

# **Investigating Power Consumption in 802.11 WLANs: Measurement, Visualization, and Improvement**

**By  
Meng Li**

**Thesis submitted in Partial Fulfillment of the  
Requirements for the Degree Master of Technology in  
Information and Communication Technology**

**Faculty of Engineering and Science  
University of Agder**

**Grimstad  
May 2009**

# Abstract

802.11 based wireless local area network (WLAN) has been increasingly supported by handheld devices. However, supporting WLAN functionality is extremely energy consuming since the connectivity has to be maintained even when the device is in an idle state. The power management specified in the 802.11 standards do not specify detailed techniques to handle the problem caused by power consumption affecting factors (PCAFs), since most of these factors have not been specified in the standards.

The objective of this master dissertation is to investigate how different PCAFs affect the power consumption in 802.11 WLANs.

As a starting point, several PCAFs are identified and investigated, including the beacon period, the beacon size, the signal strength, the foreground traffic, the background traffic, and the delivery traffic indication map (DTIM). Following the test instructions specified for both an idle mode and a traffic mode, measurement data for these two modes have been collected from the test-bed. According to the experimental study, it has been shown that all the PCAFs have different qualitative and quantitative effects on the power consumption of 802.11 WLANs. For example, the uplink traffic mode consume more energy than the downlink one. The background traffic eliminates the effectiveness of the power management the most among all PCAFs.

By making use of the findings from the experimental study combined with the concept of 802.11 theory, analytical expressions for calculating the power consumption under a 802.11 legacy polling scheme are derived. The accuracy of the obtained equations have been validated by the experimental data measured from the test-bed.

Moreover, a WLAN Power visualization tool (WPVT) has been implemented by Java, which can display theoretical power consumption curves for both an idle mode and a traffic mode if PCAF configurations are fixed. All theoretical curves are determined according to the derived analytical formula. Different with the traditional way where a test-bed, hardware configurations, multimeter testing, etc., are required, the WPVT only needs simple PCAF configurations. In a sense, the WPVT facilitates the investigation of

---

PCAFs in 802.11 WLANs due to the fact of less time consuming as compared with the traditional experimental study.

Finally, a new power saving algorithm has been proposed, which has taken the time drift of the synchronization clock between an access point (AP) and a station (STA) into account. Probability density function (PDF) curves of the power consumption have been presented and simulated for both the Legacy polling scheme and the new mechanism. It turns out that the new mechanism outperforms the Legacy polling scheme with respect to the power consumption.

The investigation presented in this dissertation provides first-hand empirical results and guideline for studying PCAFs in the 802.11 WLANs. Both the methodologies and the investigation results can supplement existing prior research work related to power conservation topics. This is the main potential outcome through the thesis, which can also act as the main contribution to knowledge.

# Preface

This report documents a Master's thesis in Information and Communication Technology, Faculty of Engineering and Science, University of Agder Norway. The master project has been carried out from 1st January to 25th May 2009.

I would especially like to thank my principal supervisor Prof. Frank Yong Li at University of Agder, for the insightful guidance of my work, active involvement, and helpful support at all times during the project.

The thesis has been initiated by Ericsson, which has provided technical supervision, lab, and testing equipments. My particular thanks go to all of my supervisors at EMP, Paulo Pamplona, Kim Tommy Humborstad, and Arild Løvendahl, for their insightful support, many fruitful discussions, and successful cooperation.

Special thanks to my parents, sister, and my girlfriend Yuanyuan Ma for supporting in every step of my life, having confidence in me and praying for my success all the time and thereby making the master project possible.

Grimstad, 25 May, 2008

Meng Li

# Contents

<b>Abstract</b>	<b>i</b>
<b>Preface</b>	<b>iii</b>
<b>Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Problem statement . . . . .	2
1.3 Motivation . . . . .	2
1.4 Research approach . . . . .	3
1.5 Limitations and assumptions . . . . .	4
1.6 Dissertation outline . . . . .	5
<b>2 Theoretical Background</b>	<b>6</b>
2.1 Wireless local area network . . . . .	6
2.1.1 Introduction . . . . .	6
2.1.2 WLAN mode . . . . .	7
2.2 WLAN standard . . . . .	7
2.2.1 802.11 . . . . .	7
2.2.2 802.11a . . . . .	7
2.2.3 802.11b . . . . .	7
2.2.4 802.11g . . . . .	8
2.2.5 802.11n . . . . .	8
2.2.6 802.11e . . . . .	8
2.3 Power management . . . . .	9

## CONTENTS

---

2.3.1	802.11 legacy polling scheme . . . . .	9
2.3.2	APSD . . . . .	10
2.3.3	PSMP . . . . .	11
2.4	Related work . . . . .	11
<b>3</b>	<b>Experimental Study on PCAF investigation</b>	<b>14</b>
3.1	Test-bed structure . . . . .	14
3.2	Definition for testing scenario . . . . .	16
3.2.1	WLAN testing mode . . . . .	16
3.2.2	PCAF . . . . .	17
3.2.3	Testing instruction for the idle mode . . . . .	17
3.2.4	Testing instruction for the traffic mode . . . . .	21
3.3	Testing results . . . . .	26
3.3.1	Beacon period . . . . .	26
3.3.2	DTIM . . . . .	27
3.3.3	Beacon size . . . . .	28
3.3.4	Signal strength . . . . .	30
3.3.5	Foreground traffic . . . . .	31
3.3.6	Background traffic . . . . .	32
<b>4</b>	<b>WLAN Power Consumption Calculation</b>	<b>42</b>
4.1	Power calculation in the idle mode . . . . .	42
4.2	Power calculation in the traffic mode . . . . .	46
<b>5</b>	<b>WLAN Power Visualization Tool</b>	<b>50</b>
5.1	Requirement analysis . . . . .	50
5.1.1	Functional requirement . . . . .	50
5.1.2	Non-functional requirement . . . . .	51
5.2	Design specification . . . . .	52
5.3	Implementation . . . . .	55
5.4	Testing . . . . .	58
5.4.1	Testing of WPVT for the idle mode . . . . .	58
5.4.2	Testing of WPVT for the traffic mode . . . . .	60
5.4.3	Validation of configuration . . . . .	61
<b>6</b>	<b>The Proposed Algorithm for Power Saving in WLANs</b>	<b>62</b>
6.1	Background . . . . .	62
6.2	Proposed algorithm . . . . .	63

## CONTENTS

---

6.3	Performance evaluation . . . . .	65
6.3.1	Time drift analysis . . . . .	65
6.3.2	Mathematical argument . . . . .	66
<b>7</b>	<b>Discussions</b>	<b>71</b>
7.1	Search and identify PCAFs . . . . .	71
7.2	Experimental study . . . . .	72
7.3	WPVT . . . . .	73
7.4	New proposal . . . . .	73
<b>8</b>	<b>Conclusions and Future Work</b>	<b>75</b>
8.1	Conclusion . . . . .	75
8.2	Contributions . . . . .	77
8.3	Further Work . . . . .	77
<b>A</b>	<b>Agilent Script</b>	<b>79</b>
<b>B</b>	<b>Cisco AP Configuration File</b>	<b>81</b>
	<b>Bibliography</b>	<b>88</b>

# Abbreviations

AC:	access category
ACK:	acknowledgement
AP:	access point
$AP_{REF}$ :	access point associated by the reference phone
$AP_{DUT}$ :	access point associated by DUT
APSD:	automatic power saving delivery
BSS:	basic service set
$BG_{\alpha}$ :	background traffic in the same BSS
$BG_{\beta}$ :	background traffic in the different BSS, and channel interval = 0
$BG_{\gamma 1}$ :	background traffic in the different BSS, and channel interval = 1
$BG_{\gamma 2}$ :	background traffic in the different BSS, and channel interval = 2
$BG_{\gamma 3}$ :	background traffic in the different BSS, and channel interval = 3
CTS:	clear to send
CW:	contention window
IBSS:	independent basic service set
JAR:	Java archive
MVC:	model view controller
$M_{BG}$ :	mode of background traffic
$M_{FG}$ :	mode of foreground traffic
DUT:	device under test
DIFS:	distributed coordinate function interframe space
DTIM:	delivery traffic indication message
PCAF:	power consumption affecting factor
$Phone_{REF}$ :	reference phone
PSMP:	power save multiple poll
QOS:	quality of service
$R_U$ :	data rate of an uplink traffic
$R_D$ :	data rate of an downlink traffic



## CONTENTS

---

RF:	radio frequency
RTS:	request to send
SIFS:	short interframe space
STA:	station
SYN:	synchronization
TIM:	traffic indication message
UML:	united modeling language
WLAN:	wireless local area network
WMM:	Wi-Fi multimedia
WNIC:	wireless network interface card
WPVT:	WLAN power visualization tool

# Symbols

$I_{channel}$ :	channel interval
$I_{wlan}$ :	current through a WLAN chipset
$P_{BL}$ :	beacon frame lost probability
$P_{downlink}$ :	power consumed by a receiver during a traffic testing scenario
$P_{idle}$ :	power for an idle state
$P_{idleBL}$ :	power for an idle state considering beacon losses
$P_{receiver}$ :	power consumed by a receiver during an atomic interaction
$P_{transmitter}$ :	power consumed by a transmitter during an atomic interaction
$P_{sleep}$ :	power for a sleep state
$P_{test}$ :	power shown on the multimeter
$P_{wake}$ :	power for an awake state
$P_{wlan}$ :	power consumed by a WLAN chipset
$R_{measure}$ :	resistance of the attached resistor
$R_{phone}$ :	resistance of DUT
$R_{PHY}$ :	physical link data rate
$R_{wlan}$ :	resistance of a WLAN chipset
$S_{beacon}$ :	size of a beacon frame
$S_{header}$ :	size of a MAC header
$S_{PHY}$ :	size of a physical frame
$S_{SYN}$ :	size of physical synchronization
$T_{ACK}$ :	time for one ACK
$T_{beaconpd}$ :	beacon period
$T_{BL}$ :	time in the awake state for one waking up due to a beacon lost
$T_{CTS}$ :	time for one CTS
$T_{drift}$ :	time drift of the SYN clock
$T_{duration}$ :	time of one testing scenario
$T_{Frame}$ :	time for transmitting a frame
$T_{idle}$ :	time of the idle state during a traffic testing scenario

## CONTENTS

---

$T_{normal}$ :	time in the awake state for one normal waking up
$T_{partial}$ :	time for parsing a beacon frame
$T_{RTS}$ :	time for one RTS
$T_{RX.R}$ :	time for receiving of a receiver during an atomic interaction
$T_{RX.T}$ :	time for receiving of a transmitter during an atomic interaction
$T_{SIFS}$ :	time for one SIFS
$T_{traffic}$ :	time of the active state during a traffic testing scenario
$T_{TX.T}$ :	time for transmitting of a receiver during an atomic interaction
$T_{TX.T}$ :	time for transmitting of a transmitter during an atomic interaction
$T_{wake}$ :	total time for the awake state during one test duration
$T_{wakepd}$ :	period of waking up
$T_{wakesingle}$ :	time in the awake state for one waking up
$U_{input}$ :	input voltage of the test-bed
$U_{multimeter}$ :	voltage value shown on the multimeter
$U_{wlan}$ :	voltage of a WLAN chipset

