cases of bandwidth=2M and bandwidth=6M under three different values of R_{cs}. The two types of results agree quite well, especially in the vicinity of the optimal points. One interesting observation of Fig. 3.6 is that all the curves are quite flat in the range of CW$\in$[32,64]. Within this range, the difference in the per-node throughput is less than 1%. This is in sharp contrast to the results obtained in WLANs where the system throughput varies more dramatically [14]. This is perhaps because when the contention window size increases, the rate of change in p is smaller in multi-hop wireless networks Recall that p_2 and p_3 increases as the contention window size increases, and p is proportional to p_2 and p_3. As a result, when the contention window size increases, the rate at which p decreases is slower than in WLANs.

Another interesting observation is that the contention window size that achieves the optimal per-node throughput in multi-hop networks is much smaller than that in WLANs. Based on Cal$\acute{i}$'s model, the optimal CW is 137 in the case of bandwidth = 2M and 107 in the case of bandwidth = 6M, respectively. In constrast, our simulation results show that in the six cases reported, the optimal contention window size is less than 64. There are two major reasons that may account for this observation. First, in multi-hop networks, R_{cs} is usually chosen to be larger than the interference range R_i in order to avoid the hidden terminal problems. With physical carrier sense, a pair of nodes within $R_{cs}(> R_i)$ of each other cannot transmit, unless they transmit simultaneously. A smaller contention window size (or equivalently a larger attempt probability) is thus desired to increase the probability of simultaneous transmission and promote spatial reuse. Second,

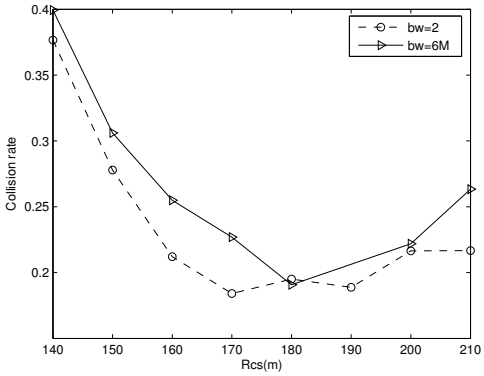


Figure 3.7: Collision ratio when the optimal per-node throughput is achieved

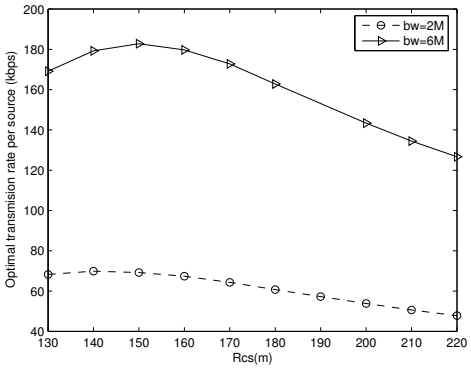
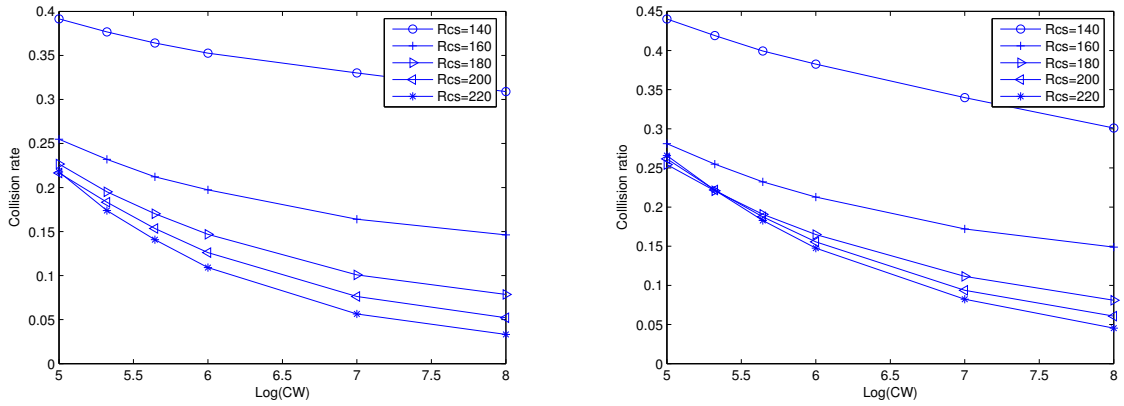


Figure 3.8: Optimal per-node throughput for given carrier sense range

recall that p_2 and p_3 characterize the effect of physical carrier sense in the attempt probability p in multi-hop networks, while p_1 is related to the contention window size through $p_1 = 2/(E(CW) + 1)$. If the contention window size were set to the same value as that in WLANs, the $p(= p_1 \times p_2 \times p_3)$ would be smaller than its counterpart in WLANs. This suggests use of a larger CW value to obtain the optimal per-node throughput.

3.5.3 Analysis of the Collision Rate

Fig. 3.7 gives the simulation results of the collision ratio, i.e., the number of collisions divided by the total number of transmission attempts, when the optimal per-node throughput is achieved for a give value of R_{cs} by adjusting CW. The collision ratio is higher than 20% when the throughput is maximized. This implies that one cannot achieve high throughput and low packet loss at the same time. An interesting finding is that the collision ratio first increases and then decreases with the increase in R_{cs}. This supports our assertion that when R_{cs} is large, moderately increasing the CW size can improve the throughput although it also leads to the increase in collisions. Fig. 3.8 shows the optimal throughput versus R_{cs}. The optimal value of R_{cs} is 140m in the case of bandwidth=2M and 150m in the case of bandwidth=6M. Both are much smaller than those determined by $CS_{th,1}$ in the pessimistic model (Section 3.2) (by 50m and 55m, respectively). On the other hand, Fig. 3.9 gives the collision ratio when the optimal throughput is achieved. One important observation is that when R_{cs} is set to be smaller than that determined by $CS_{th,1}$ (i.e., 190m and 205m, repectively) the collision ratio also increases. When R_{cs} is large, we can always increase the contention window size in order to control the collision ratio within a certain bound. When R_{cs} is smaller than the carrier sense range determined by $CS_{th,1}$, the corresponding collision ratio is always above 20% and cannot be well controlled by adjusting the CW size.



bw=2M bw=6M

Figure 3.9: collision ratio (bw=2,6)

3.6 Conclusion

In this chapter, we propose a new analytical model that extends Calí's model to multi-hop wireless networks and incorporates the effects of physical carrier sense, SINR, and collision caused by accumulative interference. We carefully identify the parameters that are needed to re-defined and re-derived. The model is then validated through simulation. Our model can be used to determine the optimal contention window size and the carrier sensing range, as well as to give insights on explaining the behavior of multi-hop wireless networks. For example, it shows that in multi-hop networks, physical carrier sense and contention window size jointly determine the attempt probability of a node. It also points out that in multi-hop networks the optimal per-node throughput can be achieved using a smaller contention window at the expense of a higher collision ratio.

Chapter 4

Coordinated Heavy-traffic Efficient Access in WLANs

4.1 Introduction

IEEE 802.11 Distributed Coordination Function (DCF) is a simple and robust MAC protocol that is dominantly used in WLANs nowadays. However, due to its uncoordinated and random access nature, DCF has considerable overhead coming from the the wasted airtime of random backoff as well as collisions. We define the *packet transmission airtime*, $T_s := T_{DIFS} + T_{packet} + T_{SIFS} + T_{ACK}$, where T_{DIFS} denotes the distributed inter-frame space, T_{packet} denotes the transmission time of a data packet, and T_{SIFS} is the short inter-frame space. The *overhead per successful medium access*, $T_{oh} = T_{idle} + T_{coll}$, where T_{idle} (T_{coll}) is the average backoff time (the average time cost for collisions) for each successful medium access. Then, the efficiency of DCF can be computed as follows,

$$\rho_{DCF} = \frac{T_s}{T_s + T_{oh}}. \tag{4.1}$$

It is well known that DCF has low efficiency in two scenarios [15][63]. The first such scenario is a network with high contention due to a large number of active hosts, where T_{oh} becomes very high. The other scenario is when most of packets have a tiny transmission time (i.e., tiny T_s), thus resulting in low ρ_{DCF}. Unfortunately, both scenarios are quite common in existing WLANs. Firstly, with the popularity of 802.11 networks, wireless channels have become increasingly crowded. Secondly, some widely used applications inherently involve small packets that use only small transmission time, *e.g.*, Voice Over IP. Further, with the deployment of new high-speed wireless technologies, like 802.11n, the transmission time of a packet is reduced proportionally. As a result, the MAC efficiency is further reduced [59].

For the downlink traffic, one possible solution is to perform centralized DCF-compatible scheduling at the AP. Several centralized scheduling schemes have been proposed in academia [54] [62] as well as in industry [4]. However, although scheduling schemes outperform DCF in many circumstance, they are difficult to apply to explicitly schedule *uplink* traffic (from client hosts to APs) due to the likely huge communication overhead

and vulnerability to missing control messages.

In the literature, many distributed MAC protocols have also been proposed to improve the efficiency of DCF. They can be largely classified into three categories. The first category tries to find the optimal backoff schemes that minimize the overhead caused by backoff waiting and collisions. This category includes a large body of work, such as *Enhanced DCF* [14], *Idle Sense* [26], *MFS* [33]), *Implicit Pipelining* [66] and *CM-CSMA* [60]. However, the efficiency of these protocols still drops significantly when the packet airtime decreases. In the second category, a batch of packets, instead of one as in DCF, are transmitted after each successful medium access. Proposals in this category include *802.11e EDCF* [49], *FCR* [37], *OAR* [51] and *CM* [64]. Suppose that on average k packets get transmitted for each successful media access, then the MAC efficiency is

$$\rho = \frac{k \times T_s}{k \times T_s + T_{oh}} = \frac{T_s}{T_s + T_{oh}/k}. \tag{4.2}$$

Comparing (4.1) and (4.2), it is clear that higher efficiency is achieved by amortizing the contention overhead over k packets. However, with large batch transmission, the average channel access delay, d_c, increases as well [53] [21]. d_c is defined as the time interval between the two instants when a client host, or say, client, starts backoff and when it sucessfully delivers the packet. When the WLAN has m active clients and each client occupies the channel for a duration of t_o, we have $E[d_c] \propto mt_o$. A large access delay will adversary impact on many applications that have some QoS requirements. Thus, the usage of these approaches is limited. Besides, it does not resolve the contention. The third category is similar to the second one, but without increasing the average channel access delay. *GAMA-PS* [32] is a typical approach in this category. It maintains a "transmission group" consisting of all the hosts that have data to transmit. It divides the transmission channel into a sequence of cycles. For each cycle, the hosts in the "transmission group" transmit packets sequentially. Although *GAMA-PS* is efficient, it suffers from several drawbacks. First, it relies on explicit control packets to maintain the "transmission group" in every cycle. Thus, it is very sensitive to errors or losses on these control packets. Second, *GAMA-PS* requires tight control on all nodes in the network. So it cannot coexist with DCF or other similar protocols, posing a significant problem vis-a-vis deployment.

We propose CHAIN, a Coordinated Heavy-traffic efficient Access scheme. CHAIN adapts its behavior automatically to the dynamic traffic load. When the network is idle, it operates in a way similar to DCF. But when the network becomes congested, the clients automatically adapt their behavior to reduce the contention, without explicit control messages from AP or a scheduler. As a consequence, the network throughput is significantly improved. CHAIN is designed mainly to improve the efficiency for uplink traffic. By improving the uplink efficiency, it also indirectly improves the down-link throughput. We show that CHAIN co-exists

fairly with DCF. It is also complementary to other existing designs to improve the down-link performance in WLAN.

Hereafter in the chapter, we consider only uplink traffic in WLANs. The rest of the chapter is organized as follows. Section 4.2 introduces the basic ideas and design details. We present an analytical model for CHAIN in Section 4.3. A thorough simulation study is presented in Section 4.4. Finally, Section 4.5 concludes the paper.

4.2 Protocol Design of CHAIN

This section first introduces the basic idea of CHAIN, and then describes CHAIN's design in details.

4.2.1 CHAIN Basic Design

In DCF, every host contends for its own medium access opportunity independently. We therefore address the following question: Can we build a loose bond among a group of hosts, so that when one host in the group gains a medium access opportunity, it can pass the opportunity to other hosts in the group without them having to go through contention? Thus, we arrive at the basic idea of CHAIN: *Each host has two types of transmission opportunities, namely spontaneous transmission and piggyback transmission. Spontaneous transmission* follows a modified DCF contention window-based media access scheme. Before sending a packet, the sender chooses a random number (in a different way from DCF) and starts transmission after waiting for that many idle slots. With *piggyback transmission*, each link l is given one (or multiple) *predecessor* links. After a successful transmission of its predecessor link (i.e., after overhearing an ACK of a packet sent on the predecessor link), instead of counting down its backoff timer, the sender of link l may immediately transmit after waiting an idle duration of SIFS. Since all other hosts should wait for at least a longer DIFS, the transmission on link l will not collide with other transmissions. Under heavy traffic, each host may always have a pending packet. Thus, one successful medium access will trigger a *chain* of collision-free piggyback transmissions from multiple hosts.

CHAIN has three major components: (1) A protocol to form and maintain the piggyback precedence relation among links; (2)The medium access protocol with piggyback transmissions; and (3) A modified exponential backoff scheme.

4.2.2 Piggyback Precedence Relation Maintenance

CHAIN forms and maintains a *legal* Piggyback Precedence Relation Set (denoted as R) on link set L in

Algorithm 1 PIGGYBACK PRECEDENCE RELATION MAINTENANCE (PM)

1: $i \leftarrow 1$
2: **while** nonstop **do**
3: $R_i \leftarrow \emptyset$
4: Sort clients $X[1, \cdots, m]$ such that $Th_{i-1}(X[k]) > Th_{i-1}(X[k+1])$, $k = 1, \cdots, m-1$
5: Choose j such that $Th_{i-1}(X[j]) > \delta_i > Th_{i-1}(X[j+1])$
6: **for** $k = 1, \cdots, j$ **do**
7: Add $(X[k\%j+1], X[k])$ to R_i
8: Rate($X[k]$'s ACK) $\leftarrow$ min{Rate($X[k\%j+1]$),Rate($X[k]$)}
9: **end for**
10: **for** $k = j+1, \cdots, m$ **do**
11: Add $(X[(k-j)\%(m-j)+j+1], X[k])$ to R_i
12: Rate($X[k]$'s ACK) $\leftarrow$ min{Rate($X[(k-j)\%(m-j)+j+1]$),Rate($X[k]$)}
13: **end for**
14: Broadcast R_i
15: $i \leftarrow i+1$
16: Sleep for P
17: **end while**

a network. Let $l_{ij} \in L$ denote the directional link from host i to host j. Link l_{nk} can piggyback on to link l_{ij}'s transmission if and only if $(l_{nk}, l_{ij}) \in R$. If $(l_{nk}, l_{ij}) \in R$, we call l_{nk} as l_{ij}'s *follower*, and call l_{ij} as l_{nk}'s *predecessor*. In a general wireless network, a piggyback precedence relation R is *legal* when it satisfies the following two conditions:

- Condition (1): any two links $(l_{nk}, l_{ij}) \in R$ only if host n either can overhear and decode host i's data transmission while knowing the corresponding ACK time, or can overhear and decode the ACK sent from host j to i.
- Condition (2): any link can have at most one follower, or all its follower links have the same sender. (Otherwise, it is possible that two or more hosts piggyback on to the same transmission which results in an unavoidable collision).

Since only uplink traffic is of interest in this paper, an uplink can be simply referred to by its sender (i.e., the host). For brevity, in the rest of the chapter, we say "client i R client j" instead of the longer version $l_{i \to AP}$ R $l_{j \to AP}$.

There may be many legal choices for the precedence relations on the set of uplinks, and we propose a simple protocol to generate such a relation R in a wireless LAN that may contains multiple APs. Our protocol lets each AP generate R locally without requiring a global conflict graph. This *Piggyback Precedence Relation Maintenance Algorithm (PM)* is shown in Algorithm 1. PM runs on each AP with m clients. R is updated periodically in every P seconds (we choose $P = 1$). R_i is the effective relation in the ith period, $Th_i(X[j])$ is the throughput (in unit of packets) of client $X[j]$ in the ith period, δ_i is a throughput threshold.

At the beginning of each period, the AP classifies all associated clients as "active" node or "inactive" node by comparing the traffic generated by the client in last period, $Th_{i-1}(X[j])$, to the threshold δ_i. Then, all active clients and all inactive clients form two separated rings. The final relation R_i is the union of these two rings. The AP adjusts the modulation rate of ACK to ensure R_i satisfies condition (1). Note that this is necessary because there is no guarantee that any two clients satisfy condition (1) due to the employment of multi-rate transmissions. Clearly R_i satisfies condition (2). Therefore, we conclude that PM can always generate a legal R_i. Note that one subtle overhead here is that the AP may send some ACKs at a lower rate. But we believe this will not be a big issue. The relation assignment is broadcast to all clients every P seconds. If a client does not receive a update for P seconds, the old assignment is expired and the client will disable piggyback transmission until it receives a new assignment.