# List of Figures

2.1	Unicast buffering transmission [1]. . . . .	9
2.2	Broadcast and multicast buffering transmission [1]. . . . .	10
3.1	Test bed structure. . . . .	14
3.2	Simplified measurement circuit. . . . .	16
3.3	Power as a function of the beacon period in the idle mode. . . . .	26
3.4	Power as a function of the beacon period in the traffic mode. . . . .	27
3.5	Power as a function of the DTIM in the idle mode. . . . .	28
3.6	Power as a function of the DTIM in the traffic mode. . . . .	28
3.7	Power as a function of the beacon size in the idle mode. . . . .	29
3.8	Power as a function of the signal strength in the idle mode. . . . .	30
3.9	Power as a function of the signal strength in the traffic mode. . . . .	31
3.10	Power as a function of the foreground traffic. . . . .	31
3.11	Power as a function of the $BG_\alpha$ in the idle mode. . . . .	33
3.12	Power as a function of the $BG_\beta$ in the idle mode. . . . .	33
3.13	Power as a function of the $BG_{\gamma_1}$ in the idle mode. . . . .	34
3.14	Power as a function of the $BG_{\gamma_2}$ in the idle mode. . . . .	35
3.15	Power as a function of the $BG_{\gamma_3}$ in the idle mode. . . . .	35
3.16	Power as a function of the downlink background traffic in the idle mode. . . . .	36
3.17	Power as a function of the uplink background traffic in the idle mode. . . . .	36
3.18	Power as a function of the $BG_\alpha$ in the traffic mode. . . . .	37
3.19	Power as a function of the $BG_\beta$ in the traffic mode. . . . .	38
3.20	Power comparison ( $R_U = 2335\text{ kbps}$ and $M_{BG} = \text{downlink}$ ). . . . .	38
3.21	Power comparison ( $R_U = 3523\text{ kbps}$ and $M_{BG} = \text{downlink}$ ). . . . .	39
3.22	Power comparison ( $R_U = 2335\text{ kbps}$ and $M_{BG} = \text{uplink}$ ). . . . .	40
3.23	Power comparison ( $R_U = 3523\text{ kbps}$ and $M_{BG} = \text{uplink}$ ). . . . .	40
4.1	Interaction between a STA and an AP in the idle mode. . . . .	42

## LIST OF FIGURES

---

4.2	Validation of the theoretical calculation in the idle mode. . . . .	46
4.3	The atomic interactions between a transmitter and a receiver. . . . .	47
4.4	Validation of the theoretical calculation for the traffic pattern. . . . .	49
5.1	MVC architecture [2]. . . . .	52
5.2	UML class diagram. . . . .	53
5.3	UML sequence diagram for the idle mode. . . . .	55
5.4	Model transform from the class diagram to java codes. . . . .	56
5.5	Add external jar library for JFreeChart. . . . .	57
5.6	Export an executable jar file using Fat Jar eclipse plug-in. . . . .	58
5.7	Testing for WPVT illustration in the idle mode. . . . .	59
5.8	Testing for WPVT illustration in the traffic mode. . . . .	60
5.9	Configuration validation. . . . .	61
6.1	Awake types of a STA in the 802.11 WLAN. . . . .	63
6.2	The flow chart of the proposed algorithm. . . . .	64
6.3	PDFs of the time drifts. . . . .	67
6.4	Comparison of the power PDFs. . . . .	69
7.1	Collision statistics in the traffic mode. . . . .	73

# List of Tables

3.1	Equipment list . . . . .	15
3.2	Default configuration of the test bed. . . . .	18
3.3	Description of <i>TC_1_01_Idle_BeaconPeriod</i> . . . . .	18
3.4	Description of <i>TC_1_02_Idle_DTIM</i> . . . . .	19
3.5	Description of <i>TC_1_03_Idle_BeaconSize</i> . . . . .	19
3.6	Description of <i>TC_1_04_Idle_SignalStrength</i> . . . . .	19
3.7	Description of <i>TC_1_05_Idle_BG<math>\alpha</math></i> . . . . .	20
3.8	Description of <i>TC_1_06_Idle_BG<math>\beta</math></i> . . . . .	20
3.9	Description of <i>TC_1_07_Idle_BG<math>\gamma</math></i> . . . . .	21
3.10	Description of <i>TC_2_01_Traffic_BeaconPeriod</i> . . . . .	22
3.11	Description of <i>TC_2_02_Traffic_DTIM</i> . . . . .	22
3.12	Description of <i>TC_2_03_Traffic_SignalStrength</i> . . . . .	23
3.13	Description of <i>TC_2_04_Traffic_BG<math>\alpha</math></i> . . . . .	23
3.14	Description of <i>TC_2_05_Traffic_BG<math>\beta</math></i> . . . . .	24
3.15	Description of <i>TC_2_06_Traffic_BG<math>\gamma</math></i> . . . . .	25
3.16	Description of <i>TC_2_07_Traffic_ForegroundTraffic</i> . . . . .	25
3.17	Beacon size samples . . . . .	29
3.18	Predefined data rate of background traffic . . . . .	32
4.1	Default values for the power calculation in the idle mode . . . . .	44
4.2	Default values for the power calculation in the traffic mode . . . . .	48
4.3	Dynamic parameters in the traffic mode . . . . .	48
5.1	Functions and descriptions of the Java classes . . . . .	54

# Chapter 1

## Introduction

### 1.1 Background

IEEE 802.11 WLAN achieves more recognition, due to its flexibility, mobility and scalability [1]. Flexibility lies in the easy changes between an infrastructure mode and an independent mode. A (STA) is no longer limited to a fixed position, as long as it is covered by a WLAN. Actually, a WLAN plays a role of a supplement to an existing wired networks. Moreover, the transmission speed in WLAN can reach up to 54Mbps or higher [1], which can be qualified for the normal access to resources. Finally, the popularity of portable devices, such as laptops, PDAs, cell phones, etc. enabled with WLAN functionality is another factor to push WLAN development forward.

On the other hand, battery power in a portable WLAN enabled device is a scarce resource. A WLAN chipset is extremely energy consuming, because once the wireless network interface is activated, the connectivity has to be maintained even if it's in idle state. Provided that a WLAN connectivity is maintained, the less power is used, the longer the normal functionalities can last. So power conservation has become a hot issue in the WLAN field.

As we know, the 802.11 standards have specified power management mechanisms. However, the standards themselves are not sufficient to handle WLAN power management problems, since in practice there are many factors that affect power consumption and these scenarios- or application- dependent factors are not specified in the standards. Moreover, the existence of diverse types of 802.11 standard amendments, chips, and application scenarios makes the problem more challenging.

Finally, the thesis proposal is initiated by Ericsson, due to the power consumption prob-

lem caused by WLAN functionality and insufficient scientific research on how a power consumption affecting factor (PCAF) affect power consumption.

## 1.2 Problem statement

The main task of this thesis is to investigate the PCAFs, which affect the efficiency of power consumption in 802.11 WLAN. To simplify the problem, only infrastructure mode is taken into account. A couple of sub-problems are derived from the main problem.

Firstly, search and identify power consumption factors in 802.11 WLANs, mainly based on 802.11g standard. Since PCAFs exist in the real WLAN activities whereas they are not taken into account in the specification of 802.11 standards, the deviation of power consumption between a practical testing and a theoretical estimate is inevitable. This specific problem is supposed to identify PCAF effect against the power conservation.

Secondly, based on a test-bed provided by Ericsson, the thesis has to investigate using real-life measurements how PCAFs quantitatively affect the power conservation. And both the idle mode and the traffic mode have to be taken into account.

Thirdly, a visualization tool has to be implemented in order to facilitate PCAF investigation. The tool is supposed to support power illustration for both the idle mode and the traffic mode, which diagrammatically illustrate power consumption levels. The basic logics are based on the WLAN power mathematical formula, derived from the second problem. It can save much more time than a real measurement to do PCAFs investigation over power consumption.

Finally, propose a new algorithm for improving power conservation performance in 802.11 WLANs. The main drawback of the 802.11 legacy polling scheme is that it can not be adjusted according to the time drift of the system synchronization (SYN) clock, which indicates more energy consumed, on the contrary, the new mechanism is supposed to improve power conservation with dynamical adaption to the time drift. Moreover, mathematical deductions are supposed to be used to prove the improvement of the new algorithm.

## 1.3 Motivation

WLAN increasingly becomes more popular due to its flexibility and the availability of WLAN on hand held devices. WLAN power conservation has become a tricky issue, due to the problem caused by the limitation of battery power. A cellular phone with WLAN disabled can last days while it can only last for hours if WLAN is always enabled. So



this thesis is supposed to supplement existing research in the field of 802.11 power conservation, which can finally benefit the end users by prolonging the battery duration.

The thesis is initiated by Ericsson, since WLAN functionality consumes more energy than the theoretical estimate, which means the power management doesn't function as expected. This thesis is tailored to investigate PCAFs. Based on this point, it is possible to utilize the existing resources and adjust different PCAFs to optimize the power conservation. Moreover, the power visualization tool can facilitate power consumption analysis with PCAFs. How each PCAF affecting energy conservation can be observed intuitively, by setting PCAFs with different values and illustrating power consumption.

Moreover, WiFi alliance [3], as a non-profitable organization, aims to certificate WLAN products based on 802.11. Power consumption is a hot issue focused by this organization. At present it is currently trying to estimate the power consumption of different STAs on the market through anonymous tests, based on which it will publish referenced power consumption measurements and environment factors to WLAN enabled device manufactures. As a regular member in WiFi alliance, Ericsson has to take responsibility to the test task. That is another reason why the project is derived. So this thesis exactly makes sense, since it may contribute to WiFi alliance activities.

### **1.4 Research approach**

Due to different characteristics of these sub-problems, a step-by-step research approach has to be adopted. Regarding the PCAFs investigation with an experimental study, a test-bed has to be constructed. And then according to testing scenarios, a series of test instructions are specified for PCAFs, based on which PCAF testing will be handled on the test-bed. To guarantee the validity of measurement results, each test case is repeated for ten times, followed by an average calculation. Furthermore, the measurement results, representing the relationship between PCAFs and the power consumption, are analyzed with curve figures, based on which a series of observation are concluded.

Secondly, the implementation of the power visualization tool is categorized into software development and the model driven development is adopted. A class diagram representing the static structure of the application is created using IBM Software Rational Architect[4], which can also achieves the transformation from the class diagram to the java source code. As long as the tool is implemented and verified, an executable JAR file will be exported for the portability.