4.2.3 MAC Protocol

The detailed MAC protocol of CHAIN is shown in Figure 4.1. The key processes are highlighted in italics and their pseudo code is given in Algorithms 2-4, where BT denotes the backoff timer, IC (*idle slot count*) denotes the number of idle slots counted till a packet transmission, CW denotes the contention window size, β is defined as the ratio of piggyback transmissions over spontaneous transmissions, and λ is a constant chosen less than and close to 1. The CHAIN MAC protocol is derived from DCF, and has the following three major modifications.

Firstly, the new media piggyback access opportunity is added. As explained earlier, piggyback allows a host to transmit immediately after overhearing the ACK of its predecessor's packet, without going through the contention phase. The follower client starts its piggyback transmission after a SIFS and therefore it can grab the medium before other clients, who wait for a longer DIFS.

Secondly, an innovative *debt system* is added. This is done to restrict piggyback clients from taking too much airtime, and is the key feature that allows co-existence of CHAIN with DCF. A *debt* (denoted as D) is the media access opportunity a host owes to the system as a whole when piggyback happens, and it needs to paid back later just like a real debt. *Debt* plays an important role in CHAIN. We explain it using a simple example illustrated in Figure 4.2. Consider a WLAN with one AP (node C) and two clients (nodes A, B), where link $B \rightarrow C$ is a follower of link $A \rightarrow C$. At time t_0, B has two packets in queue and A has one. Both of their debts are initialized to zero. Each node sets its BT to a random number in the range $[0, CW)$, and starts its packet transmission after observing BT idle slots. Obviously $IC \equiv BT$ in DCF. However, this is not true in CHAIN. Looking at Figure 4.2, at t_0, A chooses $BT_A = 3$ and B chooses $BT_B = 7$. After $DIFS + 3$ idle slots, node A's backoff timer fires and it starts transmission. Node B overhears the ACK

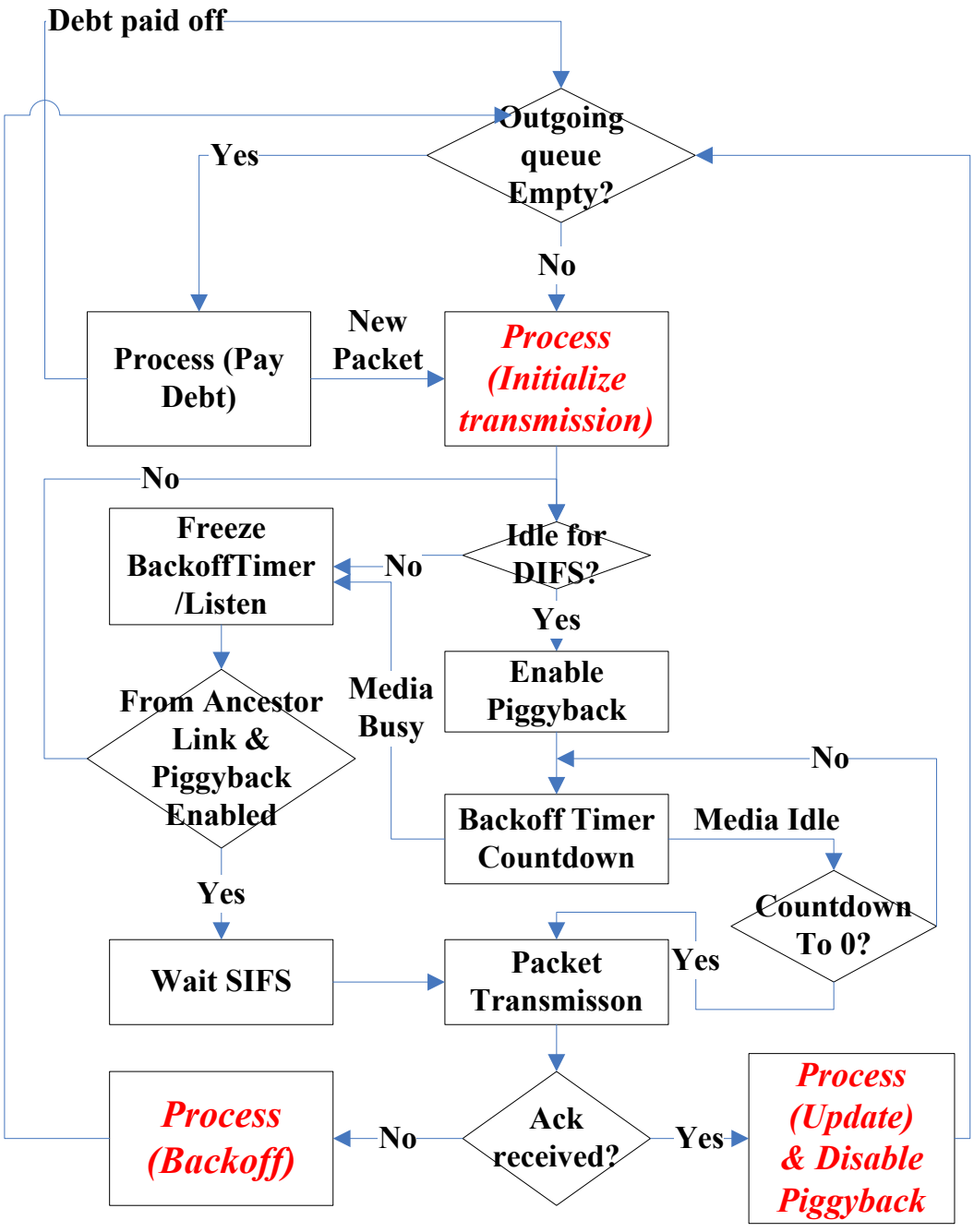


Figure 4.1: The flow chart of the MAC protocol of CHAIN

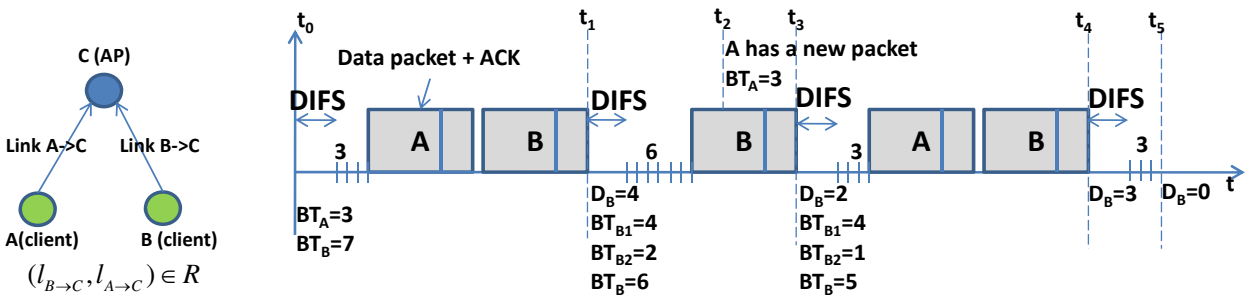


Figure 4.2: A simple example illustrating how the debt system works in the CHAIN protocol

Algorithm 2 Exponential Backoff

1: $D \leftarrow \lambda \times D + \beta \times E[CW]$
2: $CW \leftarrow CW \times 2$
3: $E[CW] \leftarrow$ average contention window size

sent from C to A and starts transmission later after only waiting a SIFS duration; thus $IC_B = 3 < BT_B$. Compared with its behavior in DCF, node B has gotten to transmit its packet $BT_B - IC_B = 4$ slots earlier. We define these 4 slots to be the *debt* of node B (i.e., D_B). At t_1 B receives its ACK from C and initializes the transmission of its second packet with D_B=4. Now instead of choosing BT_B uniformly within range $[0, CW)$, BT_B is chosen uniformly within $[0, CW + D_B)$, as described in Algorithm 2. $BT_{B2} = 2$ can be viewed as the balance B plans to pay. At time t_3, D_B decreases from 4 to 2, as described in Algorithm 3. Then BT_B is set to 5; however, this time B piggybacks on to A's transmission again and owes the system $BT_{B1} - IC_B = 1$ more idle slot. Hence D_B increases to 3 at time t_4. Now B has no packet to transmit, and as the flow chart in Figure 4.1 suggests, it starts the process "pay debt": After observing the channel being idle for a period of DIFS, D_B is decremented as long as the channel is sensed idle. Finally at time t_5, D_B decreases to zero.

The *debt* system serves two purposes. First, it prevents a piggyback client from taking too much medium time. This is the key reason why a WLAN running CHAIN can co-exist with a WLAN running DCF. Second, the "amortization" scheme helps to reduce the collision ratio under heavy traffic. We will elaborate more about this in Section 4.3.

Finally, we add an additional rule to prevent a transmission chain from lasting forever. A chain of transmission automatically stops when the current sender has no follower, or the follower has no packet to

Algorithm 3 Initialize Transmission

1: $r \leftarrow random(0, 1]$
2: $BT_1 \leftarrow r \times CW$, $BT_2 \leftarrow \lambda \times r \times D$
3: $BT \leftarrow BT_1 + BT_2$

Algorithm 4 Update

1: $D \leftarrow \lambda \times D + BT_1 - IC$
2: update piggyback ratio β
3: update average contention window size $E[CW]$

transmit. Because clients of the same AP belong to a ring under piggyback precedence relation, once the chain of transmissions starts it may grab the medium forever under saturated traffic. We therefore restrict each client to only transmitting once in any chain of transmissions. We have two ways to enforce this rule. The first one is implicit. As shown in Figure 4.1, a client's piggyback right is disabled after a successful transmission and re-enabled only after it observes the channel being idle for over an interval of length DIFS. Recall that within a chain of transmissions the media cannot be idle for a DIFS duration. If this client can carrier-sense all other clients within the ring, then this scheme guarantees that it cannot transmit more than once in a single chain of transmissions. However, although very rare, clients associated with the same AP can be hidden terminals. In this case, we can use an explicit solution: Each AP keeps track of the chain of transmissions. When it observes that the chain is too long, the AP can always use one reserved bit in the MAC header of the ACK it sends out to forbid the follower to piggyback. Usually, however, the implicit solution is enough, while the explicit solution is always there for safety.

4.2.4 Modified Exponential Backoff

Algorithm 4 describes the backoff scheme in CHAIN. We can see that with an increasing β, a client increases its debt more aggressively. This is to ensure the fairness among multiple piggyback rings. A client with more piggyback transmissions will see less collisions. This is easy to understand because only the spontaneous transmissions are going through the regular contention process and may result in collisions, and thus exponentially enlarge the contention window. Therefore, clients with a larger piggyback ratio, β, will have an advantage in contention over clients with a smaller β. Algorithm 4 compensates this by making clients enlarge their debts proportionally to β when they detect a collision.

4.2.5 The Downlink Coordination

CHAIN groups the client nodes together to compete for the medium access opportunities. While CHAIN helps in improving the uplink performance by favoring clients, it may hurt the downlink performance since the access point gets less transmission opportunities. Plus, most applications are either symmetric or require more downlink than uplink capacity, which means that an access point needs to transmit far more data than each individual client does, with the ratio growing as more clients are connected to the access point.

Traditional Hub-like access points are unable to accommodate this because they treat the access point as just another client contending for network access.

Fortunately, there exist schemes in the 802.11 standard which enable a station to transmit multiple frames consecutively within a burst after it gains the channel, such as Transmission Opportunity (TXOP) in the IEEE 802.11e standard [49] and frame aggregation mechanisms in the latest 802.11n standard [55]. With the access point adopting such schemes, capacity is shared equally between uplink and downlink transmissions when both need to be sent, ensuring optimum performance. In this way, CHAIN no longer exaggerates the unfairness between uplinks and downlinks. As a matter of fact, CHAIN should always be adopted along with a scheme enabling bursty frame transmission by the access point. In Section 4.4 where we evaluate CHAIN in simulation, we have implemented such bursty transmission mechanism on the access points.

4.3 Analytical Evaluation

In this section we first provide an analytical model to evaluate CHAIN's performance under an ideal scenario, from which we prove rigorously that CHAIN has several nice properties. Then we relax the constraints of the ideal case and discuss CHAIN's behavior under realistic scenarios with co-existence of other WLANs, mobility, channel errors, and idle clients.

4.3.1 Analytical Model

We study a single WLAN consisting of one AP and m clients. Each client has saturated uplink traffic. There are no errors on the transmission channel, and any packet losses are only caused by collisions among the m clients.

The CHAIN protocol coordinates the clients to form a piggyback ring of size m. The wireless media is divided into a sequence of transmission cycles by CHAIN. Each cycle begins with a contention/backoff period followed by a transmission period. The transmission period is either a successful delivery of m packets (one packet from each client), or a single packet collision. We use $T(n)$ to denote the nth transmission cycle, saying that $T(n)$ succeeds (fails) if its transmission period succeeds (fails). $I(n)$ is the number of idle slots of $T(n)$'s backoff period. $D_i(n)$, $B_i(n)$, $W_i(n)$ and $r_i(n) \in (0, 1)$ are the debt, the backoff interval, the contention window size and the random variable to set the backoff interval of client i at the beginning of $T(n)$. Client i must wait $B_i(n)$ idle slots to start a spontaneous transmission during $T(n)$. According to the CHAIN MAC protocol, $D_i(n+1)$ and $B_i(n+1)$ are set at the end of each $T(n)$ for all clients. If $T(n)$

succeeds, then every client i sets

$$D_i(n+1) = B_i(n) + \lambda[1 - r_i(n)]D_i(n) - I(n), \tag{4.3}$$

$$W_i(n+1) = CW_{min}, \tag{4.4}$$

$$B_i(n+1) = \lambda r_i(n+1)D_i(n) + r_i(n+1)W_i(n), \tag{4.5}$$

where $I(n) = \min_{j=1}^{m} B_j(n)$. If $T(n)$ fails, then for all clients i that encounter this collision, set

$$D_i(n+1) = \lambda D_i(n) + (m-1)E[W_i], \tag{4.6}$$

$$W_i(n+1) = \max\{2W_i(n), CW_{max}\}, \tag{4.7}$$

$$B_i(n+1) = \lambda r_i(n+1)D_i(n) + r_i(n+1)W_i(n+1). \tag{4.8}$$

For other clients i which do not encounter this collision, their debts remain unchanged and their backoff interval decreases by $I(n)$, i.e.,

$$D_i(n+1) = D_i(n), \tag{4.9}$$

$$B_i(n+1) = B_i(n) - I(n). \tag{4.10}$$

Lemma 1. *If $0 < \lambda < 1$, the debt of any client i is bounded, i.e., there exists a constant C such that*

$$\limsup_{n\to\infty} D_i(n) < C. \tag{4.11}$$

Proof. At the beginning of each $T(n+1)$, $D_i(n+1)$ can be updated in three ways as defined in (4.3), (4.6) and (4.9). If using (4.3), by substituting $B_i(n)$ in (4.3) with the expression given in (4.5), it can be rewritten as

$$D_i(n+1) = r_i(n)W_i(n) + \lambda D_i(n) - I(n) \leq \lambda D_i(n) + CW_{max}. \tag{4.12}$$

If $D_i(n+1)$ is updated using (4.6) or (4.9), we have

$$D_i(n+1) \leq \lambda D_i(n) + (m-1)CW_{max}, \tag{4.13}$$

or

$$D_i(n+1) = D_i(n). \tag{4.14}$$

Combining (4.12), (4.13) and (4.14) yields

$$D_i(n+1) \leq \max\{\lambda D_i(n) + (m-1)CW_{max}, D_i(n)\}.$$

Because $\lambda < 1$ and $D_i(0) = 0$, it is obvious that for any $(i,\ n)$, $D_i(n) \leq \frac{(m-1)CW_{max}}{1-\lambda}$. In any case,

$$\limsup_{n\to\infty} D_i(n) \leq \frac{(m-1)CW_{max}}{1-\lambda} = C. \tag{4.15}$$

□

Since $D_i(n)$ is bounded, $B_i(n)$ and $I(n)$ are bounded too. Therefore the expected values of $I(n)$, $D_i(n)$ and $B_i(n)$ exist, which we denote by $E[I]$, $E[D_i]$ and $E[B_i]$ respectively. Define $E[D] = \sum_{i=1}^{m} E[D_i]/m$, and $E[B] = \sum_{i=1}^{m} E[B_i]/m$.

Proposition 1. *The average backoff interval $E[I]$ of every T, when $\lambda > 1 - 2/(m(E[W]+1)$, is bounded by*

$$\frac{E[W]-2}{2} \leq E[I] \leq \frac{E[W]+3}{2}. \tag{4.16}$$

In order to model the system, we assume that for each new transmission attempt of client i, it chooses a backoff interval sampled from a geometric distribution with parameter p_i, where $p_i = 1/(E[B_i]+1)$, and

$$E[B_i] = \frac{E[W_i] + E[D_i] - 1}{2}. \tag{4.17}$$

This assumption was proposed and justified in [15], and widely adopted by other researchers ever since. Under this assumption, we replace (4.5) and (4.8) with

$$B_i(n) \sim Geom(p_i). \tag{4.18}$$

Note that the geometric distribution is memoryless; hence (4.10) can be replaced by (4.18) too. The reason is as follows. If $T(n)$ fails and client i does not encounter this collision, then we must have $B_i(n) \geq I(n)$,

and

$$P[B_i(n+1) \geq k] = P[B_i(n) \geq I(n) + k | B_i(n) \geq I(n)]$$
$$= P[B_i(n) \geq k], \tag{4.19}$$

i.e., $B_i(n+1)$ is also a geometric random variable.

Because $I(n) = \min_{j=1}^{m} B_j(n)$, it is straightforward that $I(n) \sim Geom(p')$, where

$$p' = 1 - \prod_{j=1}^{m}(1 - p_i). \tag{4.20}$$

The exact value of $E[I]$ depends on every $E[B_i]$ and is difficult to derive since it is possible that $E[B_i] \neq E[B_j]$ when $i \neq j$. However, according to [38], given a set of geometric random variables $\{X_i : i = 1, ..., m\}$ and a set of exponential random variables $\{Y_i : i = 1, ..., m\}$, $E[\min(X_i)]$ can be bounded by $E[\min(Y_i)] = Avg(E[Y_i])/m$ as long as $E[X_i] = E[Y_i]$ for $i = 1, ..., n$. Therefore, we have

$$\frac{E[B]}{m} \leq E[I] \leq \frac{E[B]}{m} + 1. \tag{4.21}$$

Now we only need $E[B]$ to obtain $E[I]$.

In order to derive $E[B]$, let us consider two successful transmission cycles $T(n-1)$ and $T(n+k)$, with $T(n+k)$ being the first successful transmission cycle after $T(n-1)$. Letting $N_c = k$ denote the number of failure cycles between $T(n-1)$ and $T(n+k)$, and N_c' denote the number of clients that encounter collisions between $T(n-1)$ and $T(n+k)$, statistically

$$E[N_c] = \frac{1 - \prod_{i=1}^{m}(1 - p_i)}{\sum_{i=1}^{m} p_i \prod_{j \neq i}(1 - p_j)} - 1, \tag{4.22}$$

$$E[N_c'] = \sum_{i=1}^{m} p_i \prod_{j \neq i}(1 - p_j). \tag{4.23}$$

Note that if one client encounters two collisions, it counts as twice when calculating N_c'. Suppose the expected number of collisions that client i encounters between $T(n-1)$ and $T(n+k)$ is Nc_i'. Note that $\sum_{i=1}^{m} Nc_i' = E[N_c']$. If collisions do not occur to client i, $D_i(n+k) = D_i(n)$; otherwise $D_i(n+k)$ is updated

$$E[D] \geq \frac{E[W]+E[D]-1}{2} + \frac{\lambda}{2}(E[D] + \frac{(m-1)}{m}E[N_c]E[W]) - (1+E[N_c'])(\frac{E[W]+E[D]-1}{2m}+1) \tag{4.28}$$

$$E[D] \leq \frac{E[W]+E[D]-1}{2} + \frac{\lambda}{2}(E[D] + \frac{(m-1)}{m}E[N_c]E[W]) - (1+E[N_c'])\frac{E[W]+E[D]-1}{2m} \tag{4.29}$$

according to (4.6). Therefore at the beginning of $T(n+k)$ we have

$$\begin{aligned} E[D_i(n+k)] = & E[D_i(n)] - (1-\lambda)E[Nc_i']E[D_i(n)] \\ & +(m-1)E[Nc_i']E[W_i]. \end{aligned} \tag{4.24}$$

This implies that

$$\begin{aligned} E[D(n+k)] = & E[D(n)] - (1-\lambda)\frac{E[N_c']}{m}E[D(n)] \\ & +(m-1)\frac{E[N_c']}{m}E[W]. \end{aligned} \tag{4.25}$$

Then at the beginning of $T(n+k+1)$ we have

$$\begin{aligned} E[D_i(n+k+1)] = & E[B_i(n+k)] + \frac{\lambda}{2}E[D_i(n+k)] \\ & -(1+E[N_c])E[I], \end{aligned} \tag{4.26}$$

which implies that

$$\begin{aligned} E[D(n+k+1)] = & E[B(n+k)] + \frac{\lambda}{2}E[D(n+k)] \\ & -(1+E[N_c])E[I]. \end{aligned} \tag{4.27}$$

Because both $T(n)$ and $T(n+k+1)$ are cycles following successful cycles, when $n \to \infty$ and the system is stable, we should have $E[D(n+k+1)] = E[D(n)]$. Using this equality and substituting (4.16), (4.17) and (4.25) into (4.27) yields the two inequalities: (4.28) and (4.29). If we choose $\lambda = 1 - \frac{1}{m\theta}$, then we derive the lower bound of $E[D]$ from (4.28),

$$E[D] \geq \frac{(1+E[N_c'])}{(1+E[N_c]+\theta)}(m-1)(E[W]-2) + o(1), \tag{4.30}$$

and the upper bound of $E[D]$ from (4.29),

$$E[D] \leq \frac{(1+E[N_c'])}{(1+E[N_c]+\theta)}(m-1)E[W]+o(1). \tag{4.31}$$

From (4.22) and (4.23) we obtain approximations $E[N_c] \approx \frac{m}{2(E[B]+1)}+o(\frac{m}{(E[B]+1)})$ and $E[N_c'] \approx \frac{m}{(E[B]+1)}+o(\frac{m}{(E[B]+1)})$. If we choose small $\theta < \frac{m}{(E[B]+1)}$, then (4.30) yields

$$E[D] \geq (m-1)(E[W]-2) \tag{4.32}$$

which implies

$$E[B] \geq m(E[W]-2)/2. \tag{4.33}$$

(4.33) yields $E[N_c] \leq \frac{1}{(E[W]-2)}+o(\frac{1}{(E[W]-2)})$, and furthermore $\frac{(1+E[N_c'])}{(1+E[N_c]+\theta)} \leq \frac{E[W]-1}{E[W]-2}$. Then (4.31) becomes

$$E[D] \leq (m-1)(E[W]+1), \tag{4.34}$$

which implies

$$E[B] \leq m(E[W]+1)/2. \tag{4.35}$$

Finally we obtain (4.16) by substituting (4.33) and (4.35) into (4.21). Proposition 1 is thus proved.

Proposition 2. *The probability that one transmission cycle fails remains bounded regardless of m. The packet loss ratio decreases as m increases.*

Proof. The probability that one transmission cycle fails is $E[N_c]/(1+E[N_c])$. We have shown that $E[N_c]$ does not increase with m in the proof of Proposition 1. The packet loss ratio is $E[N_c']/[m(1+E[N_c'])]$, which is proportional to $1/m$. □

We have thus provided an analysis of CHAIN under the basic scenario. Next, in the remainder of this section, we study CHAIN's behavior under more realistic scenarios by relaxing assumptions made for the ideal case one by one.

4.3.2 More Causes for Packet Losses

Now we relax the assumption that the intra-WLAN collision is the only cause of packet losses. A packet loss may be due to channel error or collisions with clients of other WLANs. Suppose that a $T(n)$ that does not encounter intra-WLAN collision has probability q to fail due to channel error or by colliding with a

transmission from outside its own WLAN. Hence the values of N_c and N_c' that we derive in Section 4.3.1 no longer apply.

Proposition 3. *Proposition 1 holds regardless of q.*

Proof. We prove this by showing that $\frac{1+N_c'}{1+N_c}$ is independent of q. Denote by p_1 the proportion of T's that do not encounter intra-WLAN collisions, and by p_2 the proportion of packets that do not collide with packets from another client inside the WLAN. Then we have

$$N_c = \frac{1}{(1-p_1)(1-q)}, \quad N_c' = \frac{1}{(1-p_2)(1-q)},$$

which yields

$$\frac{1+N_c'}{1+N_c} = \frac{1-p_2}{1-p_1}.$$

□

4.3.3 Existence of Clients with Unsaturated Traffic

Now we relax the assumption that every client has saturated traffic. Recall that according to the CHAIN protocol design, after the transmission of client i, if its follower has no data packet to transmit, the chain of transmissions in this cycle stops. Therefore a client i with small traffic load staying in the piggyback relation ring where other members have saturated traffic, may reduce the number of packets delivered per cycle, which decreases the efficiency of CHAIN. Moreover the follower of this client may benefit less from CHAIN. The smaller the traffic load of client i, the worse the situation is. However, our piggyback precedence relation maintenance (PM) scheme as defined in Algorithm 1 only allows clients which are busy enough to join the same ring, it therefore prevents any severe efficiency drop in CHAIN due to relative idle clients. Another design detail targeted at this scenario is that in Algorithm 4 we use the client's piggyback ratio β instead of $m-1$ as given in (4.8), because when $\beta \neq m-1$, the client benefits as though it were in a piggyback ring of size $\beta+1$ where every member always has data to transmit.

4.4 Simulation Evaluation

In this section, we use the J-SIM [31] simulator to evaluate CHAIN. We first compare CHAIN with 802.11 DCF, Enhanced DCF with optimal CW, and 802.11e in a single-AP WLAN. The simulation results also verify the analytical results in Section 4.3. Then, we look into the fairness issue when CHAIN co-exists with DCF. Later in the last two subsections, we study CHAIN's performance with various traffic patterns. Table 4.1 summarizes the configuration parameters used in our simulation, which follow the IEEE 802.11g

Table 4.1: WLAN Configuration

Parameter	**Value**
SIFS	10μsec
DIFS	28μsec
Slot time	9μsec
Phy Preamble	16μsec
CW_{min}	16
CW_{max}	1024
Bit rate	54Mbps

standard [6]. Unless otherwise stated, the default packet size is 1400 Bytes, all clients always have packets to send, and there is no down-link traffic in our simulations. Throughout this section, we call the piggyback precedence relation built by the CHAIN protocol a ring (rings) for short.

4.4.1 Basic Performance Evaluation

We compare the performance of four MAC protocols in a single-AP WLAN with m clients: 802.11 DCF, CHAIN, Enhanced DCF using optimal contention window (CW), and 802.11e EDCF with TxOP=$1.5ms$. The value of m ranges from 10 to 50. Figures 4.3 and 4.4 compare the system throughput when the packet size is 400B and 1400B, respectively. We see that when packet size is 400 Bytes, CHAIN has $70\% \sim 112\%$ throughput gain over DCF, and $65\% \sim 76\%$ throughput gain over enhanced DCF, when n ranges from 10 to 50. When packet size is 1400 Bytes, the throughput gain is lower since the efficiency of DCF increases with packet size, but we still achieve at least 40% throughput gain over enhanced DCF. The throughput gain of CHAIN over EDCF increases with m, because EDCF does not reduce contention as CHAIN does. Probability p_c is defined as the probability that one medium access fails due to collision. Figure 4.5 indicates that CHAIN has much lower p_c. Figure 4.6 verifies Proposition 1 that we proved in Section 4.3: $E[I]$ is bounded regardless of m. This property promises that CHAIN can co-exist with DCF fairly.

4.4.2 Multiple-AP Scenario

In this subsection, we consider a network with multiple APs but all within a single contention domain. There are twelve clients, and we test four cases: (1) clients use DCF and associate with one AP; (2) clients use CHAIN and associate with one AP, hence we have a ring of size twelve; (3) clients use CHAIN and associated with two APs, with each AP serving six clients; (4) clients use CHAIN and associate with four APs, with each AP serving 3 clients. Table 4.2 lists the system throughput and the packet loss ratio in the four cases. The system throughput decreases when we have more rings of smaller size. This is because of two reasons.

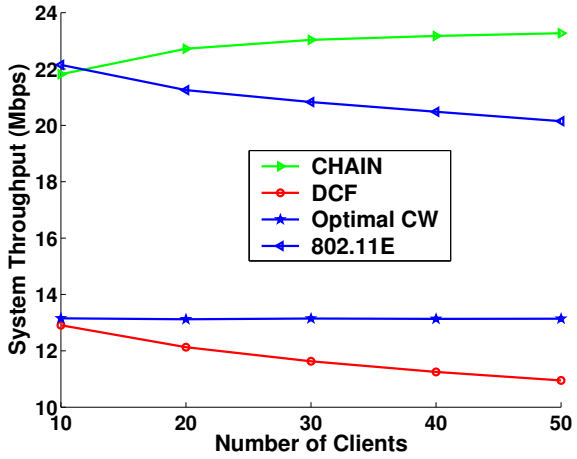


Figure 4.3: System throughput (packet size=400B)

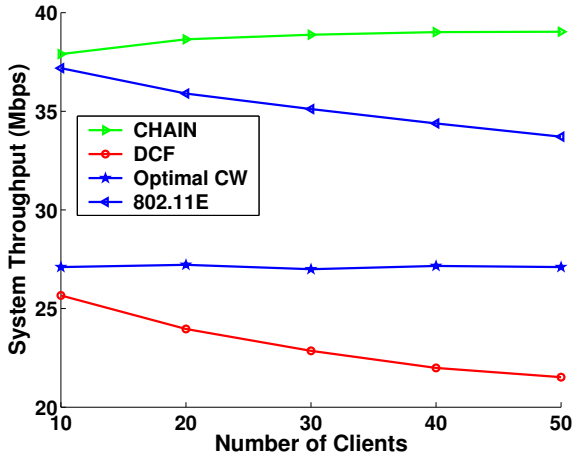


Figure 4.4: System throughput (packet size=1400B)

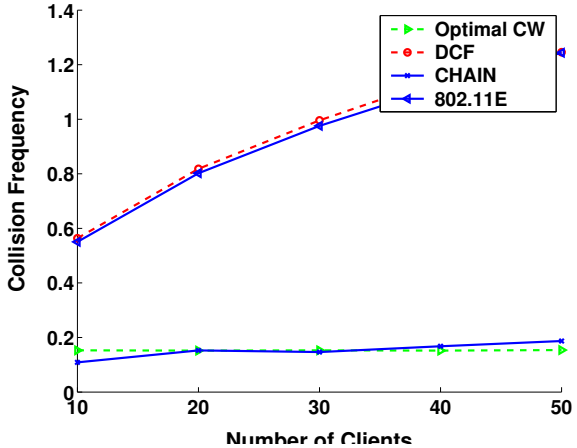


Figure 4.5: Media access collision frequency

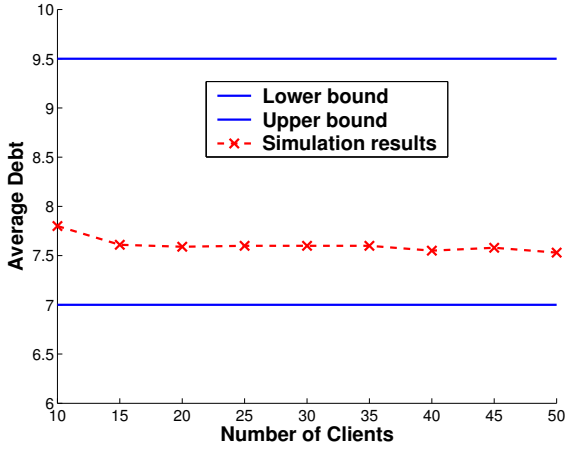


Figure 4.6: The average number of idle slots before each media access

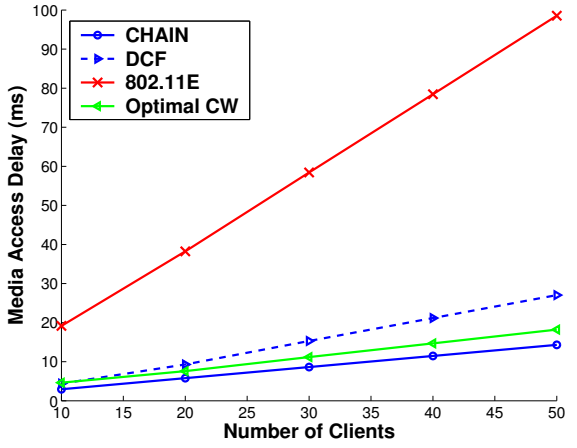


Figure 4.7: Media access delay

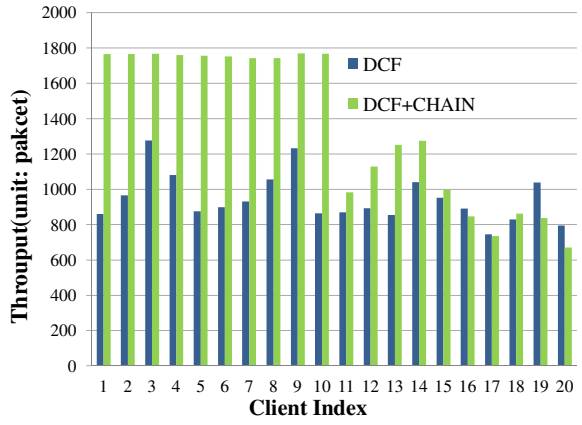


Figure 4.8: The comparison of per-client throughput in pure DCF WLANs v.s. DCF+CHAIN WLANs

Table 4.2: Comparison of Multiple-AP Scenarios

	DCF	1 Ring	2 Rings	4 Rings
Packet loss ratio	38.9%	1.1%	4.1%	10.3%
System throughput (Mbps)	21.65	32.13	36.39	38.08

First, the length of a transmission chain is limited by the size of the ring; hence the efficiency of CHAIN increases with the size of the ring. Second, CHAIN can keep intra-ring collision ratio low but cannot reduce inter-ring collision ratio. Nonetheless, the collision ratio when there are four rings is still much lower than DCF. This is easy to understand. In DCF the contention happens among all clients. Thus, there are 12 contending clients. In CHAIN, the contention happens at ring level and there are only 4 contending rings.