Finally, in addition to the existing mechanisms and theoretical knowledge, based on the

experimental study and the derived mathematical formula, an adaptive algorithm is proposed, which mainly based on the time drift of the SYN clock. Based on analyzing the air logs during testing, the distribution of the time drift is obtained using Matlab. And then a series of mathematical calculations are derived to prove the improvement against the power conservation.

## 1.5 Limitations and assumptions

There are a series of limitations and assumptions during the problem solution, which need to be confirmed and mentioned firstly.

- It is assumed the test bed acts as the real WLAN. Because the testings are done in WLAN testing Lab, where the interferences from other existing WLANs extremely strong, so all the radio frequency connections between components are through shielded radio frequency (RF) cables. The signal transmitted along a shield cable has less attenuation than that through the real wireless fading channels.
- The testing results are based on power consumed by the WLAN chipset. In order to test the power of the chipset, a resistor is attached from device under test (DUT). Since the power consumed by the resistor is extremely low compared with the chipset, it is assumed that the negative effect derived from the resistor can be neglected. For details, refer to section 3.1.
- The configuration of DUT stays unchangeable for each testing scenario, so the WLAN chipset stays in a relatively static status. It is assumed that power consumed by the chip set is stable regardless of changes of temperature and humidity.
- There are a plenty of scenarios during real WLAN activities, but only the most common scenarios can be tested due to the limitation of the time and the necessity. Both the idle mode and the traffic mode have to be considered when creating scenarios. It is assumed that the testing scenarios are persuasive to demonstrate the generality of the testing results.
- Regarding the WLAN power visualization tool (WPVT), the background traffic option is based on the measurement results, which is obtained based on the default configuration. Although the configuration is changed due to the different configuration of parameters, it is assumed that the background traffic roughly has the same effect over the default configuration as that over other configurations with an acceptable extent.
- It is assumed that the investigation can be used in real life. So the investigation results can provide WLAN device manufactures with a reference and a guideline

to take the PCAFs into account in the development stage or before supporting new 802.11 standards.

## 1.6 Dissertation outline

The outline of the whole dissertation is presented below.

- Chapter 1 (this chapter) gives an introduction to the thesis. Based on the relevant background, the problem statement are presented where the main assumptions are discussed.
- Chapter 2 provides technical overview of 802.11 WLANs, including power management specification and the related work.
- Chapter 3 deals with the experimental study on power consumption. Based on the test-bed, the measurement results are collected and analyzed.
- Chapter 4 produces the mathematical formula for both the idle mode and the traffic mode, which have been validated with the experimental results.
- Chapter 5 describes the WLAN power consumption visualization tool in terms of the issues of the design, the implementation and the testing.
- Chapter 6 proposed a new algorithm, which focuses on the effect of the time drift. And the algorithm has been proved with mathematical deduction.
- Chapter 7 discusses the results and some tricky issues during the problem solutions.
- Chapter 8 presents the conclusion of the thesis and the further work.

## Chapter 2

# Theoretical Background

### 2.1 Wireless local area network

#### 2.1.1 Introduction

Before introduce the concept of 802.11 WLAN, several essential concepts have to be described firstly. A STA normally means a handheld computing device, equipped with at least one wireless network interface card (WNIC). In a WLAN, data is transmitted among stations. An access point (AP) acts as two roles in a WLAN. The first one is transferring packets to the destined stations, whereas the other one is that it connects a wireless network and a wired network. A basic service set (BSS) is a WLAN entity set, which is composed of multiple STAs and at most one AP.

A WLAN is the complement to the existing wired networks, but it can not replace the fixed networks. The flexibility can be indicated when deploying a WLAN by set up an AP and authorizing clients, instead of wire deployment. And the most obvious advantage provided by a wireless network is the mobility. Within the coverage of an AP, the wireless station can move without losing the connectivity. Roaming provide a more advanced mobility, which means the station moves to another BSS across the overlapped area without losing the connectivity [1].

A WLAN medium does not have an exact physical boundary. In a wired network, the signals are limited to a specific pathway defined by wire deployment. But, a WLAN has a more open medium, radio signals can transmit in all direction. This character is the basis for mobility, but it exerts a security issue on a WLAN. It is easier to intercept radio signals in such an open medium. So a WLAN has to be authenticated and the connection has to be encrypted to get rid of interception [1].

### **2.1.2 WLAN mode**

Normally, a BSS is either an independent BSS (IBSS) or an infrastructure BSS. The option depends on the existence of an AP in the BSS. An infrastructure BSS is composed by an AP and several STAs, and it can be defined by the distance between STAs and the AP, which means all STAs within the coverage of the AP. An AP in an infrastructure BSS plays a role of coordinator, which takes responsibility for broadcasting, transferring packets to destined STAs, connecting the WLAN with other types of network, such as an Ethernet, etc. A STA is not allowed to directly send packets to other STAs, since all the packets have to be routed via the AP.

On the contrary, an IBSS is composed of STAs without any AP. All the packets are directly sent by source STAs to destined STAs, so there is no coordination involved by an AP. Without a limitation of an AP, several STAs can associated with each other to construct a temporary WLAN for a specific purpose.

## **2.2 WLAN standard**

### **2.2.1 802.11**

“IEEE 802.11 is a set of standards carrying out wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands. They are implemented by the IEEE LAN/MAN Standards Committee (IEEE 802)” [5].

### **2.2.2 802.11a**

802.11a “specifies the PHY entity for an orthogonal frequency division multiplexing (OFDM) system and the additions that have to be made to the base standard to accommodate the OFDM PHY. The OFDM system provides a wireless LAN with data payload communication capabilities of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s. The support of transmitting and receiving at data rates of 6, 12, and 24 Mbit/s is mandatory” [6].

### **2.2.3 802.11b**

“This extension of the DSSS system builds on the data rate capabilities, as described in Clause 15 of IEEE Std 802.11, 1999 Edition, to provide 5.5 Mbit/s and 11 Mbit/s payload data rates in addition to the 1 Mbps and 2 Mbps rates. To provide the higher rates, 8-chip complementary code keying (CCK) is employed as the modulation scheme. The

chipping rate is 11 MHz, which is the same as the DSSS system described in Clause 15 of IEEE Std 802.11, 1999 Edition, thus providing the same occupied channel bandwidth. The basic new capability described in this clause is called High Rate Direct Sequence Spread Spectrum (HR/DSSS). The basic High Rate PHY uses the same PLCP preamble and header as the DSSS PHY, so both PHYs can co-exist in the same BSS and can use the rate switching mechanism as provided” [7].

### **2.2.4 802.11g**

“This clause specifies further rate extension of the PHY for the Direct Sequence Spread Spectrum (DSSS) system of Clause 15 and the extensions of Clause 18. The PHY defined in this clause will be known as the Extended Rate PHY (ERP). This PHY operates in the 2.4 GHz ISM band. The ERP draws from Clause 17 to provide additional payload data rates of 6, 9, 12, 18, 24, 36, 48, and 54 Mbit/s. Of these rates, transmission and reception capability for 1, 2, 5.5, 11, 6, 12, and 24 Mbit/s data rates is mandatory” [8].

### **2.2.5 802.11n**

“Clause 20 (High Throughput (HT) PHY specification ) specifies the PHY Entity for a high throughput (HT) orthogonal frequency division multiplexing (OFDM) system.” “The HT OFDM PHY is based on the OFDM PHY defined in Clause 17, with extensibility up to 4 spatial streams, operating in 20 MHz bandwidth. Additionally, transmission using 1 to 4 spatial streams is also defined for operation in 40 MHz bandwidth. These features are capable of supporting data rates up to 600 Mb/s” [9].

### **2.2.6 802.11e**

“This amendment defines the medium access control (MAC) procedures to support local area network (LAN) applications with quality of service (QoS) requirements. The procedures include the transport of voice, audio, and video over IEEE 802.11 wireless LANs (WLANs)” [10].

## 2.3 Power management

### 2.3.1 802.11 legacy polling scheme

802.11 legacy polling scheme is the most common and widely used specification for power management in 802.11 WLANs. Generally, in a 802.11 WLAN, a station stays in either in an active state or in a sleep state. The active state means the WNIC is powering on, and the STA can be involved in data transmission or reception, or even in idle state, whereas the sleep state indicates the power off of the WNIC, which can not sense the WLAN medium at all. A active state costs about 100 times more than a sleep state. So the scheme aims to control the transition between these state of a STA.

An AP plays an essential role in the scheme, and it take responsibility to keep track of all the associated STAs, buffer frames, and periodically broadcast the buffering information with beacon frames. Traffic Indication Map (TIM), a field in a beacon frame, is used for unicast transmission. Actually if the association ID of a STA is indicated in a TIM, that means there are frames buffered for the STA. All STAs have to periodically wake up to get TIMs. A STA has to use PS-Poll control frames to fetch the packet buffered. If several STAs want to retrieve their own packets at the same time, then back off mechanism is used to avoid collisions.

As shown in Fig 2.1, *STA1* has a listen interval of 2 while *STA2* has a interval of 3. In the first beacon interval, there is no frame buffered for *STA2*, so it moves to sleep state. In the second beacon interval, frames buffered for *STA1* will be retrieved with the PS-Poll. Finally, in the fourth interval, both *STA1* and *STA2* want to retrieve the buffered frames.

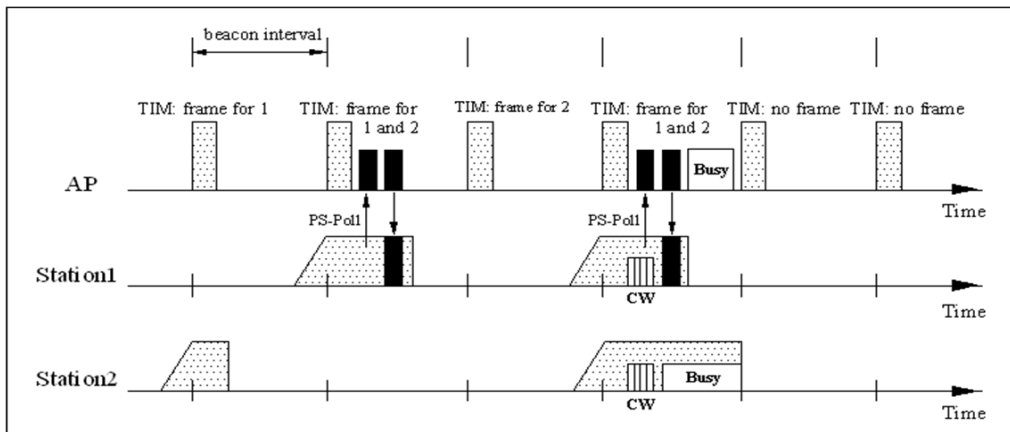


Figure 2.1: Unicast buffering transmission [1].

After a contention window, *STA1* wins, so it can fetch the buffered frames. If other STAs win during the following competition with *STA2*, it has to stay in the active mode until

next TIM.