4.4.3 Fairness

We evaluate fairness of CHAIN in a network with 24 clients. Clients 1-4 are associated with AP1, clients 5-12 are associated with AP2, and clients 13-24 are associated with AP3. All the three APs runs CHAIN, and we have three rings of sizes 4, 8, and 12 respectively. The per-client throughput is : 1.35Mbps for clients 1-4, 1.51Mbps for clients 5-12, and 1.55Mbps for clients 13-24, respectively. We can see that CHAIN achieves fairness very well among rings with different sizes. The throughput of a client of AP1 is only 13% less than that of a client of AP3, even though their ring sizes differ by a factor of three. It is the binary backoff scheme that causes this slight unfairness. Recall that the medium access probability of one ring is inversely proportional to $E[CW]$. Clients from small rings are more likely to encounter collisions and backoff, than clients from large rings; hence have larger $E[CW]$ and a smaller share of the medium.

4.4.4 Co-existence with 802.11

Here we study CHAIN's behavior when it co-exists with DCF. We run the following simulation in a two AP network. AP1 has 10 clients (clients 11-20) running DCF, and AP2 has 10 clients which run either DCF or CHAIN with a ring of size ten. Figure 4.8 compares the throughput of each client when AP2 chooses DCF and when it chooses CHAIN. It is obvious that when CHAIN competes with DCF, clients of AP2 achieve much higher throughput. However, when AP2 switches to CHAIN from DCF, the average throughput of clients in AP1 actually slightly increases by 7%. This is because when AP2 chooses CHAIN, its ten clients act as one contention entity instead of 10 entities; hence the collision probability is reduced and clients of AP1 benefit from it despite the fact that clients of AP2 grab more medium access opportunity. Therefore we conclude that a CHAIN system can co-exist with a DCF system without hurting them badly even under

Table 4.3: Comparison of the performance of five different MAC settings

MAC	Relative throughput	Delay (ms)	1-$\frac{\min(Th_1,...,Th_6)}{\max(Th_1,...,Th_6)}$
DCF	1	247.5	31.7%
CHAIN,R_1	1.23	45.9	23.2%
CHAIN,R_2	1.16	32.1	1.6%
CHAIN, δ_1	1.22	48.7	2.6%
CHAIN, δ_2	1.22	52.3	0.3%

reasonably high contention level.

4.4.5 Clients with Unsaturated Traffic

In this section we study CHAIN's performance under a more realistic traffic pattern. The network in the simulation has one AP with twelve clients. Clients 1-6 have bulk traffic. Clients 7-12 have smaller packet arrival rate, and each has 120 packets to transmit per second. Their packet arrival is a Poisson process. The six of them together only require one fifth of the WLAN capacity. We compare the performance of three MAC protocols: DCF, *static* CHAIN (CHAIN using fixed piggyback precedence relation assignment instead of PM) and the *standard* CHAIN MAC. We generate two legal piggyback precedence relation: R_1 = (1,2,3,4,5,6,7,8,9,10,11,12), R_2 = (1,7,2,8,3,9,4,10,5,11,6,12). Then we run simulations using five different MAC settings: DCF, *static* CHAIN using R_1, *static* CHAIN using R_2, CHAIN with threshold δ_1 <120 packets/sec, CHAIN with threshold δ_2 > 120 packets/sec. Table 4.3 compares the average throughput of clients 1-6 and the queueing delays of packets of clients 7-12. We see that CHAIN always outperforms DCF. R_2 has less throughput gain than the other three since it is most likely for an idle client to break a chain of transmissions and make CHAIN less efficient. However, R_2 yields the least queueing delay since the chance for clients 7-12 to piggyback is higher in R_2. This property can be very helpful under certain circumstances, for example to support VoIP. If a VoIP user always piggybacks from a client who transmits often, its access latency can be significantly reduced. Another observation is that when R_1 is used, client 1 has larger throughput than client 6 since client 6 piggybacks to a less active predecessor. However, PM has handled such unfairness well by updating the piggyback precedence relation periodically.

4.4.6 The Choice of Throughput Threshold δ

We evaluate the impact of setting threshold δ. We use the same network topology as in Section 4.4.5. Define Λ as the maximum system throughput when there is only one client with saturated traffic in the WLAN.

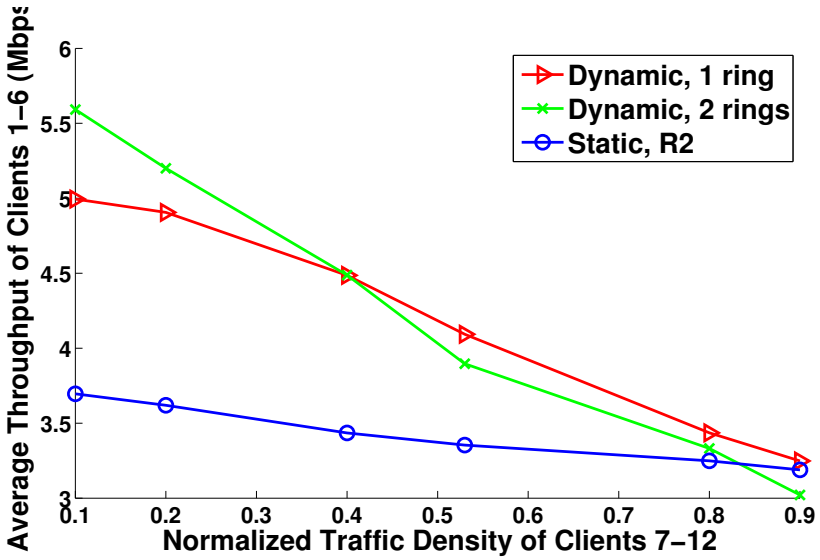


Figure 4.9: The comparison of three different CHAIN settings

The uplink traffic density of each client is measured quantitatively in units of $\Lambda/12$. Through the simulation, the normalized traffic density of clients 1-6 is always 2, and that of clients 7-12, which are all of the same value, varies between 0.1 and 0.9. We compare the performance of three MAC settings. The first is a static CHAIN using $R2$. The second is CHAIN running PM with proper δ settings such that clients 7-12 always join the busy ring; hence there is only one ring. The third is CHAIN running PM with clients 7-12 always forming a separate ring. Clients 7-12 are able to deliver all their packets under all of the three MAC settings. Figure 4.9 compares the average throughputs of clients 1-6. Using $R2$ provides least throughput because $R2$ makes idle clients break the chain of transmissions most often, and therefore being the worst at ring formation, and hence one that we should avoid given that our goal is to maximize system throughput. It is clear that when clients 7-12 are less than 40% busy, clients 1-6 have larger throughput if clients 7-12 form a separate piggyback ring. However, when clients 7-12 are more than 40% busy, it is better for all clients to form a single ring. Therefore we suggest that δ be set around 40%. Note that whatever δ is used, CHAIN running PM outperforms static CHAIN using $R2$ significantly, when clients 7-12 are less than 80% busy. The reason is that even when there is only one ring, PM keeps busy clients together in the ring and avoids the formation of the "bad" ring(s) such as $R2$.

4.4.7 Trace-Driven Experiments

In the previous two sections we have studied the performance of CHAIN under unsaturated traffic. This section shows that CHAIN outperforms 802.11 DCF under real traffic. We evaluate CHAIN using traces

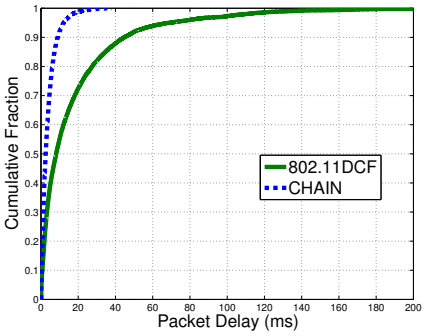


Figure 4.10: The comparison of media access delay

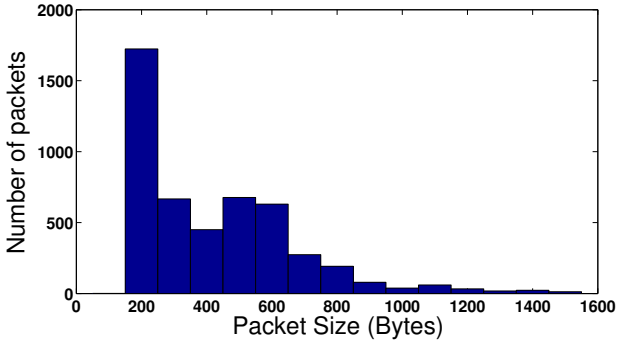


Figure 4.11: Histogram of Facetime application packet size

from Facetime applications running on Iphones. We collected traces for the Facetime video call application by running the application and logging the timestamps of packets. Because there is no way to collect traces directly on Iphones, we set a sniffer node nearby fir overhearing wireless transmissions and run tcpdump on the sniffer node to collect Facetime packets. We observe that the frequency of packet generation for one direction of transmission is roughly 100 packets/second, with packet size varying between 200Bytes and 1500Bytes. Figure 4.11 shows the distribution of the packet size over 4000 packets. We can see that over 50% of them are small packets of size less than 400Bytes. We also modify the J-sim simulator to generate traffic from the collected trace. The performance of CHAIN and 802.11 DCF is compared under two network setups.

Congested network: Consider one AP with nine clients. The link data rate is 11Mbps. We mimic the scenario that every client is Facetiming by replaying the collected traces in the simulation. Note that there are both uplink and downlink traffic. When the CHAIN protocol is running, the packet deliver ratio of all clients is 98.9% with average queueing delay of 16.7ms. However, when DCF is adopted, the packet deliver ratio of is 77.6%.

Busy but unsaturated network: Consider one AP with twelve clients. The link data rate is 24Mbps and every client is Facetiming. The bandwidth is sufficient no matter which MAC protocol is adopted. However, CHAIN significantly reduces the queueing delay in comparison to 802.11 DCF, as shown in Figure 4.10. CHAIN manages to reduce the average queueing delay to 3.92ms, and 98.1% of packets have delay less than 20ms. In comparison, when 802.11 DCF is used, the clients have average queueing delay of 18.1ms, and 26.9% of packets have delay larger than 20ms.

4.5 Conclusion

This chapter presents CHAIN, a distributed random media access protocol, that significantly improves uplink performance of WLANs. CHAIN introduces a novel *piggyback transmission opportunity* by defining a precedence relation among clients. A client can immediately transmit a new packet after it overhears a successful transmission of its predecessor, without going through regular contending process. When the network load is low, CHAIN behaves similar to DCF; But when the network becomes congested, clients automatically start chains of transmissions to improve efficiency. Therefore, the overall contention overhead is significantly reduced. Based on overhearing, CHAIN is a light-weight protocol that adds little coordination overhead. It also co-exists friendly with DCF, making it possible to be incrementally deployed in existing WLANs. We analytically prove the correctness and fairness of CHAIN. Our extensive simulations on J-SIM verify our analytical results, and demonstrate significant performance gain of CHAIN over DCF.

Chapter 5

SOFA: A Sleep-Optimal Fair-Attention Scheduler

5.1 Introduction

There is increasing use of battery-powered portable devices, such as smart phones and PDAs, to access the Internet through Wi-Fi. Unfortunately Wi-Fi communication consumes a significant amount of energy. For instance, the Lucent IEEE 802.11 WaveLAN card [56][43] consumes 1.65 W, 1.4 W, and 1.15 W in the transmitting, receiving, and idle modes, respectively. Another example is the Motorola Droid phone, for which power consumption measurements that we have conducted show that the base energy consumption is around 200mW with the backlight off, and close to 400mW with the backlight on. In comparison, the phone's energy consumption soars to over 800mW when the Wi-Fi radio is active. It is obvious that reducing Wi-Fi power consumption is crucial for extending the battery life of portable devices.

The IEEE 802.11 Power Saving Mode (PSM) has been proposed to reduce such power drain. Unlike the original Wi-Fi Constantly Awake Mode (CAM) which draws power over 1W all the time, a mobile device running PSM can go to sleep which incurs power consumption of only around 50mW [43]. However, when a PSM device is sleeping, it cannot transmit or receive any packets; hence, PSM clients conserve energy at the cost of larger packet delivery delay.

We now provide a brief introduction to how standard PSM works. In an infrastructure WLAN, the access point (AP) sends beacon packets periodically. Each beacon packet indicates the beginning of a new beacon period, and informs the PSM clients about the presence of buffered packets via the Traffic Indication Map (TIM) field in the beacon packet. The PSM clients wake up periodically, slightly prior to the beginning of each beacon period, to listen to the TIM. If the bitmap for the client is not set to 1 in the TIM, then the client goes back to sleep immediately. Otherwise, the client has to remain awake till the last packet scheduled for it in the current beacon period is delivered. The client knows whether a packet it has received is the last one in the beacon interval by inspecting the MORE DATA bit field in the packet header. If it is set to 0, then the packet is the last one, hence, after receiving this packet from the AP, the corresponding client can go back into and stays in the sleep mode till the beginning of the next beacon period, if the client

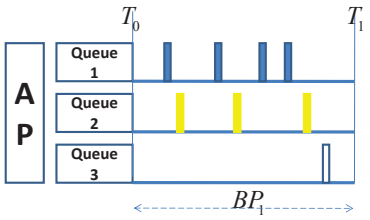


Figure 5.1: Packets arrive and are buffered at the AP in Beacon Period 1, denoted BP_1.

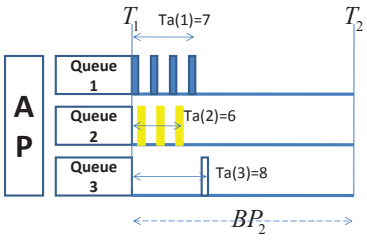


Figure 5.2: The AP delivers the buffered packets to clients employing the FCFS policy in BP_2.

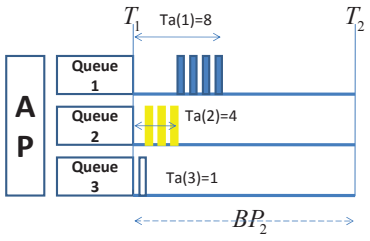


Figure 5.3: The alternative: The AP delivers the buffered packets to clients employing a policy that optimizes the total system's sleep time in BP_2.

has no packet to transmit.

However, as pointed out by several researchers [50][23][8], the power consumption of a PSM client depends not only on the traffic load destined for it, but also on the traffic loads destined for other clients. The reason is that when the AP serves more than one client or there is background traffic, the packet transmission from the AP to the PSM client may be susceptible to interruption by data transmission from other hosts or from the AP to other hosts. This results in the prolongation of the time for the PSM client to receive the buffered data client, and thus more power consumption.

In order to better understand this phenomenon as well as the desired objective, consider the simple example illustrated in Figures 5.1-5.3, where there is a WLAN with one AP serving three clients, 1, 2 and 3, who have downlink traffic only. At the beginning of the first beacon period BP_1, there is no downlink packet buffered at the AP, and all the clients go back to sleep. During BP_1, four packets to Client 1, three packets to Client 2, and one packet to Client 3, arrive at the AP from the Internet, as shown in Figure 5.1. The x-axis is the time line. Since all clients are sleeping, the AP buffers all the packets. When the second beacon period BP_2 commences, the clients wake up, and are ready to retrieve their buffered data. Figure 5.2 shows the time line of what happens if the packets are delivered by the AP in the normal first-come-first-serve (FCFS) order. Figure 5.3 shows the more desirable alternative time line of packet deliveries if they are delivered in the order that the sum of the three clients' awake time is minimized, or, equivalently, if the total sleep time over all nodes in the system, which we call system sleep time, is maximized. In both figures, $Ta(i)$ represents the duration that Client i keeps awake, in units of packet transmission time. If we consider the sum of clients' awake time as a metric representing the system energy consumption, then the second scheduling policy also achieves *system sleep optimality* by minimizing the system's total energy consumption. This example reveals that although we usually consider FCFS policy to be a fair scheduling policy that gives

all clients fair service and attention, it is in fact not optimal vis-a-vis energy savings. Another point worth noting is that the FCFS policy makes Client 3 wait eight units of time to retrieve a single packet. Intuitively, users expect that the more service/attention a user gets, the more energy consumption it can afford, and thus should bear. Thus we may regard Client 3 as not being treated fairly in terms of energy consumption. In other words, FCFS does not provide energy-fairness in this example. Even the alternative scheduling in Figure 5.3 is not perfect. Suppose the wireless channel is lossy in BP_2 and only the first four packets are delivered before BP_2 ends. Then Client 1 does not receive its *fair attention*. Moreover, all its packets are *deferred* to the next beacon period although one of its packets would have been the first to transmit in BP_2 under the FCFS principle.

Motivated by this simple example, we address the following design problem: *What is the optimal strategy to deliver the downlink packets so that the PSM clients waste the least time staying awake and overhearing downlink traffic being transmitted to other clients?* Clearly, the absolutely sleep-optimal solution is for the AP to wake up one client, send all and only its packets to it, put it back to sleep, then wake up the next client, and repeat the process. However, this solution is not practical since the AP cannot wake up a client anytime it wants to, since the client is not capable of receiving any information from the AP while it is sleeping.

To provide a solution to this basic design problem, in this chapter we present SOFA, a downlink traffic scheduler for the AP that achieves *system sleep-optimality*, and *energy fairness*, while meeting an *attention fairness constraint*, and *non-beyond beacon period deferral constraint*. Further, SOFA is analytically proved to be stable. The terminologies in italics will be formally introduced and explained in Section 5.3, but the readers should have got a flavor of them from the above example.

The remainder of the chapter is organized as follows. We discuss related work in Section 5.2. Section 5.3 describes the problem formulation, lists the notation and terminology, and describes the objectives. Following that, in Section 5.4, we present the design of SOFA. Section 5.5 analytically proves the optimality and fairness of SOFA. Section 5.6 summarizes simulation results. Finally, concluding remarks are made in Section 5.7.

5.2 Related Work and Motivation

We classify the related work in the area of energy savings for Wi-Fi radios into three categories. The first category comprises of dynamic sleep/wakeup control strategies on the client side. The second category comprises of strategies that do scheduling on the AP side to reduce the energy consumption. Our work falls

in this category. The third category comprises of strategies that use support from upper layers, such as network layer and application layer, to help PSM users save power.

5.2.1 The Client Side

A lot of work has been done concerning tuning the adaptive PSM to achieve a tradeoff between delay and energy saving. Krashinsky and Balakrishnan [35] have carried out a simulation analysis of PSM focusing on web-browsing traffic, and presented a Bounded Slowdown Protocol (BSD). Similarly, smart-PSM [45] provides bounded delay while minimizing energy consumption, by estimating the response-time distribution, and determines the PSM client's action sequence accordingly. Other related work includes XEM [7], which switches the wireless interface into off-mode instead of sleep-mode during long idle durations, and Forced Idling [11] which puts the radio in a low-power idling state, to avoid wasting energy due to overhearing background communications.

OPSM [9] targets a similar problem as us: the authors observe that the performance of standard PSM degrades due to congestion and background traffic. However, they choose to work on the client side.

5.2.2 The Access Point Side

There is a lot of work on the AP side that realizes that the PSM performs poorly under heavy background traffic. The authors of [24] point out that for a selected PSM client, the existence of background traffic from the other PSM clients can result in extra energy drain. Their following work, Scheduled PSM [25], employs a TDMA scheme to handle this problem. Although it saves energy, Scheduled PSM requires modifications on both the AP side and the client side, and changes the IEEE 802.11 standard, which makes it hard to deploy.

LPTSPT and DEEM [39] use heuristics to approximate the globally optimal energy consumption of PSM clients. However, they do not consider fairness issues. Napman [50] leverages AP virtualization and a new energy-aware fair scheduling algorithm to make different PSM clients wake up at different times, so that the AP can serve them separately. This solution is beautiful; however, due to hardware limitation the AP running Napman can serve at most four PSM clients.

There is much work that embraces the idea of staggering the awake times of different PSM clients to save power, such as Centralized PSM [65] and LAWS [41]. Reference [57] studies alternative protocols that use multiple bits of the TIM to indicate the order of transmissions. All these approaches require both client and AP modifications. Our solution is more readily deployed than theirs, although in theory their solution can help the PSM clients save more energy than ours.

5.2.3 Upper layer Support

Some researchers try to help PSM users save energy through obtaining reports from or modifying upper layers, such as network layer and application layer. PSM-throttling [58] reshapes the TCP traffic into periodic bursts, so that the client can turn on/off its wireless network interface accordingly. In [16], a proxy is used to batch packets from various streaming applications, so that the device can sleep between the receptions of batched packets.

5.3 Problem Formulation and System Model

We consider a WLAN system consisting of one access point, m PSM clients which are numbered $\{1, 2, ..., m\}$, and n Constant Active Mode (CAM) clients which are always awake and numbered $\{m+1, m+2, ..., m+n\}$. We assume that the AP has full control of the downlink traffic and can use any scheduling policy as long as it does not violate the IEEE 802.11 standard. which is true in many enterprise WLANs nowadays. At the beginning of each beacon period BP, the AP notifies the PSM clients of the presence of buffered packets for them, through the TIM field in the beacon packet. We assume that every PSM client wakes up for beacons at the beginning of every beacon period. In the future we can relax this assumption and allow each client to choose a different DTIM window to trade delay for power conservation, but in this paper we stick to the basic model. If the client's corresponding TIM field in the beacon packet is set, it stays awake and receives its buffered packets; otherwise it goes back to sleep immediately. A MORE bit in the data packet from the AP indicates whether more packets are buffered at the AP. When there are no more packets for it buffered at the AP, the client goes back to sleep after sending a NULL packet to the AP.

5.3.1 Notation

Let $BP(k)$ denote the kth beacon period, and $AVEATT_i(k)$ denote the average *attention* per beacon period that Client i gets by the end of $BP(k)$. Throughout this paper the *attention* of Client i is defined as the time that the AP spends to transmit downlink packets to Client i, since in this paper our goal is to enforce *airtime fairness* [30], which has been widely regarded as a better fairness goal than throughput fairness, and has in fact been enforced by some enterprise WLAN AP manufacturers such as Meru Networks [4]. However, others may adopt other attention metrics of interest to them, such as throughput, as they wish. Let $AWAKE_i(k)$ denote the time that Client i stays awake in the k-th beacon period, $BP(k)$, till the AP allows the client to go back to sleep by setting the MORE bit to zero in the last data packet it sends to Client i in the beacon interval.

In our scheduler design, the AP transmits downlink packets to PSM clients before it sends packets to any CAM clients. In order to achieve attention fairness between PSM clients and CAM clients, the AP must decide "attention quotas" for the PSM clients. We define $CAP_{PSM}(k)$ as the maximum (or "cap" on the) time for the AP to serve all PSM clients during $BP(k)$, which is determined at the beginning of $BP(k)$, and $ACTUAL_{PSM}(k)$ (or $ACTUAL_{CAM}(k)$ respectively) as the actual time that the AP spends to serve PSM (CAM) clients during $BP(k)$, which is measured by the AP at the end of $BP(k)$. $ATT_i(k)$ denotes the attention, i.e., service time, that Client i gets during $BP(k)$.

5.3.2 Desired Objectives

Here we describe the targets and constraints that we have in mind concerning the design the scheduler at the AP. Our goal is to achieve *system-sleep optimality* and *energy-fairness* by allowing the AP to schedule downlink traffic properly. As mentioned in Section 5.1, we are mostly interested in reducing the energy that the PSM clients waste overhearing the transmissions to other clients. Therefore, in the remainder of this paper, "energy" refers to the specific energy consumption of the PSM clients. *System-Sleep Optimality* is then defined as follows.

Definition 1. *System-Sleep Optimality: In any $BP(k)$, suppose that the set of dowlink packets for the AP to deliver to the clients during a beacon period is given, or, equivalently say all the $ATT_i(k)$ are given. We call a WLAN scheduling policy as system-sleep optimal if the total awake time of the PSM clients, $\sum_{i=1}^{m} AWAKE_i(k)$, is minimized.*

We say the scheduling policy is *energy-fair* if it attains the min-max *Energy per Unit Attention*, where the latter is a new metric that we propose:

Definition 2. *Energy per Unit Attention (E_{ua}): Recalling that Client i must keep awake for $AWAKE_i(k)$ during $BP(k)$ in order to get service $ATT_i(k)$ from the AP, Client i's Energy per Unit Attention during the kth beacon period is defined as*

$$EUA_i(k) := \frac{AWAKE_i(k)}{ATT_i(k)}. \tag{5.1}$$

We now define *energy-fairness* as follows

Definition 3. *Energy fairness: Suppose that for a beacon period $BP(k)$, all the service attentions, i.e., the $ATT_i(k)$s, are specified. Then, a WLAN scheduler is said to be energy-fair if it achieves*

$$\min \max_{i=1}^{m} EUA_i(k) \tag{5.2}$$

over all scheduling schemes.

The logic behind this definition is quite intuitive and can be explained by a simple example. Suppose there are two clients, A and B. If A requires more attention than B does, but waits less time to get attention than B does, then we regard B as having been treated unfairly, and the goal of our scheduler design is to prevent such events from happening.

Besides the goal of saving energy, the design should meet three additional constraints described below. We call the first constraint *attention fairness* among all PSM clients and CAM clients. The motivation for this goal is that although PSM clients are given the privilege of being served before the CAM clients in order to save their energy, we do not want PSM clients to gain more attention then CAM clients. No client should take more than its fair share of the available wireless media capacity. Attention fairness is a long-term min-max fairness as defined below:

Definition 4. *Attention fairness: Suppose that in every $BP(k)$ with $k \geq 1$, the attention requested for downlink traffic of Client i is λ_i, while the total amount of attention (for downlink traffic) that the AP can serve is $TOTALATT$, with $\sum_{i=1}^{m+n} \lambda_i < TOTALATT$. We say that the system achieves attention fairness when there is a threshold such that the clients' attention allocation is as follows:*

$$AVEATT_i(k)=\lambda_i \texttt{ if } \lambda_i < \xi, \tag{5.3}$$
$$AVEATT_i(k)=\xi \texttt{ if } \lambda_i \geq \xi,$$

and

$$\sum_{i=1}^{m+n} AVEATT_i(k) = TOTALATT. \tag{5.4}$$

We call ξ as the attention upper bound*.*

The second constraint is a *no unnecessary deferral-beyond-beacon-period constraint*, motivated by the design objective that the energy savings should not be attained at the expense of introducing excessively large media access delay to the PSM clients. It is obvious that the less frequently a client wakes up, the less energy the client consumes, but the price that is paid is the large delay that the client encounters. For example, if a client with downlink constant rate UDP traffic only requires 1/10 of the wireless airtime capacity, then the optimal power saving schedule for this client is for the AP to put this client to sleep for the first 9 beacon periods out of every 10 beacon periods, and then serve this client only during the 10th beacon period. In this way the client's attention request is met with least awake time. However such scheduling results in large communication delay and delay jitter exceeding a beacon period. Starving one client for one or several beacon periods can be catastrophic for some applications such as TCP connections and sometimes

can make the client drop the WLAN connection. Therefore we do not allow such an approach in this paper. We enforce the constraint that if a client wakes up at the beginning of $BP(k)$ and has data packets buffered at the AP, then the AP cannot put it back to sleep without serving any of its already buffered traffic.

The last constraint is a *practical* constraint, requiring that all the designs and modifications should be done on the AP side only, and that they should conform to the IEEE 802.11 standard. This is for practical reasons, since a solution requiring modifications on the client side would be very difficult, if not impossible, to deploy. Similarly, designs not compatible with IEEE 802.11 will find it much more difficult to gain widespread acceptance over the short or medium term.

5.4 SOFA Design

5.4.1 Overview

The SOFA scheduler is designed to minimize the total energy consumption of all PSM clients and achieve energy fairness among PSM clients, while satisfying *attention fairness, no-deferral-beyond-beacon-period* and *practical* constraints. The basic idea is that in each beacon period the access point serves the PSM clients one by one. After it finishes serving one PSM client, which occurs when there are no more buffered packets at the AP for that client, or because the client has run out of its attention quota for this BP, the AP sets the MORE DATA bit of the last packet to 0 so that the client knows that it can go back to sleep. After serving all PSM clients, the AP spends the rest of the beacon period serving CAM clients.

This AP scheduling procedure consists of three steps. First, at the beginning of the kth beacon period $BP(k)$, the AP determines $CAP_{PSM}(k)$. Second, it determines $CAP_i(k)$ for each Client i. More generally, $CAP_i(k)$ is defined as the maximum attention that PSM Client i can get during $BP(k)$, while throughout this paper $CAP_i(k)$ refers to the maximum time for the AP to deliver downlink packets to Client i during $BP(k)$. The AP then orders the PSM clients based on $CAP_i(k)$ and the *Predicted Attention Request* PA_i. By PA_i we mean the estimated transmission time to deliver all packets of Client i that are currently buffered at the AP. Last, the AP spends the remaining portion of the beacon period to serve the CAM clients. We note that the protocol is flexible with respect to how it treats the CAM clients: it can adopt any existing research [4][30] to achieve attention fairness among CAM clients. Next we elaborate on the first two steps of SOFA.

5.4.2 CAP Determination

Although the AP knows the duration of its own beacon period (denoted by BP in the sequel), the total available attention/service time for downlink traffic, $TOTALATT$, is usually less than T, because of the existence of uncoordinated uplink traffic, other WLANs in the neighborhood, transmission failures caused by channel fading, collisions, and many other reasons. Even worse, since $TOTALATT$ is also time-varying, it is very difficult to estimate $TOTALATT(k)$ at the beginning of kth beacon period $BP(k)$. Therefore $CAP_{PSM}(k)$ cannot be directly assigned. We propose an adaptive algorithm to assign $CAP_{PSM}(k)$ based on the AP's historical observation of the system's wireless medium, and the performance of the clients. The pseudo-code specification is given in Procedure 5. Recall that $ACTUAL_{PSM}(k-1)$ ($ACTUAL_{CAM}(k-1)$, respectively) is the actual time that the AP spends to serve PSM (CAM, respectively) clients during $BP(k)$. When the wireless medium is not fully utilized, $Idle(k)$ in Procedure 5 is the wireless media idle time experienced by the AP during $BP(k)$. The main idea of this Procedure 5 is that we divide the clients into two groups: the PSM group and the CAM group. When the wireless capacity is not large enough to support all clients' attention requests, then during the current beacon period, the AP gives more attention to the group which has historically received less attention. The fairness and optimality properties of the protocol are proved in section 5.5.

5.4.3 CAP Allocation and Serving Sequence

After $CAP_{PSM}(k)$ is determined, the AP starts scheduling downlink traffic for the PSM clients. The AP needs to make two decisions: (1) In the current beacon period, among all the PSM clients that have packets buffered at the AP, how much service should the AP assign to each client, and (2) In what sequence should the AP deliver data packets to those clients. We have already pointed out that although packet level first-come-first-served (FCFS) is a good fairness policy, scheduling packets in a FCFS manner is not sleep optimal. This raises the question of what policy to use. Theorem 1 helps in answering this question by determining the best order to serve PSM clients once their service times ATT_i for $1 \leq i \leq m$ are given.

Since all PSM clients wake up at the beginning of the beacon period, and Client i does not go to sleep until it has been served for ATT_i, $AWAKE_i$ consists of the time interval spent on the clients that are served prior to Client i, and the time interval spent on Client i (i.e., ATT_i).

Let us define a *noninterrupted schedule* as a schedule in which once the AP starts to serve a client, it continuously works on this client till the service assigned to this client is completed. In contrast, an *interleaved schedule* is a schedule in which some client's service is interrupted by service to others.