In addition, regarding multicast or broadcast, frames with a group address can not rely on the poll mechanism. Delivery TIM (DTIM) is used to implement multicast and broadcast transmission in power save mode. DTIM is sent for a fixed number of beacon intervals. A station has to wake up for DTIM, and get broadcast or multicast packets if it belongs to the group address. The AP will handle the PS-Poll after broadcast and multicast are concluded. As shown in Fig. 2.2, DTIM is set to 3, so it occurs in every three beacon

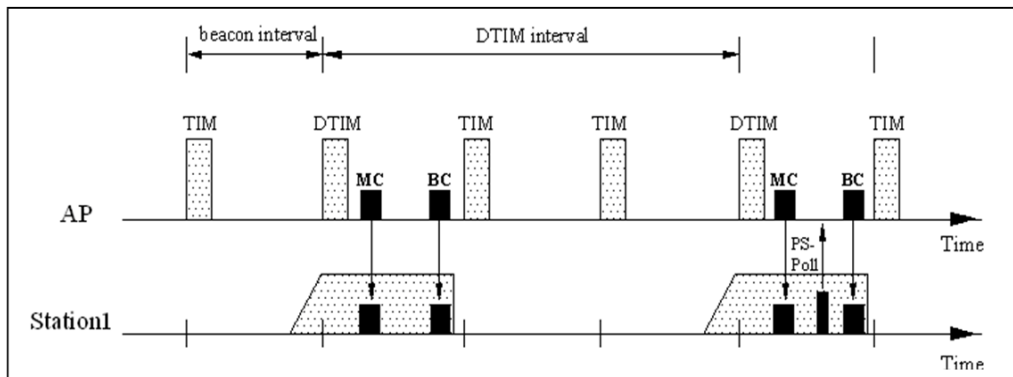


Figure 2.2: Broadcast and multicast buffering transmission [1].

intervals. In the second beacon interval, both multicast frames and broadcast frames are buffered, and will be retrieved after the DTIM. In the fifth interval, a unicast frame is also buffered, indicated by the PS-Poll, but it will be fetched after multicast and broadcast conclusion.

### 2.3.2 APSD

The 802.11 legacy polling scheme is not qualified to guarantee Quality of Service (QoS) in a 802.11 WLAN, which is the main issue discussed in 802.11e [10], so another power conservation protocol, Automatic Power Saving Delivery (APSD), is proposed in [10]. The APSD can be categorized into two types: Unscheduled APSD (UAPSD), and Scheduled APSD (SAPSD). As specified with UAPSD, a STA can fetch buffered frames from the AP by sending a trigger frame. After acknowledging the trigger frame, the AP releases what has been buffered. And then the STA can also send uplink data, followed by sleeping provided that there is no frames buffered in the AP. UAPSD is mainly used for WiFi Multimedia (WMM), where four access categories (AC) are defined, voice, video, best effort and background, descending priority sorted. AC can be configured into delivery enabled or trigger enabled. A STA has to send trigger enabled data to the AP in order to retrieve the delivery enabled data.



On the contrary, SAPSD requires both the AP and the STAs to negotiate with each other about the delivery time. So all the buffered frames have to be scheduled firstly, followed by delivery according to the designated time. If the negotiation is successful, the schedule element is sent by the AP to the STA, which specifies the starting time of service, the service duration and the service period. This mechanism requires resource reservation, which means if resource usage is predicable, it performs better than the legacy polling scheme, otherwise it greatly degrades.

### **2.3.3 PSMP**

Power Save Multiple Poll, specified in 802.11n [9], provides a more enhanced power conservation mechanism compared to APSD. In the same way, PSMP is also divided into two groups, *Unscheduled PSMP (UPSMP)* and *Scheduled PSMP (SPSMP)*, which rely on trigger and the pre-negotiated schedule, respectively. The normal PSMP sequence starts with PSMP frame, followed by a series of downlink frames, and ends with a couple of uplink frames. A PSMP frame includes fields to specify information of PSMP enabled STAs in the session, where the starting time and duration of both uplink transmission and downlink transmission for all the PSMP enabled STAs are defined. The strength of PSMP lies in the possibility to merge different kinds of application stream into a single stream, which obviously increases throughput.

## **2.4 Related work**

Regarding the experimental study over WLAN power consumption, a couple of practical experiments based on real life environments have been done in the prior research [11, 12, 13, 14]. Paper [11] discusses the energy consumption of Lucent IEEE 802.11 WLAN PC cards in an independent networking environment. At first, a linear equation is presented, where the coefficients are not determined. The testing scenarios considers different kinds of traffics, such as broadcast traffic, point to point traffic, discard traffic, etc. With the help of oscilloscope, the coefficients can be determined for each scenario, and then several linear equations are derived. Their work sets up an experimental basis for evaluating 802.11 power save mode (PSM) in independent mode. The approach in which they handled data and obtained several equations can be referenced.

Paper [12] studies the power consumption for the key components, including a WNIC, in a laptop. The way to collect testing data is direct measurement or calculation. The power usage is measured against different workload, which affects the power consump-

tion of different components. CPU is the main dominant factor confronted with heavy workloads, and the power management greatly decreases CPU power as expected. In addition, the display takes power consumption over when the CPU is in idle state.

White paper [13] investigates WLAN power consumption based on the experiments for diverse WNICs. The power consumption caused by a WLAN lies in not only the card itself but also other components operating on WNIC interfaces. The power conservation performance of WNIC lies in several factors, such as WLAN protocol, WLAN implementation, host CPU, etc.

Paper [14] studies multiple factors, such as beacon period, DTIM, background traffic, in an experimental method. It first compares PSM against continuous active mode (CAM) to validate the effect of PSM. Both these factors (DTIM, Beacon period) can affect the power consumption, and the effect is obvious when the value is small. And with the increase of the values, the effect will relieve quickly. Moreover, background traffic can relieve the effect of PSM. The main weak aspect is that the neighbor frequency interference is not taken into account

With regard to visualization tools in 802.11 WLAN field, WLANTV is presented in [15], which can be used to present both traffic patterns and load situation to facilitate the monitor of WLAN throughput. And it supports parsing both captured and live captured log files, based on Wireshark [16]. The precondition of the usage is that a sniffer hardware (a wireless network interface card and a corresponding driver) has to be installed. The parsing results are shown up using different charts, such as train charts, pie charts, information panels, etc. It provides a more efficient and accurate way to monitor WLAN activities compared to that in a manual way.

In addition, there are more publications regarding new power conservation proposals compared with the preceding topics. And these proposals are based on different OSI levels. Both [17] and [18] focus on the routing layer. [17] proposes a distributed and random mechanism, Span, for the multi-hop ad hoc wireless network. Whether a node is supposed to be sleep or awake depends on how many its neighbors can benefit from its awake state. And then a minimal routing path, which can guarantee the minimal power consumption, can be confirmed.

Moreover, in [18] a new mechanism, geographical adaptive fidelity (GAF), is present. The GAF classifies all the relevant nodes from a routing view, and then based on the classification all the unnecessary nodes will be shutdown to save more power.

On the contrary, other proposals provide power saving solutions in the MAC layer. Paper [19] proposed a new mechanism, namely a scheduled Power Save Mode. The new mech-

anism is based on time slicing, which schedules an AP to transmit data in a specific time slice. On the hand, STAs can flexibly adjust its own state. In this way, the scheduled PSM can minimize the negative effect of the background traffic, and optimize the power consumption to a theoretical level. According to the simulation results with NS, for all traffic scenarios, The conclusion is that the new mechanism can save more power than the unscheduled PSM, and it negatively affects one way delay.

Futhermore, [20] proposed a optimal version for Distributed Coordination Function (DCF) in 802.11, which enables ATIM windows with dynamical sizes. So a STA can be awake in only a part of a beacon period following ATIM windows, which means the STA spends more time in sleep compared with that specified in 802.11.

## Chapter 3

# Experimental Study on PCAF investigation

### 3.1 Test-bed structure

In order to investigate the PCAFs, a test-bed has to be constructed for the experimental study. As shown in Fig. 3.1, the experimental approach only takes infrastructure mode into account. And the test-bed is independent, so no connection is required to other existing networks.

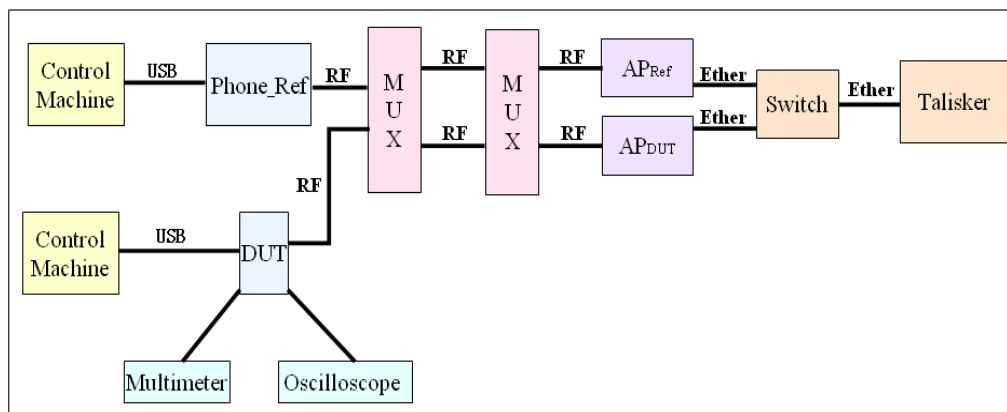


Figure 3.1: Test bed structure.

Generally, the equipments, shown in Fig. 3.1, are categorized into two groups, WLAN equipments and measurement equipments. The WLAN equipments are used to achieve WLAN communications and simulate WLAN activities, which mainly includes STAs and APs, whereas the measurement equipments aims to monitor and record run time values of power consumption. The equipments are listed in Tab.3.1, while the purposes and

Table 3.1: Equipment list

<i>Equipment</i>	<i>Type</i>	<i>OS</i>
Control Machine	HP Compaq nw8440	Windows 2000
Talisker	HP Compaq DC5750 GE020ET	Linux Fedora 6.0
$AP_{DUT}$	Cisco AIR-AP1242AG-E-K9	None
$AP_{REF}$	U.S.Robotics Wireless MAXg ADSL Gateway	None
DUT	Device under test	None
$Phone_{REF}$	Reference phone	None
Multimeter	Agilent 34411A	None
Oscilloscope	Tektronix MSO4054	None

remarks of the equipments are listed below.

- Talisker is a server. The first role played by the Talisker is a DHCP server, which can dynamically assign an IP address to DUT. Moreover it acts as an IP test server, which controls the connection to DUT, send/receive WLAN frames to/from DUT, and echo all messages back to the DUT.
- $AP_{DUT}$  is the AP associated with DUT, which participates in the foreground traffic transmission. The uplink packets sent by DUT reach Talisker through  $AP_{DUT}$ , and go back to DUT along a reverse path, whereas the downlink flow is triggered by a packet sent from DUT, and then Talisker sends downlink packets to the DUT through  $AP_{DUT}$ . [21] can be referenced to configure  $AP_{DUT}$ .
- $AP_{REF}$  is used to generate background traffics with  $Phone_{REF}$ . Since  $AP_{REF}$  is configurable, background traffics can be set with different channels, different data rate and different direction. So both the same frequency interference and the neighbor frequency interference can be simulated. It can be set up with the help of [22].
- Control machine is used to configure both STAs and APs. Moreover, traffic PCAFs are implemented using a specific testing framework, Atlet, installed on the machine. User-defined test cases are supported by the framework to achieve specific testing scenarios.
- MUX represents multiplexing, which is used to combines RF signals into a single signal stream.
- RF shielded cables are used to connect the WLAN equipments to avoid interferences generated by other existing WLANs, which are not under control. Along cables, attenuators are attached in order to adjust the WLAN signal strength.

- DUT attaches a resistor, which facilitates the measurement of WLAN power consumption. As shown in Fig. 3.2,  $R_{phone}$ ,  $R_{wlan}$  and  $R_{measure}$  denote the resistances of DUT, the WLAN chipset, and the resistor under test, respectively. The power of the WLAN chipset,  $P_{wlan}$ , is expressed as

$$\begin{aligned} P_{wlan} &= U_{wlan} \cdot I_{wlan} \\ &= U_{wlan} \cdot \frac{U_{multimeter}}{R_{measure}}, \end{aligned} \quad (3.1)$$

where  $U_{wlan}$  and  $I_{wlan}$  denote the voltage and the current of the WLAN chipset, whereas  $U_{multimeter}$  denotes the voltage obtained using the multimeter. Since  $R_{wlan} \gg R_{measure}$ , it is concluded that  $U_{wlan} \approx U_{input}$ , and we have

$$P_{wlan} = U_{input} \cdot \frac{U_{multimeter}}{R_{measure}}. \quad (3.2)$$

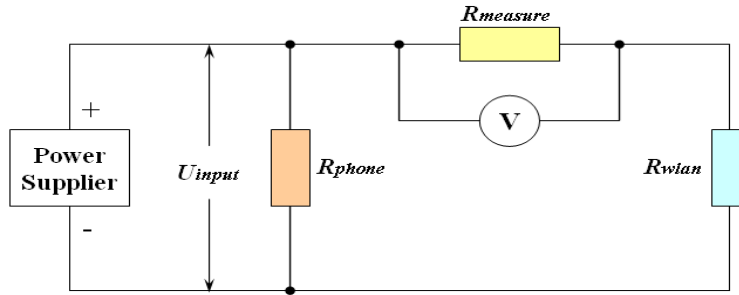


Figure 3.2: Simplified measurement circuit.

## 3.2 Definition for testing scenario

### 3.2.1 WLAN testing mode

Generally, the testing scenarios are roughly divided into two groups, the idle mode and the traffic mode. In the idle state, DUT is in the idle state and not involved in any traffic. It periodically wakes up to receive beacon frames sent by  $AP_{DUT}$  and check existence of any frame buffered. Since there is no buffered frames, it transits to sleep state immediately after beacon frames. And then the process repeats.

On the contrary during the traffic mode, DUT has to stay in active state to transmit/receive frames. A downlink traffic indicates that DUT has to stay in awake state as long as at least one frame is buffered for it. On the other hand, uplink traffic means that the DUT can send frames to the AP whenever frames are needed to be transmitted. Further,

it does not make sense if DUT is always in the awake state, since the power management mechanism completely depends on the sleep state. So the testing scenario for the traffic mode is that DUT download/upload 1 Mbyte data using UDP traffic, followed by sleeping for 30 seconds. And then the process repeats. Actually this scenario is implemented by a tester-defined test case of *Active\_Idle\_Mixture*.

### 3.2.2 PCAF

The PCAFs have to be specified before defining test instructions, then each PCAF has a strict definition to avoid ambiguity.

- Beacon size means the size of a beacon frame in terms of byte. And it can be adjusted by  $AP_{DUT}$
- Signal strength means the strength of the WLAN signal at the receiver port of a STA, in terms of dBm. And it can be adjusted by the attenuators attached.
- DTIM, as discussed in section 2.3.1.
- Beacon period, as discussed in section 2.3.1.
- Foreground traffic means the traffic, where DUT actively participates. Since  $AP_{DUT}$  is configurable, the channel can be adjusted. Moreover, the test case of *Active\_Idle\_Mixture* can be used to adjust both the data rate and the direction of a foreground traffic.
- Background traffic acts as interferences from other existing WLANs. The background traffic can be divided into three types. The first one, denoted by  $BG_{\alpha}$ , means both DUT and  $Phone_{REF}$  are associated to  $AP_{DUT}$ . So both of them works in the same channel. The second one, denoted by  $BG_{\beta}$ , means DUT and  $Phone_{REF}$  join in different BSS but works in the same channel. Finally, the last one, denoted by  $BG_{\gamma}$ , is the same as the second one except that they works in different channels, which means the channel interval, denoted by  $I_{channel}$ , is not equal to zero. Furthermore, three subtypes of  $BG_{\gamma}$  are adopted during experiments,  $BG_{\gamma 1}$ ,  $BG_{\gamma 2}$  and  $BG_{\gamma 3}$ , which correspond to  $I_{channel}=1$ ,  $I_{channel}=2$ , and  $I_{channel}=3$ , respectively. The channel, data rate, and the direction of a background traffic can be adjusted by the test case of *Long\_Duration\_Traffic*.

### 3.2.3 Testing instruction for the idle mode

The most common values of these PCAFs can be treated as the default configuration of the test bed, shown in Tab.3.2. With regard to the idle mode testing, Tab. 3.3 to Tab. 3.9

Table 3.2: Default configuration of the test bed.

<i>Parameter</i>	<i>Default Value</i>
Beacon period	100 <i>ms</i>
DTIM	2
Beacon size	192 <i>bytes</i>
channel	2.437 <i>GHz</i>
Signal strength	55 <i>dBm</i>
BSSID of $AP_{DUT}$	<i>LiSSID</i>
BSSID of $AP_{REF}$	<i>LiSSID_REF</i>
Data rate	2335 <i>kbps</i>

Table 3.3: Description of *TC\_1\_01\_Idle\_BeaconPeriod*

TC_Name	<i>TC_1_01_Idle_BeaconPeriod</i>
Purpose	Test beacon period over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up connection between DUT and <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change beacon period to 50ms, 150ms, 200ms, 300ms, 400ms, and 500ms. Then repeat (1)-(4).</li> </ol>

describe the test instructions of beacon period, DTIM, beacon size, signal strength,  $BG_{\alpha}$ ,  $BG_{\beta}$ , and  $BG_{\gamma}$  for the idle mode, respectively.



Table 3.4: Description of *TC\_1\_02\_Idle\_DTIM*

TC_Name	<i>TC_1_02_Idle_DTIM</i>
Purpose	Test DTIM over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up connection between DUT and <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change DTIM to 1, 3, 4, 5, 6 and 7. Then repeat (1)-(4).</li> </ol>

Table 3.5: Description of *TC\_1\_03\_Idle\_BeaconSize*

TC_Name	<i>TC_1_03_Idle_BeaconSize</i>
Purpose	Test beacon size over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up connection between DUT and the <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change beacon size to 179byte, 214byte, 222byte, and 256byte. Then repeat (1)-(4).</li> </ol>

Table 3.6: Description of *TC\_1\_04\_Idle\_SignalStrength*

TC_Name	<i>TC_1_04_Idle_SignalStrength</i>
Purpose	Test signal strength over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up connection between DUT and <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change signal strength to -65dbm, -75dbm, and -85dbm. Then repeat (1)-(4).</li> </ol>

Table 3.7: Description of  $TC_{1.05\_Idle\_BG_\alpha}$

TC_Name	$TC_{1.05\_Idle\_BG_\alpha}$
Purpose	Test $BG_\alpha$ over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Execute <i>Long_Duration_Traffic</i> to generate a uplink background traffic with specific data rate between <math>Phone_{REF}</math> and <math>AP_{DUT}</math>.</li> <li>(2) Set up connection between DUT and <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(3) After 10 mins, read the value displayed on multimeter.</li> <li>(4) Repeat (2)-(3) for 10 times.</li> <li>(5) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(6) Change the data rate of <math>BG_\alpha</math> to 583kbps, 1167kbps, 1754kbps, 2932kbps, 3523kbps, 4099kbps and 4671kbps. Then repeat (1)-(5).</li> <li>(7) Change the direction of background traffic to downlink, and repeat (1)-(6).</li> </ol>

Table 3.8: Description of  $TC_{1.06\_Idle\_BG_\beta}$

TC_Name	$TC_{1.06\_Idle\_BG_\beta}$
Purpose	Test $BG_\beta$ over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up <math>AP_{REF}</math> with another bssid, <math>LiSSID_{REF}</math>, and the same channel as <math>AP_{DUT}</math>.</li> <li>(2) Execute <i>Long_Duration_Traffic</i> to generate a uplink background traffic with specific data rate between <math>Phone_{Ref}</math> and <math>AP_{ref}</math>.</li> <li>(3) Set up connection between DUT and <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(4) After 10 mins, read the value displayed on multimeter.</li> <li>(5) Repeat (3)-(4) for 10 times.</li> <li>(6) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(7) Change the data rate of <math>BG_\beta</math> to 583kbps, 1167kbps, 1754kbps, 2932kbps, 3523kbps, 4099kbps and 4671kbps. Then repeat (2)-(6).</li> <li>(8) Change the direction of background traffic to downlink, and repeat (2)-(7).</li> </ol>

Table 3.9: Description of  $TC\_1\_07\_Idle\_BG_\gamma$ 

TC_Name	$TC\_1\_07\_Idle\_BG_\gamma$
Purpose	Test $BG_\gamma$ over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up <math>AP_{REF}</math> with <math>LiSSID_{REF}</math>, and <math>I_{channel}=1</math>.</li> <li>(2) Execute <math>Long\_Duration\_Traffic</math> to generate a uplink background traffic with specific data rate between <math>Phone_{Ref}</math> and <math>AP_{ref}</math>.</li> <li>(3) Set up connection between DUT and <math>AP_{DUT}</math>, until DUT is assigned an IP address by Talisker.</li> <li>(4) After 10 mins, read the value displayed on multimeter.</li> <li>(5) Repeat (3)-(4) for 10 times.</li> <li>(6) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(7) Change the data rate of <math>BG_\gamma</math> to 583kbps, 1167kbps, 1754kbps, 2932kbps, 3523kbps, 4099kbps and 4671kbps. Then repeat (2)-(6).</li> <li>(8) Change the direction of background traffic to downlink, and repeat (2)-(7).</li> <li>(9) Change <math>I_{channel}</math> to 2 and 3. Then repeat (2)-(8).</li> </ol>

### 3.2.4 Testing instruction for the traffic mode

With regard to the traffic mode testing, Tab. 3.10 to Tab. 3.15 describe the test instructions for beacon period, DTIM, beacon size, signal strength,  $BG_\alpha$ ,  $BG_\beta$ , and  $BG_\gamma$ , respectively.