Algorithm 5 ADJUSTMENT of $CAP_{PSM}(k)$

Require: $CAP_{PSM}(k-1)$, $ACTUAL_{PSM}(k-1)$,
$TOTAL_A(k-1)$, $ACTUAL_{CAM}(k-1)$, $Idle(k-1)$
Ensure: $CAP_{PSM}(k)$

if $k = 1$ **then**
  $CAP_{PSM}(k) \leftarrow T/4$
  return
end if
if $Idle(k-1) > 0$ **then**
  if $ACTUAL_{PSM}(k-1) < CAP_{PSM}(k-1)$ **then**
    $CAP_{PSM}(k) = CAP_{PSM}(k-1)$
  else
    $CAP_{PSM}(k) = CAP_{PSM}(k-1)$
    $+ACTUAL_{CAM}(k-1) \times \theta$
  end if
else
  if $ACTUAL_{PSM}(k-1) = TOTAL_A(k-1)$ **then**
    $CAP_{PSM}(k) \leftarrow ACTUAL_{PSM}(k-1)$
  end if

$$a = \max_{i=1\ldots m} AVEATT_i(k-1)$$

$$b = \max_{i=m+1,\ldots,m+n} AVEATT_i(k-1)$$

  if $ACTUAL_{PSM}(k-1) = CAP_{PSM}(k-1)$ **then**
    $CAP_{PSM}(k) += (b-a) \times \varphi$
  else
    $CAP_{PSM}(k) += \min\{0, (b-a) \times \varphi\}$
  end if
end if

Lemma 2. *The system sleep optimum is achieved by a noninterrupted schedule.*

Proof. Suppose the system sleep optimum is achieved by an interleaved schedule. Then, the serving time of some clients is split into several subintervals τ. Denote the last subinterval of the client by τ_e. A client can go to sleep only after its τ_e has been served. Since the schedule is interleaved, there must be a client whose τ_e follows a subinterval of another client that is not the last subinterval. Swapping the serving order of the two subintervals yields another schedule. The new schedule reduces the total energy consumption because client i can go to sleep earlier. Therefore, the optimal scheduling must be a noninterrupted schedule. □

Thus, we consider only noninterrupted schedules. That is, once the AP starts to serve client i, it continuously serves it for time interval ATT_i.

Theorem 1. *Given a set of service times $\{ATT_i\} : 1 \leq i \leq m$ for a beacon period, serving the corresponding clients in ascending order of ATT_i achieves the system sleep optimum.*

Proof. Consider a schedule σ that is not in the ascending order of ATT_i and denote its total energy consumption by $E(\sigma)$. Then, there must be some ATT_n and ATT_m such that $ATT_n \leq ATT_m$ and m is served prior to n. Another schedule σ' can be obtained by swapping the serving order of n and m. σ' reduces the total energy consumption since

$$E(\sigma') = E(\sigma) - (k+1)(ATT_m - ATT_n) < E(\sigma),$$

where k is the number of clients that are served between ATT_m and ATT_n. Thus, σ' is a strictly better schedule than σ, and establishing the result. □

Theorem 2. *Given a set of service times $\{ATT_i\} : 1 \leq i \leq m$ for a beacon period, serving the corresponding clients in the ascending order of ATT_i achieves energy fairness.*

Proof. If a schedule σ is not in the ascending order of ATT_i, then there must be some ATT_n and ATT_m such that $ATT_n \leq ATT_m$ with m served prior to n. We will show that another schedule σ' that swaps only the serving orders of n and m will achieve smaller EUA than σ. By Definition 3, $EUA_n(\sigma) = \frac{T_{a_n}(\sigma)}{ATT_n}$ and $EUA_m(\sigma) = \frac{T_{a_m}(\sigma)}{ATT_m}$. Since m is served prior to n in σ, we have $T_{a_m}(\sigma) \leq T_{a_n}(\sigma)$. Thus, we have $EUA_m(\sigma) < EUA_n(\sigma)$.

σ' swaps n and m and preserves other clients in the same order as σ. Hence, only the clients between m and n change their EUA value. Let j be some client served between m and n. After swapping m and n,

$T_{a_j}(\sigma') = T_{a_j}(\sigma) - ATT_m + ATT_n \leq T_{a_j}(\sigma)$. Thus,

$$EUA_j(\sigma') \leq EUA_j(\sigma). \tag{5.5}$$

Since $T_{a_n}(\sigma') < T_{a_n}(\sigma)$, we have

$$EUA_n(\sigma') < EUA_n(\sigma). \tag{5.6}$$

We know $T_{a_m}(\sigma') = T_{a_n}(\sigma)$ and $ATT_m \geq ATT_n$, hence

$$EUA_m(\sigma') \leq EUA_n(\sigma). \tag{5.7}$$

From (5.5), (5.6) and (5.7),

$$\max_i EUA_i(\sigma') \leq \max_i EUA_i(\sigma). \tag{5.8}$$

Therefore, we can conclude that the ascending order achieves the minimum of $\max_i EUA_i$ over all possible schedules. □

To summarize, we have determined both an attention-fair and a sleep-optimal order for serving clients, *provided* we know their service times $\{ATT_i : 1 \leq 1 \leq m\}$. Therefore the next issue is to determine how to choose $\{ATT_i : 1 \leq 1 \leq m\}$ so as to yield attention fairness as well as satisfy the no-deferral constraints. In the following subsections, we start with three intuitive scheduling solutions that *do not* work well, in order to show that this is a non-trivial problem and how an examination of their failures leads us to the right solution: SOFA.

5.4.4 The Least Attention First (LAF) Scheduler

Procedure 6 provides the pseudo-code of the Least Attention First (LAF) scheduler, and Procedure 7 briefly describes how the AP handles the PSM client l that currently has the highest priority. At the beginning of the beacon period, the AP picks up the client that has had least historical attention and transmits to it all its buffered packets. Then it picks the client with second least attention to serve, and so on.

The Serve-with-Priority procedure tries to make PSM client l receive all of its buffered packets before the other clients. First it employs the idea of "extra backoff," to give l a higher priority to retrieve its packets. Note that in each beacon period, the AP must receive one or multiple control messages from a PSM client (to prove the client is awake) before it sends out the data packets to it. For a *static* PSM client, every single data packet transmission from the AP requires a PS-POLL message from the client; and for a *dynamic*

Algorithm 6 Least Attention First

Require: $CAP_{PSM}(k), AVEATT_i(k-1), i = 1, ..., m$
Ensure: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
1: $U = \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
2: **while** $U \neq \Phi$ and $CAPR(k) > 0$ **do**
3: $p \leftarrow \arg\min_l \{AVEATT_l(k-1) | c_l \in U\}$
4: $ATT_p(k) =$ Serve-with-Priority(c_p, ∞)
5: UPDATE $AVEATT_p(k)$
6: $U \leftarrow U - \{c_p\}$
7: $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
8: **end while**

PSM client, the AP must first receive a NULL packet indicating the client is awake for the rest of beacon period, before transmitting the buffered packets for that client. The "extra backoff" makes the AP wait longer for those control packets. Yet when the extra backoff terminates, the AP can switch to serving other clients without wasting too much time waiting for the control packets when client l fails, malfunctions, or just does not wake up. For practical reasons, we cannot allow the AP to wait forever for l's control packets. Therefore, besides the steps shown in Procedure 6, the AP takes away client l's priority if the PS-POLL or NULL packet does not come with a grace period, say 5ms, to give the priority to other clients, and may return to l the priority if its PS-POLL or NULL packet does arrive later in this beacon period. Furthermore, we make use of the Unscheduled Automatic Power Save Delivery (U-APSD) defined by the IEEE 802.11e standard. When a PS-POLL/NULL packet from l is received, instead of sending out one packet to l, the AP can dequeue multiple packets of l and send them in a burst, using the 802.11e TXOP-like mechanism before its CAP is met. The Serve-with-Priority procedure is also used in the other scheduling policies in this paper.

However, LAF cannot guarantee that it serves clients in the increasing order of service time ATT_l. For example, consider three clients A, B, and C, whose attentions are $AVEATT_A = AVEATT_B > AVEATT_C$. Suppose however that C has bursty traffic while the traffics of A and B are smooth. Then during the beacon period when C's burst arrives, the AP serves C first, although $ATT_A = ATT_B < ATT_C$. Moreover, the existence of bursty traffic also prevents LAF from satisfying the no-deferral constraint. Therefore we must set an upper bound on ATT_i, which leads to the next policy.

5.4.5 The History-Based Attention (HBA) Scheduler

As shown in Procedure 8, HBA calculates the attention capacity bound for each client at the beginning of each beacon period, and then serves the clients in the increasing order of estimated service time. The function $f(AVEATT_l(k-1), U)$ in the third line calculates the proportion of service time that client l can

Algorithm 7 Serve-with-Priority(Client, Attention Bound)

Require: client l, CAP_l
Ensure: $ACTUAL_l$

```
1: if PS-POLL from l has been received then
2:   GOTO 7
3: end if
4: CAP ← CAP_l
5: while (this BP not finished) and (remaining CAP > 0) do
6:   isExtraBackoff ← false, mark PS-POLL from l not received
7:   Do normal 802.11 standard backoff
8:   while No Event do
9:     Switch event:
10:    CASE (PS-POLL from l is received):
11:    mark PS-POLL from l received
12:    HeadofTXqueue ← Dequeue(l)
13:    if isExtraBackoff then
14:      GOTO 35
15:    end if
16:    CONTINUE
17:    CASE (Awake Notification (a NULL packet) from l)
18:    HeadofTXqueue ← Dequeue(l,CAP), mark l AWAKE
19:    if isExtraBackoff then
20:      GOTO 38
21:    end if
22:    CONTINUE
23:    CASE (Backoff timeout)
24:    if PS-POLL from l has been received then
25:      GOTO 35
26:    else
27:      if l AWAKE then
28:        GOTO 38
29:      else
30:        isExtraBackoff ← true; DO Extrabackoff;
31:        CONTINUE
32:      end if
33:    end if
34:    CONTINUE
35:    CASE (Extra Backoff timeout)
36:    send one of the other packets with highest priority
37:    BREAK
38:  end while
39:  packet.MORE ← 0 if it is the last packet to l
40:  send the packet popped up from TXQueue (immediately without backoff), UPDATE CAP
41: end while
42: TREAT l as a CAM client and start to transmit TXQueue
43: return ACTUAL_l = the time taken to transmit l's packets
```

Algorithm 8 HBA

Require: $CAP_{PSM}(k), AVEATT_1(k-1), ..., AVEATT_m(k-1)$
Ensure: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
1: $U = \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
2: **for** each $c_l \in U$ **do**
3: $CAP_l = CAPR(k) \times f(AVEATT_l(k-1), U)$
4: $ATT_l(k) = \min(CAP_l(k), PA_l(k))$
5: **end for**
6: **while** $U \neq \Phi$ and $CAPR(k) > 0$ **do**
7: $p \leftarrow \arg\min_l \{ATT_l(k) | c_l \in U\}$
8: Serve-with-Priority$(c_p, ATT_l(k))$
9: UPDATE $AVEATT_p(k)$
10: $U \leftarrow U - \{c_p\}$
11: $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
12: **end while**

get, based on its own historical attention as well as the attention of all clients in the set U. As long as function $f(x, U)$ decreases with x, HBA assigns the AP's attention fairly among all PSM clients, because it always assigns more service "quota" to clients with less historical attention. Although HBA appears to meet our requirements at first glance, it has a fundamental problem: HBA may waste service time. When there exists a client i whose predicted attention is less than its "quota," the extra capacity assigned to client i is wasted and cannot be reused by other clients whose attention demands are larger than their "quotas." Therefore, even if the total $CAP_{PSM}(k)$ during $BP(k)$ is sufficient to meet all clients' attention requirements, the use of HBA can result in some clients suffering degraded performance. In order to eliminate such service time wastage, one can consider another scheduler design, *Online* HBA, as we next describe.

5.4.6 Online HBA Scheduler

Procedure 9 gives the pseudo-code of *Online* HBA scheduling. Instead of pinning down the service bound of every client at the beginning of the beacon period, *Online* HBA updates the service bounds of the remaining clients that have not been served yet. The AP does such updating every time it finishes serving one PSM client, say client i, and puts it back to sleep. If client i has used up its "quota," then the remaining service time is re-assigned for usage by the remaining clients. In this way *Online* HBA reduces the service time wastage in comparison to HBA. Unfortunately *Online* HBA cannot fully get rid of service time wastage.

Let us look into a simple example. Consider a WLAN with one AP and two PSM clients 1 and 2 associated with the AP. Let $CAP(k) = 50ms$, $PA_1(k) = 10ms$, and $PA_2(k) = 40ms$ for every beacon period k. Suppose that the service time bound assignment function f is defined as follows:

$$f(AVEATT_1, \{1, 2\}) = \frac{AVEATT_2^2}{AVEATT_1^2 + AVEATT_1^2}, \tag{5.9}$$

Algorithm 9 Online HBA

Require: $CAP_{PSM}(k), AVEATT_1(k-1), ..., AVEATT_m(k-1)$
Ensure: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
1: $U = \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
2: **for** $j = 1, \cdots, m$ **do**
3: **for** each $c_l \in U$ **do**
4: $CAP_l^j = CAPR(k) \times f(AVEATT_l(k-1), U)$
5: $ATT_l(k) = \min(CAP_l^j(k), PA_l(k))$
6: **end for**
7: $p \leftarrow \arg\min_l \{AVEATT_l | c_l \in U\}$
8: Serve-with-Priority$(c_p, ATT_p(k))$
9: UPDATE $AVEATT_p(k)$
10: $U \leftarrow U - \{c_p\}$
11: $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
12: **end for**

$$f(AVEATT_2, \{1, 2\}) = \frac{AVEATT_1^2}{AVEATT_1^2 + AVEATT_2^2} \tag{5.10}$$

In this WLAN, the attention that client B achieves is never above $20ms$. The reason for this is that the service bound assignment function f over compensates client A by giving it too much service quota. Suppose $AVEATT_2(k) > 20ms$ at $BP(k)$. Then we have $f(AVEATT_2, \{1, 2\}) < 0.2$, and therefore

$$ATT_2(k) = CAP_{PSM}(k) \times f(AVEATT_2, \{1, 2\}) < 10ms = ATT_1(k). \tag{5.11}$$

Note that the AP serves client 2 before client 1 (from lines 7-8 in Procedure 5), and so client 2 only gets $10ms$ service time, which results in $AVEATT_2$ falling below $20ms$ again. *Online* HBA cannot reach a stable scheduling sequence even when the network environment and traffic are stable. We have been unable to find any function f that always avoids causing the above situation in all general WLAN settings. A scheduling policy that is not *stable* is the last thing a system needs. We come up with the next design in order to solve the stability issue,

5.4.7 The Attention Fair (AF) Scheduler

After summarizing the pros and cons of the other attempts at designing scheduling policies, we now design the Attention Fair policy, AF, as given in Procedure 10. AF is derived from *Online* HBA, with three major modifications.

First, in each beacon period the AP running AF first serves those clients whose quota is larger than their predicted attention. In this way, no service time is wasted. Second, we carefully choose a specific quota assignment function f as specified in line 5. Third, as *Online* HBA does, AF adjusts the quotas of

Algorithm 10 AF

Require: $CAP_{PSM}(k), AVEATT_1(k-1), ..., AVEATT_m(k-1)$
Ensure: $ATT_i(k), AVEATT_i(k), i = 1, ..., m$
1: $U \leftarrow \{c_1, \cdots, c_m\}, CAPR(k) \leftarrow CAP_{PSM}(k)$
2: **for** $j \leftarrow 1, \cdots, m$ **do**
3: $V \leftarrow \Phi$
4: **for** each $c_l \in U$ **do**
5: $f(l, U) = \frac{\sum_{i \in U} AVEATT_i(k-1) - AVEATT_l(k-1)}{\sum_{i \in U} AVEATT_i(k-1) \times (m-j)}$
6: **if** (j=1) **then**
7: $CAP_l^j = CAPR(k) \times f(AVEATT_l(k-1))$
8: **else**
9: $CAP_l^j = CAP_l^{j-1} + R_{j-1} \times f(AVEATT_l(k-1))$
10: **end if**
11: $ATT_l(k) = \min(CAP_l^j(k), PA_l^j(k))$
12: **if** $(CAP_l^j > PA_l^j(k))$ **then**
13: $V = V \cup \{c_i\}$
14: **end if**
15: **end for**
16: **if** $V \neq \Phi$ **then**
17: $p \leftarrow \arg\min_l \{ATT_l | c_l \in V\}$
18: **else**
19: $p \leftarrow \arg\min_l \{ATT_l | c_l \in U\}$
20: **end if**
21: $ATT_p(k) =$ Serve-with-Priority(c_p, CAP_l^j)
22: $R_j = CAP_l^j - ATT_p(k)$
23: UPDATE $AVEATT_p(k)$
24: $U \leftarrow U - \{c_p\}$
25: $CAPR(k) \leftarrow CAPR(k) - ATT_p(k)$
26: **end for**

the remaining clients which are still awake each time the AP finishes serving any client. However, instead of re-assigning the quota, AF simply increases each client's quota if the client that is just going to sleep has not used up its quota. The details of the algorithm are shown in line 9. Note that with this new quota adjustment algorithm, any client's attention quota never decreases before it gets served within a beacon period. The last two modifications will be revisited and explained in the next section when we prove the stability of AF.

5.4.8 Attention Update

The average attention $AVEATT_i$ for client i is updated as follows,

$$AVEATT_i(k+1) = \alpha \times AVEATT_i(k+1) + (1-\alpha) \times ATT_i(k), \tag{5.12}$$

at the end of each beacon period, whiere $\alpha \in (0, 1)$ is a smoothing factor.