Table 3.10: Description of *TC\_2\_01\_Traffic\_BeaconPeriod*

TC_Name	<i>TC_2_01_Traffic_BeaconPeriod</i>
Purpose	Test beacon period over power consumption in traffic pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Execute <i>Active_Idle_Mixture</i> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change beacon period to 50ms, 150ms, 200ms, 300ms, 400ms, and 500ms. Then repeat (1)-(4).</li> <li>(6) Change the data rate of the foreground traffic to 3523kbps. Then repeat (1)-(5).</li> <li>(7) Change the direction of the foreground traffic to downlink, and repeat (1)-(6).</li> </ol>

Table 3.11: Description of *TC\_2\_02\_Traffic\_DTIM*

TC_Name	<i>TC_2_02_Traffic_DTIM</i>
Purpose	Test DTIM over power consumption in traffic pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Execute <i>Active_Idle_Mixture</i> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change DTIM to 1, 3, 4 and 5. Then repeat (1)-(4).</li> <li>(6) Change the data rate of the foreground traffic, to 3523kbps and repeat (1)-(5).</li> <li>(7) Change the direction of the foreground traffic to downlink, and repeat (1)-(6).</li> </ol>

Table 3.12: Description of *TC\_2\_03\_Traffic\_SignalStrength*

TC_Name	<i>TC_2_03_Traffic_SignalStrength</i>
Purpose	Test SignalStrength over power consumption in traffic pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Execute <i>Active_Idle_Mixture</i> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change SignalStrength to -65dBm, -75dBm and -85dBm. Then repeat (1)-(4).</li> <li>(6) Change the data rate of the foreground traffic to 3523kbps. Then repeat (1)-(5).</li> <li>(7) Change the direction of the foreground traffic to downlink, and repeat (1)-(6).</li> </ol>

Table 3.13: Description of *TC\_2\_04\_Traffic\_BG $\alpha$*

TC_Name	<i>TC_2_04_Traffic_BG<math>\alpha</math></i>
Purpose	Test $BG_\alpha$ over power consumption in traffic pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Execute <i>Long_Duration_Traffic</i> to generate an uplink background traffic with specific data rate between <math>Phone_{REF}</math> and <math>AP_{DUT}</math>.</li> <li>(2) Execute <i>Active_Idle_Mixture</i> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(3) After 10 mins, read the value displayed on multimeter.</li> <li>(4) Repeat (2)-(3) for 10 times.</li> <li>(5) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(6) Change the data rate of the foreground traffic to 3523kbps. Then repeat (2)-(5).</li> <li>(7) Change the direction of the foreground traffic to downlink, and repeat (2)-(6).</li> <li>(8) Change the data rate of <math>BG_\alpha</math> to 583kbps, 1167kbps, 1754kbps, 2932kbps, 3523kbps, 4099kbps. Then repeat (1)-(7).</li> <li>(9) Change the direction of background traffic to downlink, and repeat (1)-(8).</li> </ol>

Table 3.14: Description of  $TC\_2\_05\_Traffic\_BG_\beta$ 

TC_Name	$TC\_2\_05\_Traffic\_BG_\beta$
Purpose	Test $BG_\beta$ over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up <math>AP_{REF}</math> with <math>LiSSID_{REF}</math>, and the same channel as <math>AP_{DUT}</math>.</li> <li>(2) Execute <i>Long_Duration_Traffic</i> to generate an uplink background traffic with specific data rate between <math>Phone_{REF}</math> and <math>AP_{ref}</math>.</li> <li>(3) Execute <i>Active_Idle_Mixture</i> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(4) After 10 mins, read the value displayed on multimeter.</li> <li>(5) Repeat (3)-(4) for 10 times.</li> <li>(6) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(7) Change the data rate of the foreground traffic to 3523kbps. Then repeat (3)-(6).</li> <li>(8) Change the direction of the foreground traffic to downlink, and repeat (3)-(7).</li> <li>(9) Change the data rate of <math>BG_\beta</math> to 583kbps, 1167kbps, 1754kbps, 2932kbps, 3523kbps and 4099kbps. Then repeat (2)-(8).</li> <li>(10) Change the direction of background traffic to downlink, and repeat (2)-(9).</li> </ol>

Table 3.15: Description of  $TI.2.06.Traffic.BG_{\gamma}$

TC_Name	$TI.2.06.Traffic.BG_{\gamma}$
Purpose	Test $BG_{\gamma}$ over power consumption in idle pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Set up <math>AP_{REF}</math> with <math>LiSSID_{ref}</math>, and <math>I_{channel} = 1</math>.</li> <li>(2) Execute <math>Long\_Duration\_Traffic</math> to generate an uplink background traffic with specific data rate between <math>Phone_{REF}</math> and <math>AP_{REF}</math>.</li> <li>(3) Execute <math>Active\_Idle\_Mixture</math> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(4) After 10 mins, read the value displayed on multimeter.</li> <li>(5) Repeat (3)-(4) for 10 times.</li> <li>(6) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(7) Change the data rate of the foreground traffic to 3523kbps. Then repeat (3)-(6).</li> <li>(8) Change the direction of the foreground traffic to downlink, and repeat (3)-(7).</li> <li>(9) Change the data rate of <math>BG_{\beta}</math> to 583kbps, 1167kbps, 1754kbps, 2932kbps, 3523kbps and 4099kbps. Then repeat (2)-(8).</li> <li>(10) Change the value of <math>I_{channel}</math>, and repeat (2)-(9).</li> </ol>

Table 3.16: Description of  $TC.2.07.Traffic.ForegroundTraffic$

TC_Name	$TC.2.07.Traffic.ForegroundTraffic$
Purpose	Test foreground traffic over power consumption in traffic pattern
Precondition	Default configuration of the test bed
Test Procedure	<ol style="list-style-type: none"> <li>(1) Execute <math>Active\_Idle\_Mixture</math> to generate an uplink foreground traffic with a specific data rate between DUT and <math>AP_{DUT}</math>.</li> <li>(2) After 10 mins, read the value displayed on multimeter.</li> <li>(3) Repeat (1)-(2) for 10 times.</li> <li>(4) Remove the maximal and the minimal from the sample space, followed by an average calculation.</li> <li>(5) Change data rate 1167kbps, 1754kbps, 2932kbps, 3523kbps, 4099kbps and 4671kbps. Then repeat (1)-(4).</li> <li>(6) Change the foreground traffic to uplink. Then repeat (1)-(5).</li> </ol>

### 3.3 Testing results

#### 3.3.1 Beacon period

Beacon period is a predefined PCAF in an AP, which defines the period between two consecutive beacon frames. The value can be set up by an AP. Normally a higher beacon period value indicates a lower wake up frequency of a STA, which means the STA consumes less power.

Fig. 3.3, based on the test instruction of *TC\_1\_01\_Idle\_BeaconPeriod*, displays the relationship between beacon period and the power consumption. During the measurement, the beacon period is set between 50 ms to 500 ms, with the interval of 50 ms or 100 ms. When the beacon period increases from 50 ms to 200 ms, the power consumption collapses, 8.9 mW dropped, which accounts for around 40% of the power with the beacon period of 50 ms. And then the decrease trend becomes less observable when the beacon

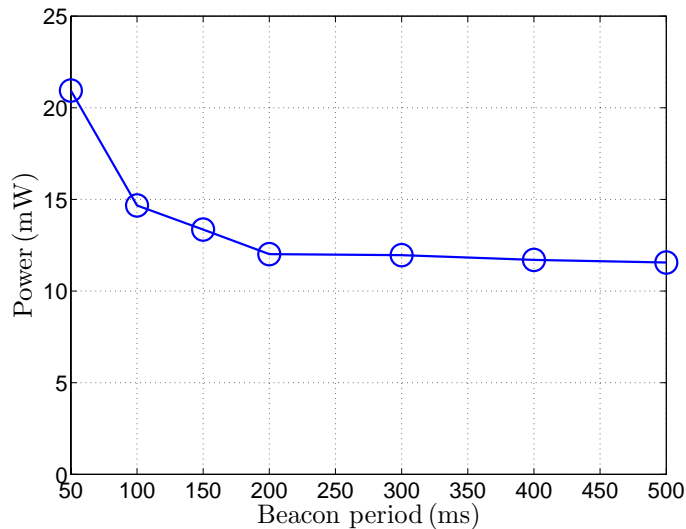


Figure 3.3: Power as a function of the beacon period in the idle mode.

period is larger than 200ms. As shown in Fig. 3.3, only 0.45 mW is dropped when the beacon period rises from 200 ms to 500 ms.

Fig.3.4, based on *TC\_2\_01\_Traffic\_BeaconPeriod*, represents the relations among power consumption, beacon period and foreground traffic (uplink/downlink). The data rate of uplink traffic and downlink traffic are denoted by  $R_U$  and  $R_D$  respectively, and both of them have two options, 2335 kbps and 3523 kbps. As shown in Fig.3.4, the power consumed with uplink traffic is more than that with downlink traffic, and for the same traffic direction, power consumption is higher with a lower data rate than that with a higher data rate. All of the four curves have the same trend as the curve in Fig. 3.3, namely a sharp



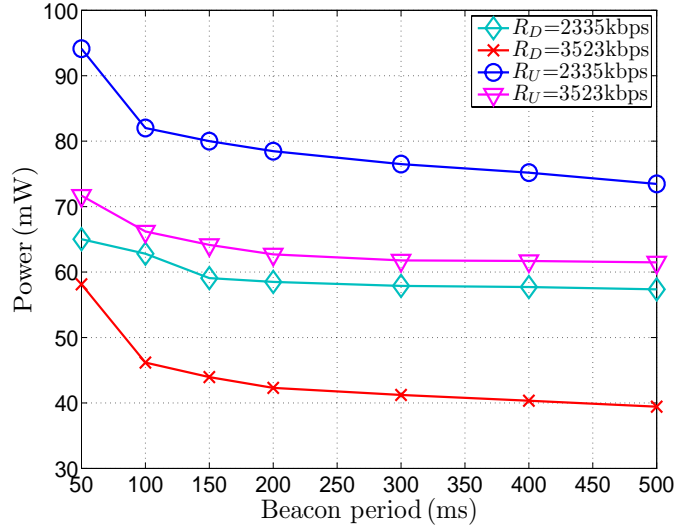


Figure 3.4: Power as a function of the beacon period in the traffic mode.

decrease at first, followed by a slow decrease. Another interesting observation is that the gap between the curve ( $R_U = 2335$  kbps) and the curve ( $R_D = 3523$  kbps) stays almost the same, around 36.10mW, while the same phenomenon also lies in the gap between the curve ( $R_U = 3523$  kbps) and the curve ( $R_D = 2335$  kbps) except the value is roughly 4.08 mW.