5.5 Protocol Analysis

In this section we prove that when both the wireless medium and the clients' attention requests are stable, then by running SOFA on the AP, the clients' service times $ATT_1, ..., ATT_m, ..., ATT_{m+n}$ converge to the fair point described in Definition 4, and furthermore, that the order that the AP serves all PSM clients within a beacon period also converges to the order that is both system sleep-optimal and energy-fair. We start by proving that if all clients of the AP are PSM nodes, then by running AF on the AP, $(ATT_1, ..., ATT_m)$ will converge to the fair point, and the order that the AP serves all PSM clients within a beacon period also converges as above.

5.5.1 The Stability of AF

First the following assumptions are made. Since we are analyzing the situation when the wireless medium and the clients' attention requests are stable, we assume that $\lambda_i(k) \equiv \lambda_i$ and $CAP_{PSM}(k) \equiv CAP$ for all k. Also, we assume $PA_i(k) \equiv \lambda_i$. The Stability of AF is trivial to prove when $\sum \lambda_i < CAP$; therefore we only study the case where $\sum \lambda_i \geq CAP$. Recall that according to Definition 4, ξ is the attention upper bound. Without loss of generality we assume, by renaming the clients if necessary, that $\lambda_i \leq \lambda_{i+1}$ and $PA_i \leq PA_{i+1}$, for $i = 1, ..., m-1$. In order to prove the stability of AF, we must show two things. First, we must show that there exists one (equilibrium) state which, once hit, the system will stay forever at. Second, we must show that the system will move to the equilibrium state eventually regardless of the initial or current state.

Lemma 3. *At $BP(k)$, if we have $ATT_i(k) = AVEATT_i(k-1) = \lambda_i$ (if $\lambda_i < \xi$); and $ATT_i(k) = AVEATT_i(k-1) = \xi$ (if $\lambda_i \geq \xi$), then at $BP(k)$, we must have $ATT_i(k+1) = AVEATT_i(k) = \lambda_i$ (if $\lambda_i < \xi$), and $ATT_i(k+1) = AVEATT_i(k) = \xi$ (if $\lambda_i \geq \xi$).*

Proof. We can check that Lemma 3 is true by analyzing AF at $BP(k+1)$ and simply calculating $ATT_i(k+1)$. □

Lemma 3 claims that the equilibrium state satisfies the property that for any client i, its ATT_i and $AVEATT_i$ both equal its fair attention share.

Note that AF does not waste any service time, which means that $\Sigma_i ATT_i(k) = CAP$ for all $k \geq 1$. Therefore, when k is large enough we must have $\Sigma_i AVEATT_i(k) = CAP$ too, no matter what the initial setting of $AVEATT_i(0)$ is. Therefore, during the proof of system state convergence, we can assume that $\Sigma_i AVEATT_i(k) = CAP$ for all $k \geq 1$ without loss of generality.

Lemma 4. $\texttt{Min}_i AVEATT_i(k)$ *increases with* k*, as long as*

$$\min_{1\le i\le m} AVEATT_i(k) \le \min\{\frac{1}{m}, PA_1, ..., PA_m\}. \tag{5.13}$$

Proof. At the beginning of $BP(k)$, the first round of attention quota assignment results in

$$\begin{aligned} CAP_i^1 =& \frac{\sum_j AVEATT_j(k-1) - AVEATT_i(k-1)}{(m-1)\sum_j AVEATT_j(k-1)} \times CAP \\ =& \frac{CAP - AVEATT_i(k-1)}{m-1} \\ =& \frac{\sum_{j\neq i} AVEATT_j(k-1)}{m-1}, \end{aligned} \tag{5.14}$$

where the second equality holds because

$$\Sigma_i AVEATT_i(k-1) = CAP$$

. Equation (5.14) shows that for any client i, its CAP_i^1 is the average attention of all other clients, which is always larger than $\min_i AVEATT_i(k-1)$. Assume $AVEATT_l(k-1) = \min_i AVEATT_i(k-1)$, then, clearly

$$CAP_i^1 \ge AVEATT_l(k-1), \texttt{ for all } i \neq l. \tag{5.15}$$

Because $AVEATT_l(k-1) \le \frac{CAP}{m}$, we have

$$\begin{aligned} CAP_l^1 =& \frac{CAP - AVEATT_l(k-1)}{m-1} \\ \ge& \frac{CAP}{m} \ge AVEATT_l(k-1). \end{aligned} \tag{5.16}$$

Putting (5.15) and (5.16) together, we have

$$CAP_i^1 \ge AVEATT_l(k-1), \texttt{for } i = 1, ..., m. \tag{5.17}$$

Let CAP_i denote the *real* attention bound that is assigned to client i right before the AP serves it. Recall that the attention bound of each client never decreases within a beacon period (from lines 6-9 in Procedure 10). Therefore

$$CAP_i \ge AVEATT_l(k-1), \texttt{for } i = 1, ..., m. \tag{5.18}$$

Since the service never exceeds the demand, we have for all i,

$$PA_i \ge AVEATT_i(k-1) \ge AVEATT_l(k-1). \tag{5.19}$$

Note that client i's service time is

$$ATT_i(k) = \min\{CAP_i, PA_i\}, \texttt{for } i = 1, ..., m. \tag{5.20}$$

By substituting (5.18), (5.19) into (5.20), we get

$$ATT_i(k) \geq AVEATT_l(k-1), \texttt{for } i = 1, ..., m. \tag{5.21}$$

Because $AVEATT_i$ is updated by taking a linear combination of old $AVEATT_i$ and current service time ATT_i as given in (5.12), and (5.21) holds for all clients, it follows that we must have

$$\min_{1 \leq i \leq m} AVEATT_i(k) \geq \min_{1 \leq i \leq m} AVEATT_i(k-1). \tag{5.22}$$

The proof is complete.

□

Lemma 5. *It is impossible that*

$$\min_{1 \leq i \leq m} AVEATT_i(k) < \min\{\frac{CAP}{m}, PA_1, ..., PA_m\}$$

for all k.

Proof. The proof is by contradiction.

Since $\min_i AVEATT_i(k)$ is bounded and non-decreasing, $\lim_{k\to\infty} \min_i AVEATT_i(k)$ exists. Suppose the limit is $\underline{A}$, with $\underline{A} < \min\{\frac{CAP}{m}, PA_1, ..., PA_m\}$. Picking the next beacon period $BP(k)$ and taking its limit, we can study what is happening as we get close to the limit. We see that $\lim_{k\to\infty} \min_i AVEATT_i(k) = \underline{A}$, and let c_l denote the client with the smallest attention quota in the limit. Then, considering the next beacon period and taking the limit, we must have $\lim_{k\to\infty} \min_i AVEATT_i(k+1) = \underline{A}$ too. Suppose the client with the smallest attention quota in the limit is c_j.

Next we will show that it is impossible that $j = l$, and also impossible that $j \neq l$. If $j = l$, then we must have $\lim_k ATT_l(k+1) = \underline{A}$, and this only happens if $\lim_k AVEATT_i(k) \equiv \frac{CAP}{m}$ for all i. Hence $\underline{A} = \frac{CAP}{m}$, which is a contradiction. On the other hand, if $j \neq l$, then we must have $\lim_k ATT_j(k+1) < \underline{A}$, which contradicts (5.21). □

Now we can prove the stability of AF.

Theorem 3. *When the AP runs the proposed AF scheduling policy, the scheduling order and attention assignment converge to the equilibrium state in Lemma 3.*

Proof. According to Lemma 5, there must exist a beacon period $BP(k)$ such that

$$\min_i AVEATT_i(k) = \min\{\frac{CAP}{m}, PA_1, ..., PA_m\} = \min\{\frac{CAP}{m}, PA_1\}. \tag{5.23}$$

The second equality holds because we have assumed that PA_1 is the smallest PA. If $\min_i AVEATT_i(k) = \frac{CAP}{m}$, then $AVEATT_i(k) = \frac{CAP}{m}$ for all i; hence we have reached the equilibrium state in Lemma 3. Otherwise, $\min_i AVEATT_i(k) = PA_1$. Then c_1 has both smallest attention request and highest attention quota. According to AF, c_1 must be the first client to be served in every beacon period from then on, which is how c_1 is supposed to be served in the equilibrium state. By recursion it follows that the behavior of the remaining $m-1$ clients also converges to the equilibrium state. □

Theorem 4. *The equilibrium state is system sleep-optimal and achieves min-max unit attention energy fairness.*

Proof. Clearly, in the equilibrium state, the AP serves the clients in the ascending order of ATT_i. Then the statement is true according to Theorems 1 and 2. □

In the next section we show that PSM and CAM clients share attention fairly.

5.5.2 Attention Fairness between PSM and CAM Clients

Theorem 5. *If CAM clients also achieve attention fairness among themselves, then all clients achieve attention fairness.*

Proof. Suppose at $BP(k)$, the CAM clients share their attention quota fairly amongst themselves, and the PSM clients share their attention quota fairly among themselves too. Procedure 5 identifies the following four cases and adjusts $CAP_{PSM}(k+1)$ accordingly. we use ξ_1 to denote the attention upper bound of PSM clients, and ξ_2 for CAM clients.

- Case 1: Neither CAM nor PSM clients use up their quota. Then $CAP_{PSM}(k+1) = CAP_{PSM}(k)$ from line 7 in Procure 5.
- Case 2: Only PSM clients use up their quota. Then CAP_{PSM} increases in the next beacon period from line 9.
- Case 3: Only CAM clients use up their quota. Then from line 20, CAP_{PSM} decreases in the next beacon period only if $\xi_1 > \xi_2$, and stays the same otherwise.

Table 5.1: The flow rates of 5 clients

Client No.	1	2	3	4	5
Flow rate (kbps)	100	300	700	1200	1200

- Case 4: Both PSM and CAM clients use up their quota. Then we compare ξ_1 and ξ_2. In the next beacon period, we give more quota to whichever is a smaller upper bound, from line 18.

From the manner in which Procedure 5 handles the four cases, we see that if the total wireless capacity is sufficient to serve all PSM and CAM clients, then the choice of CAP_{PSM} makes every client satisfied. Otherwise, the AP will keep giving more quota to the group with lower quota upper bound, till $\xi_1 = \xi_2$. Therefore the global attention fairness is achieved. □

5.6 Performance Evaluation

In this section, we evaluate SOFA via the NS-2 simulator [5]. We first verify that SOFA assigns attentions to PSM and CAM clients fairly. Then we show that SOFA significantly reduces the system energy consumption by comparing its performance with three other scheduling policies. We also show that SOFA achieves better energy fairness than other scheduling policies.

5.6.1 Simulation Setup

We use ns-2.33 with Power Management Extension [3] to conduct our simulation. For the energy model, we use the default settings of ns-2.33 Power Management Extension. The powers consumed by the wireless network interface in the transmit, receive, idle and sleep state are 660mW, 395mW, 35mW, and 1mW, respectively. The beacon period is 100ms. The default packet size is 800 bytes, and the default wireless link rate is 2Mbps if not stated otherwise. All wireless nodes are within each other's transmission range.

5.6.2 Attention Fairness Verification

The network setup is as follows. There is one AP (denoted by node A) and there are five clients associated with it. Clients numbered 1-4 are PSM nodes, and client numbered 5 is a CAM node. Each client receives a CBR UPD flow from the AP. The flow rates are given in Table 5.1. Another wireless node B is not associated with A, but it shares the wireless medium with A's WLAN, and broadcasts CBR packets. Node B plays the role of a wireless neighbor, whose existence makes it difficult for the AP to know the total available service time that it can provide to its clients. We run several sets of simulations with node B's flow

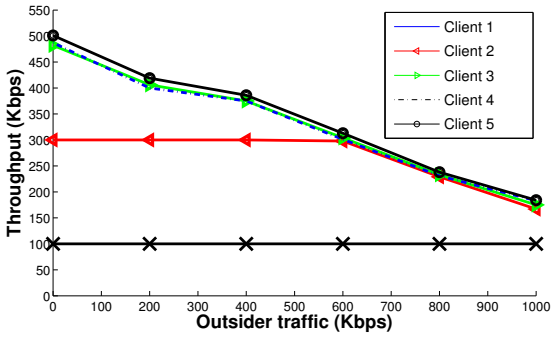


Figure 5.4: Per-client throughput for UDP downlink traffic with varying CBR background jamming traffic in a single rate WLAN.

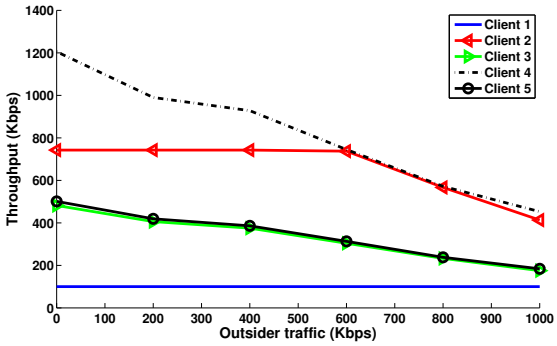


Figure 5.5: Per-lient throughput for UDP downlink traffic with varying CBR background jamming traffic in a multi-rate WLAN.

rate varying from 0 to 1Mbps. Figure 5.4 gives the throughput of every client when the link rate is 2Mbps for all clients. Since the link rates are all the same, attention/airtime fairness is equivalent to throughput fairness. In Figure 5.4, clients 3-5 which have saturated traffic share the attention fairly, while client 1 which has a lesser attention request gets well served. Client 2 is satisfied when the outsider, B's traffic is low. But when B's flow rate increases and takes more wireless medium, the available attention/service time of the WLAN increases. Therefore Client 2 cannot be fully satisfied, but it still gets a fair share of attention vis-a-vis client 4-6.

Figure 5.5 shows the throughput of every client when clients have different link rates. The network settings are the same as above, except that the link rates of Clients 2 and 4 increase to 5.5Mbps and the flow rate of Client 2 increases to 750Kbps. Figure 5.5 further verifies that SOFA achieves attention/airtime fairness among PSM and CAM clients in multi-rate wireless network. For example, when B's flow rate is 800Kbps, the throughput of Client 2 is 556Kbps and the throughput of Client 5 is 238Kbps, but that is because 2's bandwidth is 2.25 times of 5's. These two clients' airtime (the time that the AP sends data packet and receives ACK) are the same.

5.6.3 Power Consumption Evaluation

Now we compare the performance of four scheduling policies in a WLAN consisting of an AP, three PSM clients numbered 1-3, and one CAM client numbered 4. The flow rates of the downlink CBR UDP traffic to Clients 1-4 are 150Kbps, 300Kbps, 400Kbps and 400Kbps respectively. The four scheduling policies are: SOFA, Priority-Round-Robin (P-RR), Priority-First-Come-First-Serve (P-FCFS), and Normal. In the case of normal scheduling, on receiving a PS-POLL packet or a NULL frame from a PSM client, the AP enqueues one or more PSM packets at the tail of the transmission queue. In the second and third scheduling policies,

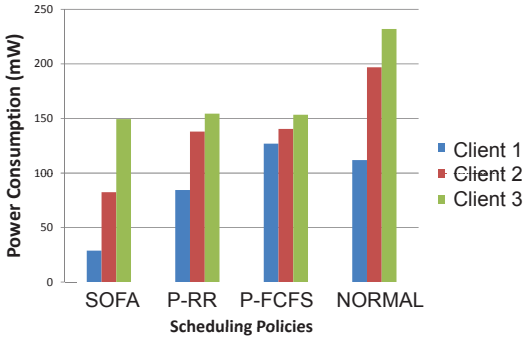


Figure 5.6: Comparing the power consumption of three clients under four scheduling policies.

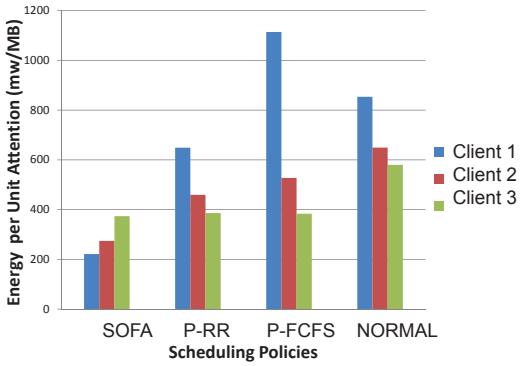


Figure 5.7: Comparing the Energy-per-Unit-Attention of three clients under four scheduling policies.

the buffered PSM packets are transmitted immediately by queueing the packets in a separate transmission queue with a higher priority. When there are multiple PSM clients, P-RR schedules their packet deliveries in Round-Robin manner, while P-RR schedules them in FCFS manner. Figure 5.6 compares the energy consumptions of three clients. Because there is no uplink traffic, and the energy the node spent in sleep state is almost negligible, therefore the client's energy consumption mostly comes from receiving packets, whether destined to it or not. Consequently, the energy saving comes mostly from putting the client to sleep more. Figure 5.7 compares the *energy per unit attention* of three clients, which is the energy for the client to receive 1MB downlink data. Here we replace airtime with throughput since they are equivalent in a single rate WLAN. Clearly SOFA outperforms the other three polices. SOFA especially helps client 1 which has the least attention request by reducing its EUA by over 2/3 compared with any other scheduling algorithm. This is because SOFA favors clients with smaller attention demand.