### 3.3.2 DTIM

DTIM is also a predefined PCAF in an AP, which specifies the number of beacon intervals between two consecutive DTIMs. It functions to notify STAs of buffered frames. Generally, the wake period of a STA in idle mode is

$$T_{wakepd} = T_{beaconpd} \cdot DTIM, \quad (3.3)$$

where  $T_{wakepd}$  and  $T_{beaconpd}$  denote the wake period and the beacon period, respectively, so DTIM and beacon period have a tight correlation to each other, which can be indicated by comparing Fig. 3.3 with Fig. 3.5.

The curve in Fig. 3.5, based to *TC\_1\_02\_Idle\_DTIM*, displays a little sharper decrease compared with Fig. 3.3. A net drop of 10.94 mW is displayed when DTIM changes from 1 to 3, which accounts for 45.83% of the power with DTIM of 1, while the net drop corresponding to the increase of DTIM from 3 to 7 is much less, 2.61 mW shown up.

The curves in Fig. 3.6 is based on *TC\_2\_02\_Traffic\_DTIM* and display almost the same trend as Fig. 3.4, except that the decrease trend is a little more substantial. The gap be-

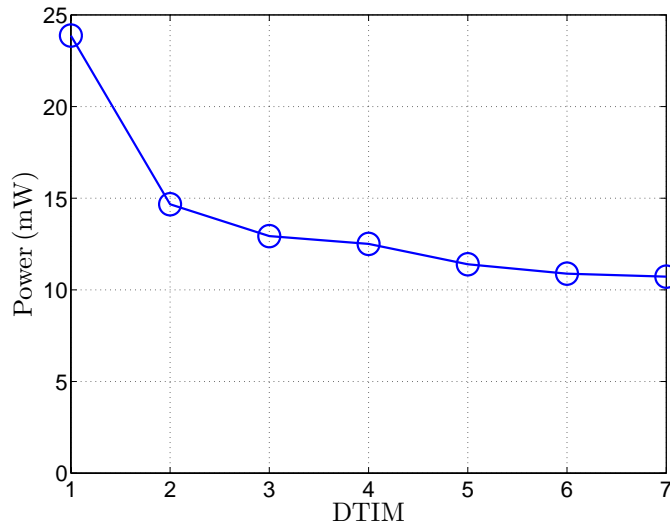


Figure 3.5: Power as a function of the DTIM in the idle mode.

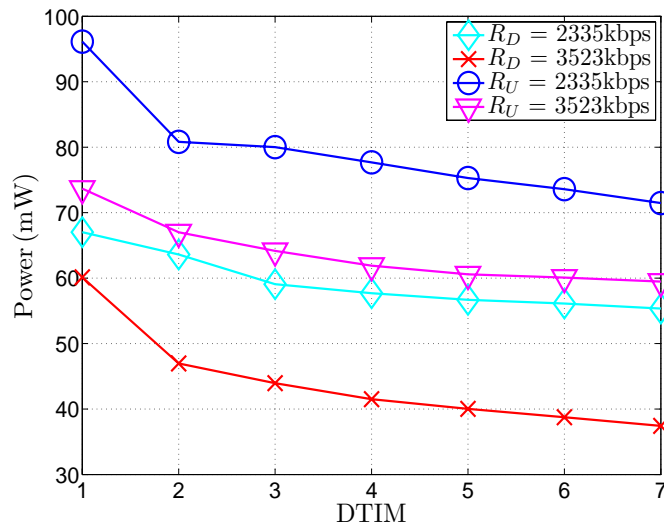


Figure 3.6: Power as a function of the DTIM in the traffic mode.

tween the curve ( $R_U = 2335$  kbps) and the curve ( $R_D = 3523$  kbps) stays around 36.89 mW, while the gap between the curve ( $R_U = 3523$  kbps) and the curve ( $R_D = 2335$  kbps) is roughly 4.78 mW.

### 3.3.3 Beacon size

A beacon frame is used to broadcast services a BSS provides and the protocol regulations a STA has to comply with if it wants to join the BSS. The more details included in a beacon frame, the bigger the beacon frame is, which indicates that it costs more time

for a STA to parse beacon frames. In Tab. 3.17, several typical beacon size samples are specified with the corresponding services

Fig. 3.7 is based on the test instruction of *TC\_1\_03\_Idle\_BeaconSize*. The beacon size

Table 3.17: Beacon size samples

<i>Beacon Size (byte)</i>	<i>BSS service</i>
179	802.11b open authentication
192	802.11g open authentication
214	802.11g WPA
222	802.11g WPA + Dot11
256	802.11g WMM

does not have much effect on power conservation compared to other parameters. The sampling range is in a discrete manner with different intervals, due to the choice of BSS service. Along the curve, there is not much fluctuation. The power consumption goes up with only 2.04mW, occupying 14.83% of the average power consumption, when the beacon size varies from 179 bytes to 256 bytes.

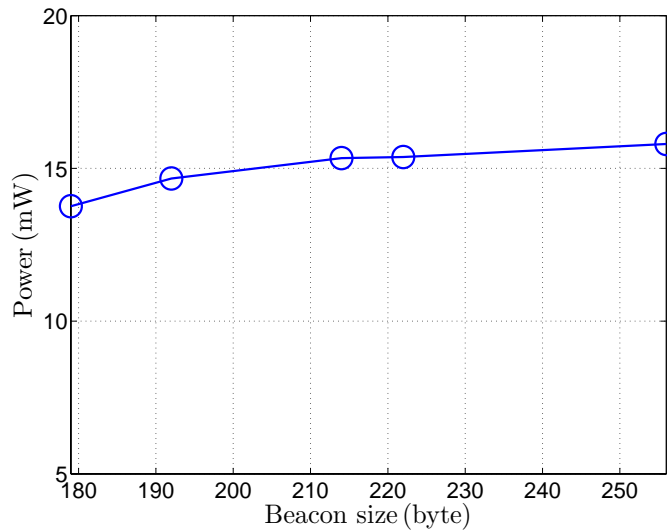


Figure 3.7: Power as a function of the beacon size in the idle mode.

The investigation of beacon size in traffic mode are not available, since it depends on implementation of test cases for specific services using Python [23]. So it is out of the scope of the thesis.

### 3.3.4 Signal strength

A signal strength represents the quality of a WLAN connection between an AP and a STA. A lower RSS value indicates a worse status of a connection. Fig. 3.8 is based on the test instruction of *TC\_1\_04\_Idle\_SignalStrength*. If the physical link data rate,  $R_{PHY}$ , is non-fixed, the curve seems incorrect, since the worse connection status is supposed to introduce more power consumption. After exploring, the reason for the strange behavior is that the  $R_{PHY}$  of DUT can be automatically adjusted based on the signal strength, which means a higher strength makes  $R_{PHY}$  shift to a higher level in order to achieve a higher throughput. On the other hand, the higher data rate introduces more complex encoding/decoding scheme, so more power are supposed to be consumed. That is why DUT with -55 dBm signal strength consumes almost the same power as that with -85 dBm.

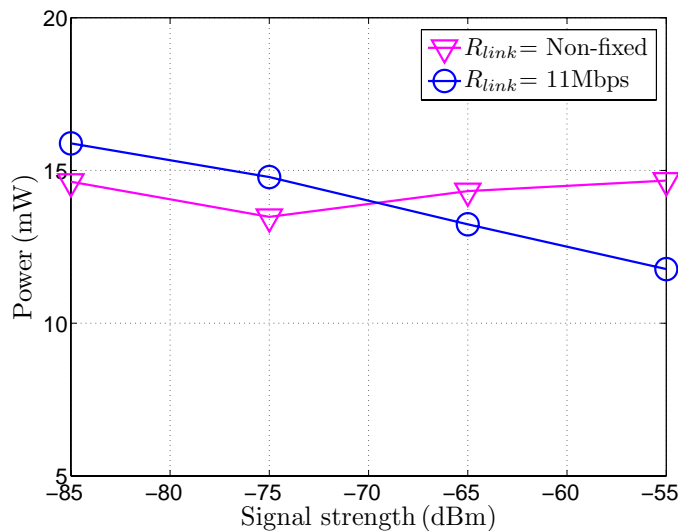


Figure 3.8: Power as a function of the signal strength in the idle mode.

Furthermore, another curve in Fig.3.8, corresponding to  $R_{PHY} = 11$  Mbps, represents an almost linear relationship between the signal strength and the power consumption. A net increase of power consumption of 4.12 mW is introduced when the strength shifts from -55 dBm to -85 dBm, accounting for 34.96% of the average power consumption. Since all the connection are through shielded RF cables, much interferences are avoided. So DUT with the same signal strength in a real environment is supposed to consume more power.

Fig. 3.9 depends on the test instruction of *TC\_2\_03\_Traffic\_SignalStrength*. All the curves in displays the same trend. The power consumption slightly decreases with a net drop of 2.21 mW in average, when the RSS increases from -85 dBm to -65 dBm, and then a rapid decline, 12.16 mW in average, is shown up when the RSS rises with one more 10dBm.

Another observation is that the uplink foreground traffic indicates more power consump-

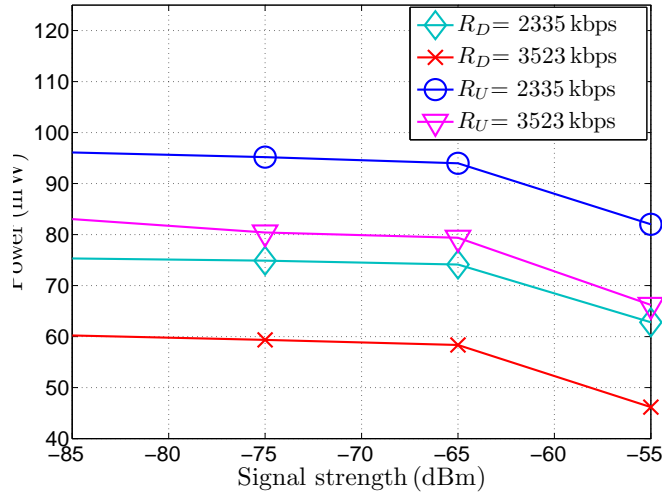


Figure 3.9: Power as a function of the signal strength in the traffic mode.

tion than that with downlink traffic. The gaps between any two consecutive curves remain the same, roughly 16.19 mW, 5.52 mW, 15.52 mW, respectively.