5.7 Conclusion

In this chapter we propose a downlink traffic scheduler on the AP of a WLAN, called SOFA, which helps its PSM clients save energy by allowing them to sleep more, hence increase battery lives. If a client has buffered packets at the AP in a beacon period, and that client decides to receive it, it has to remain awake from the beginning of the beacon period till the last packet scheduled for it in the beacon period is delivered. Therefore a large portion of energy wastage (for the client) comes from the AP transmitting other clients'a packets before it finishes transmitting the client's last packet to it. SOFA manages to reduce such energy wastage and maximizes the total sleep time of all clients. SOFA favors clients with smaller attention requests by allowing them to spend less energy to get one unit of attention, while still helping other clients with larger

attention requests to sleep more compared with other popular scheduling policies like round-robin and FCFS. Furthermore, SOFA enforces attention fairness among clients and avoids unnecessary beyond beacon period deferral of packets. SOFA is practical, and its stability and performance have been rigorously proved.

Chapter 6

Conclusion

In this thesis, we have studied three problems in wireless LANs: cross-layer design of the PHY and MAC layers, mechanisms for implicit coordination among clients, and packet dispatching on access points (APs). For the first two problems, our goal is to increase system throughput and decrease MAC layer delay. Our study of the third problem is targeted at better preserving the battery of mobile devices.

With respect to the first problem, we design a novel Packet Corruption Classification (PCC) to diagnose packet corruption in OFDM wireless networks by statistically analyzing certain available physical layer information, based on the observation that different causes of corruption result in different per-symbol-SINR patterns within a packet. Our approach introduces no additional traffic overhead. It can achieve a low miss rate (of interference) of 6%, with the false positive rate threshold set to 5%, even under the scenario where interference is hardest to detect. Since all experimental results and conclusions are based on experiments conducted on a real test bed, we expect that the implications of our results can be very useful in upper-layer problem domains, such as link scheduling, channel management, packet recovery design, etc., where a better understanding of the link behavior can greatly help.

For the second problem, we address the issue of improving the system throughput under high congestion. We first derive an analytical model in order to better understand the efficiency of 802.11 DCF protocol. Then we design CHAIN, a Coordinated Heavy-traffic efficient Access scheme, mainly to improve the efficiency of uplink traffic. By improving the uplink efficiency, it also indirectly improves the down-link throughput. The key idea in CHAIN is a novel *piggyback transmission opportunity*. Clients maintain a precedence relation among themselves, and a client can immediately transmit a new packet after it overhears a successful transmission of its predecessor, without going through the regular contention process. When the network load is low, CHAIN behaves similar to DCF; but when the network becomes congested, clients automatically start chains of transmissions to improve efficiency. CHAIN is derived from DCF and co-exists in a friendly manner with it. Moreover, it retains all the advantages of the 802.11 DCF standard - simplicity, robustness, and scalability.

With respect to the third problem, we propose Sleep-Optimal Fair-Attention Scheduler (SOFA), a down-

link traffic scheduler for the AP that achieves *system sleep-optimality*, and *energy fairness*, while meeting an *attention fairness constraint*, and a *non-beyond beacon period deferral constraint*. Further, SOFA is analytically proved to be stable.

In this thesis, our major contribution is the protocols themselves. We focus on the correctness, feasibility, fairness and performance gains of the protocols, which are analyzed both in theory and through experiments/simulations.

We believe that our work can potentially be applied to help solve other problems in WLANs. For example, the key ideas of CHAIN, the overhearing and piggyback precedence relations, can help solve exposed terminal problems. Suppose clients A and B are exposed terminals, and A and B are associated with AP1 and AP2, respectively. With DCF, they cannot transmit at the same time due to mutual carrier sensing. With CHAIN, however, this issue can be resolved with the help of a third client C, whose transmissions can be overheard by both A and B. Now, AP1 and AP2 can assign both A and B to be C's follower. This way, A and B can transmit at the same time with piggyback transmissions. Similarly, although SOFA studies packet scheduling on a single AP, it can be extended to multiple APs considering that modern enterprise WLANs have centralized control to coordinate the behavior of all the APs. The rigorous analysis of the attention assignment algorithm in SOFA can potentially provide insights for future packet scheduler designs in WLANs.

References

[1] E. Inc. Universal software radio peripheral. *"http://ettus.com."*.

[2] GNUradio wiki. *"http://gnuradio.org/trac/"*.

[3] http://ns-2.blogspot.com/2008/09/wireless-lan-power-management-extension.html.

[4] http://www.merunetworks.com/.

[5] The network simulator - ns-2. In *http://www.isi.edu/nsnam/ns/*.

[6] Ieee standard for wireless lan medium access control (mac) and physical layer (phy) specifications. ISO/IEC 802-11: 1999(E), Augest 1999.

[7] G. Anastasi A, M. Conti B, E. Gregori B, and A. Passarella B. 802.11 Power-Saving Mode for Mobile Computing in Wi-Fi hotspots: Limitations, Enhancements and Open Issues. In *Wireless Netw*, 2008.

[8] P. Agrawal, A. Kumar, J. Kuri, M. K. Panda, V. Navda, R. Ramjee, and V. N. Padmanabhan. Analytical models for energy consumption in infrastructure WLAN STAs carrying TCP traffic. In *COMSNETS*, 2010.

[9] P. Agrawal, A. Kumar, J. Kuri, M.K. Panda, V. Navda, and R. Ramjee. OPSM - Opportunistic Power Save Mode for Infrastructure IEEE 802.11 WLAN. In *Communications Workshops (ICC)*, 2010.

[10] G. Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. on Select. Areas in Commun.*, 18(3), 2000.

[11] S. Biswas and S. Datta. Reducing Overhearing Energy in 802.11 Networks by Low-Power Interface Tuning. In *International Conference on Performance, Computing and Communications*, 2004.

[12] Gurashish Brar, Douglas M. Blough, and Paolo Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *MobiCom '06*, pages 2–13, New York, NY, USA, 2006. ACM.

[13] Arvind Krishna Bruce Hajek and Richard O. LaMaire. On the capture probability for a larger number of stations. *IEEE Trans. on Commun.*, 45(2):254–260, 1997.

[14] F. Cali, M. Conti, and E. Gregori. Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit. *IEEE/ACM Trans. on Networking*, 8(6), December 2000.

[15] F. Cali, M. Conti, and E. Gregori. IEEE 802.11: design and performance evaluation of an adaptive backoff mechanism. *IEEE J. on Select. Areas in Commun.*, 18(9), September 2000.

[16] Surendar Chandra and Amin Vahdat. Application-specific Network Management for Energy-Aware Streaming of Popular Multimedia Formats. In *Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference*, 2002.

[17] R. W. Chang. Synthesis of band-limited orthogonal signals for multi-channel data transmission. *Bell System Technical Journal 46*, 1966.

[18] J. Kim et al. Cara: Collision-aware rate adatpion for ieee 802.11 wlans. In *IEEE INFOCOM*, 2006.

[19] Yan Gao, Jennifer C. Hou, and Hoang Nguyen. Topology control for maintaining network connectivity and maximizing network capacity under the physical model. In *INFOCOM 2008*, 2008.

[20] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: combating hidden terminals in wireless networks. *SIGCOMM 08*, 2008.

[21] A. Grilo and M. Nunes. Performance evaluation of IEEE 802.11 e. In *Proc. IEEE PIMRC*, 2002.

[22] Matthias Grossglauser and David Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proc. of IEEE INFOCOM*, 2001.

[23] Yong He and Ruixi Yuan. A Novel Scheduled Power Saving Mechanism for 802.11 Wireless LANs. *IEEE Transactions on Mobile Computing*, 2009.

[24] Yong He, Ruixi Yuan, Xiaojun Ma, and Jun Li. Analysis of the impact of background traffic on the performance of 802.11 power saving mechanism. *Comm. Letters.*, March 2009.

[25] Yong He, Ruixi Yuan, Xiaojun Ma, Jun Li, and Charles Wang. Scheduled PSM for Minimizing Energy in Wireless LANs. *Network Protocols, IEEE International Conference on*, 2007.

[26] 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. *SIGCOMM Comput. Commun. Rev.*, 35(4), 2005.

[27] Kyle Jamieson and Hari Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *ACM SIGCOMM*, Kyoto, Japan, August 2007.

[28] Kyle Jamieson, Bret Hull, Allen K. Miu, and Hari Balakrishnan. Understanding the real-world performance of carrier sense. In *ACM SIGCOMM Workshop on Experimental Approaches to Wireless Network Design and Analysis (E-WIND)*, Philadelphia, PA, August 2005.

[29] L. Lily Yang Jing Zhu, Xingang Guo and W. Steven Conner. Leveraging spatial reuse in 802.11 mesh networks with enhanced physical carrier sensing. In *IEEE International Conference on Communications*, June 2004.

[30] Tarun Joshi, Anindo Mukherjee, Younghwan Yoo, and Dharma P. Agrawal. Airtime Fairness for IEEE 802.11 Multirate Networks. *IEEE Transactions on Mobile Computing*, 7:513–527, 2008.

[31] Jsim. http://j-sim.cs.uiuc.edu/.

[32] Eun-Sun Jung and Nitin H. Vaidya. An energy efficient mac protocol for wireless lans. In *INFOCOM*, 2002.

[33] H. Kim and J. Hou. Improving protocol capacity with model-based frame scheduling in ieee 802.11-operated wlans. In *Mobicom*, 2003.

[34] Hwangnam Kim, Jennifer C. Hou, Chunyu Hu, and Ye Ge. Qos provisioning for ieee 802.11-compliant wireless networks: past, present, and future. *Computer Networks Journal*, 2006. (to appear).

[35] Ronny Krashinsky and Hari Balakrishnan. Minimizing Energy for Wireless Web Access Using Bounded Slowdown. In *ACM MOBICOM*, 2002.

[36] A. Kumar, E. Altman, D. Miorandi, and M. Goyal. New insights from a fixed point analysis of single cell IEEE 802.11 WLANs. In *Proc. of IEEE INFOCOM*, April 2005.

[37] Younggoo Kwon, Yuguang Fang, and Haniph Latchman. Design of mac protocols with fast collision resolution for wireless local area networks. *IEEE Transactions on Wireless Communications*, 3:793–807, 2004.

[38] Gianfranco Ciardo Lawrence, Lawrence M. Leemis, and David Nicol. On the minimum of independent geometrically distributed random variables, 1994.

[39] Jeongjoon Lee, Catherine Rosenberg, and Edwin K. P. Chong. Energy efficient schedulers in wireless networks: design and optimization. *Mob. Netw. Appl.*, 11, June 2006.

[40] W. C. Y. Lee. Elements of cellular mobile radio systems. *IEEE Trans. on Vehicular Technology*, VT-35(2):48–56, May 1986.

[41] Hsiao-Po Lin, Shih-Chang Huang, and Rong-Hong Jan. A power-saving scheduling for infrastructure-mode 802.11 wireless LANs. *Comput. Commun.*, 29, November 2006.

[42] Kate C. Lin, Nate Kushman, and Dina Katabi. Ziptx: Harnessing partial packets in 802.11 networks. In *Mobicom'08*, September 2008.

[43] Lucent. IEEE802.11 WaveLAN PC Card - Users Guide .

[44] K. Medepalli and F.A. Tobagi. Towards performance modeling of IEEE 802.11 based wireless networks: A unified framework and its applications. In *Proc. of IEEE INFOCOM*, April 2006.

[45] Daji Qiao and Kang G. Shin. Smart Power-Saving Mode for IEEE 802.11 Wireless LANs. In *IEEE INFOCOM*, 2005.

[46] Lili Qiu, Yin Zhang, Feng Wang, Mi Kyung Han, and Ratul Mahajan. A general model of wireless interference. In *MobiCom '07*, pages 171–182, New York, NY, USA, 2007. ACM.

[47] Hariharan Rahul, Nate Kushman, Dina Katabi, Charles Sodini, and Farinaz Edalat. Learning to share: narrowband-friendly wideband networks. *SIGCOMM Comput. Commun. Rev.*, 38(4):147–158, 2008.

[48] Shravan Rayanchu, Arunesh Mishra, Dheeraj Agrawal, Sharad Saha, and Suman Banerjee. Diagnosing Wireless Packet Losses in 802.11: Separating Collision from Weak Signal. In *IEEE INFOCOM*, Phoenix, AZ, April 2008.

[49] Lamia Romdhani and Christian Bonnet. Performance analysis and optimization of the 802.11e edca transmission opportunity (txop) mechanism. *WiMOB*, 0:68, 2007.

[50] Eric Rozner, Vishnu Navda, Ramachandran Ramjee, and Shravan K. Rayanchu. NAPman: network-assisted power management for wifi devices. In *MobiSys*, 2010.

[51] Bahareh Sadeghi, Vikram Kanodia, Ashutosh Sabharwal, and Edward W. Knightly. Oar: An opportunistic auto-rate media access protocol for ad hoc networks. *Wireless Networks*, 11(1-2):39–53, 2005.

[52] T. M. Schmidl and D. C. Cox. Robust frequency and timing synchronization for ofdm. *IEEE Trans. Communications*, 45(12):1613–1621, Dec. 1997.

[53] Gaurav Sharma, Ayalvadi Ganesh, and Peter Key. Performance analysis of contention based medium access control protocols. *IEEE Trans. Inf. Theor.*, 55(4):1665–1682, 2009.

[54] 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 *MobiCom*, 2009.

[55] D. Skordoulis, Qiang Ni, Hsiao-Hwa Chen, A.P. Stephens, Changwen Liu, and A. Jamalipour. Ieee 802.11n mac frame aggregation mechanisms for next-generation high-throughput wlans. *Wireless Communications, IEEE*, 15 Issue:1, 2008.

[56] Mark Stemm, Randy H. Katz, and Y H. Katz. Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices. In *IEICE Transactions on Communications*, 1997.

[57] John A. Stine and Gustavo De Veciana. A comprehensive energy conservation solution for mobile ad hoc networks. In *IEEE International Communication Conference*, 2002.

[58] Enhua Tan, Lei Guo, Songqing Chen, and Xiaodong Zhang. PSM-throttling: Minimizing Energy Consumption for Bulk Data Communications in WLANs. *Network Protocols, IEEE International Conference on*, 2007.

[59] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine-grained channel access in wireless lan. *SIGCOMM Comput. Commun. Rev.*, 2010.

[60] Y. C. Tay, Kyle Jamieson, and Hari Balakrishnan. Collision-minimizing csma and its applications to wireless sensor networks. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 22, 2004.

[61] Nitin H. Vaidya, Jennifer Bernhard, V. V. Veeravalli, P. R. Kumar, and R. K. Iyer. Illinois wireless wind tunnel: a testbed for experimental evaluation of wireless networks. In *E-WIND '05: Proceedings of the 2005 ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis*. ACM, 2005.

[62] Patrick Verkaik, Yuvraj Agarwal, Rajesh Gupta, and Alex C. Snoeren. Softspeak: making voip play well in existing 802.11 deployments. In *NSDI'09*.

[63] Y. Xiao and J. Rosdahl. Throughput and delay limits of ieee 802.11. *IEEE Commun. Lett*, 6:355C357, 2002.

[64] Yang Xiao. Ieee 802.11 performance enhancement via concatenation and piggyback mechanisms. *IEEE Trans. on Wireless Comm.*, 4, 2005.

[65] Yi Xie, Xiapu Luo, and Rocky K. C. Chang. Centralized PSM: an AP-centric power saving mode for 802.11 infrastructure networks. In *Proceedings of the 32nd international conference on Sarnoff symposium*, 2009.

[66] Xue Yang and Nitin H. Vaidya. A wireless mac protocol using implicit pipelining. *IEEE Trans. on Mobile Computing*, 5(3), 2006.

[67] Xue Yang and Nitin H. Vaidya. On the physical carrier sense in wireless ad hoc networks. In *Proc. of IEEE INFOCOM*, 2005.

[68] Ji-Hoon Yun and Seung-Woo Seo. Collision detection based on rf energy duration in ieee 802.11 wireless lan. In *Comsware*, 2006.

[69] H. Zhai and Y. Fang. Physical carrier sensing and spatial reuse in multirate and multihop wireless ad hoc networks. In *Proc. of IEEE INFOCOM*, 2006.

[70] H. Zimmermann. OSI reference model–the iso model of architecture for open systems interconnection. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 28(4):425–432, 1980.