### 3.3.5 Foreground traffic

The foreground traffic aims to implement testing scenarios for the traffic mode, which is supposed to consume more power than the idle mode. Fig.3.10, based on test instruction

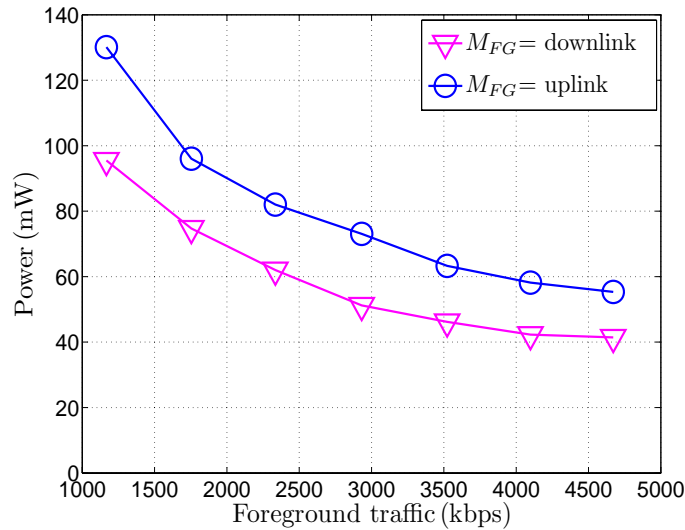


Figure 3.10: Power as a function of the foreground traffic.

of *TC\_2\_07\_Traffic\_ForegroundTraffic*, shows the relationship between the power con-

sumption and the foreground traffic. Both of the curves exhibit a downward trend with the increase of the data rate, and the uplink mode consumes more power than the downlink mode. The gap declines slightly from 34.62mW to 13.89mW, when the data rate rises from 1167kbps to 4671kbps. The data rate of the foreground traffic is inversely proportional to the power consumption, since a higher data rate indicates less time to complete a specific transmission, which means DUT spends less time in the active state.

### 3.3.6 Background traffic

This section covers all the experiments related to all types of background traffics,  $BG_\alpha$ ,  $BG_\beta$ , and  $BG_\gamma$ , in both the idle mode and the traffic mode. There are two options for the direction of a background traffic either uplink or downlink, which is denoted by  $M_{BG}$ . Moreover, a data rate is another character of a background traffic. Since the background traffic is implemented with the test case of *Long\_Duration\_Traffic*, where a series of data rates are predefined, as shown in Tab.3.18.

Table 3.18: Predefined data rate of background traffic

<i>Sequence</i>	<i>Data_Rate(kbps)</i>
1	583
2	1167
3	1754
4	2335
5	2932
6	3523
7	4099
8	4671

#### A. Idle mode

Fig.3.11, based on the test instruction of *TC\_1\_05\_Idle\_BG $\alpha$* , represents the relationship between the power consumption and  $BG_\alpha$  in idle mode, where both uplink and downlink are taken into account. The downlink  $BG_\alpha$  has much less effect compared with uplink  $BG_\alpha$ , which means power consumption is much more sensitive to the uplink  $BG_\alpha$  than to the downlink  $BG_\alpha$ . Regarding the downlink  $BG_\alpha$ , a net increase of 2.80 mW, occupying 20.11%, is observed, when the data rate goes up from 583 kbps to 4671 kbps. On the other hand, the uplink  $BG_\alpha$  causes 162.08% increase in the same condition. So the gap between two curves gets larger and larger, along with the increase of the data rate of the background traffic.

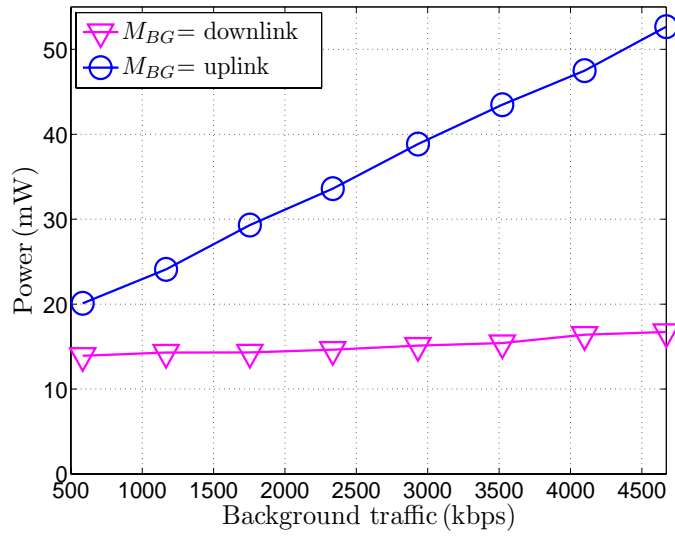


Figure 3.11: Power as a function of the  $BG_\alpha$  in the idle mode.

The relationship between the power consumption and  $BG_\beta$  is shown in Fig.3.12, based on the test instruction of  $TC\_1\_06\_Idle\_BG_\beta$ . Both the uplink  $BG_\beta$  and the downlink  $BG_\beta$  display upward increase trend against the increase of the data rate. And the uplink  $BG_\beta$  causes a little more effect on power consumption than the downlink  $BG_\beta$ , and the deviation shrinks for a higher data rate. At the lowest data rate, the gap is 4.08mW, accounting for 27.15% of the power with the uplink mode, while the gap decrease to 0.10mW, and the percentage declines to 0.25%.

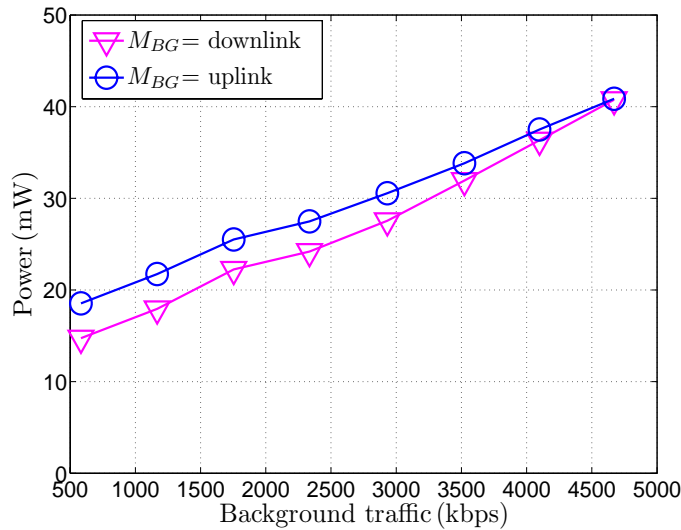


Figure 3.12: Power as a function of the  $BG_\beta$  in the idle mode.

Fig. 3.13 to Fig. 3.15 are based on the test instruction of  $TC\_1\_07\_Idle\_BG_\gamma$ . The effect

against power consumption caused by the  $BG_{\gamma_1}$  is shown in Fig. 3.13. Both of the curves exhibit upward trend against the increase of the data rate. And the DUT consumes more power with the uplink  $BG_{\gamma_1}$  than that with the downlink  $BG_{\gamma_1}$ . And the gap between these curves ascends rapidly with data rate's increase, which varies from 4.71 mW to 19.24 mW, almost 4 times rise.

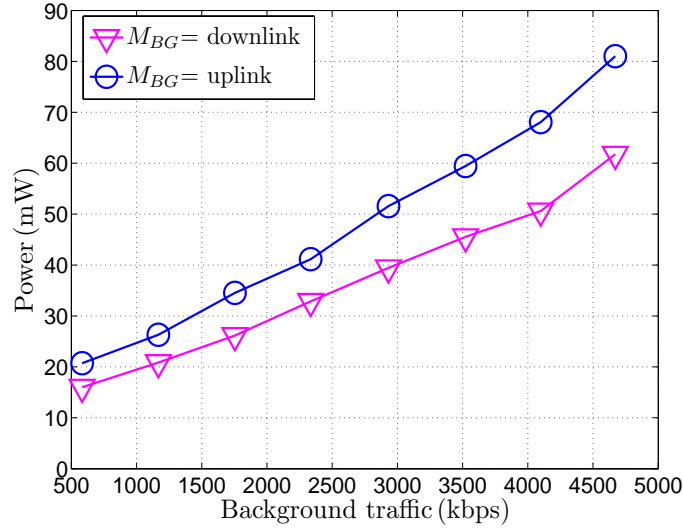


Figure 3.13: Power as a function of the  $BG_{\gamma_1}$  in the idle mode.

How the  $BG_{\gamma_2}$  affecting the power consumption can be indicated with the help of Fig.3.14. The the uplink  $BG_{\gamma_2}$  wastes more energy than the downlink  $BG_{\gamma_2}$ . When the data rate varies from 583 kbps to 2335 kbps, both the uplink  $BG_{\gamma_2}$  and the downlink  $BG_{\gamma_2}$  roughly have the same effect over power consumption, only the deviation of 1.32mW remaining. However, the gap between these curves slightly increases to 4.64mW when the data rate reaches 4671kbps.

As shown in Fig. 3.15, the  $BG_{\gamma_3}$  exhibits almost the same behavior as the  $BG_{\gamma_1}$ , namely a smaller gap at a lower data rate followed by a larger gap at a higher data rate, except a couple of points. The first one is that the gap corresponding to 583kbps is almost equal to zero, which means both the uplink  $BG_{\gamma_3}$  and the downlink  $BG_{\gamma_3}$  have the same effect on power consumption at the 583kbps point. And the second one is that the curves in Fig.3.13 are steeper than that in Fig.3.15, which means the  $BG_{\gamma_1}$  causes more power consumption compared with the  $BG_{\gamma_3}$ .

Considering all the figures related to background traffics in idle mode, Fig. 3.16 and Fig. 3.17 are created. Fig. 3.16 compares all the background traffic factors,  $BG_{\alpha}$ ,  $BG_{\beta}$ ,  $BG_{\gamma_1}$ ,  $BG_{\gamma_2}$ , and  $BG_{\gamma_3}$  in terms of downlink background traffic, while Fig.3.17 compares these factors regarding uplink background traffic. Based on them, a couple of



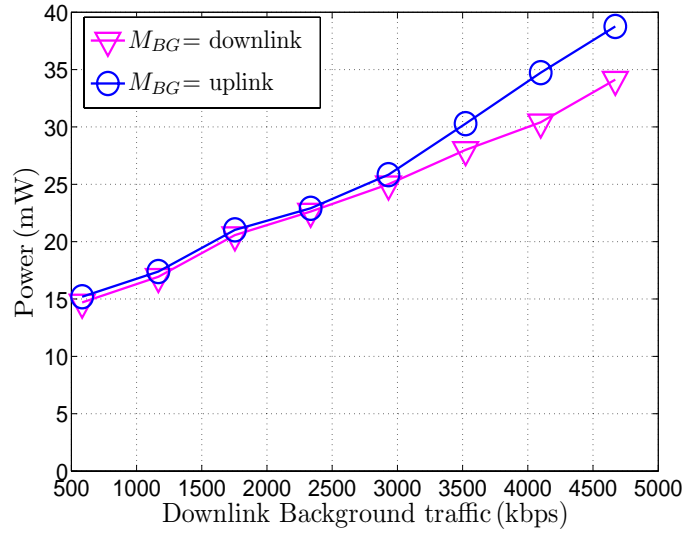


Figure 3.14: Power as a function of the  $BG_{\gamma_2}$  in the idle mode.

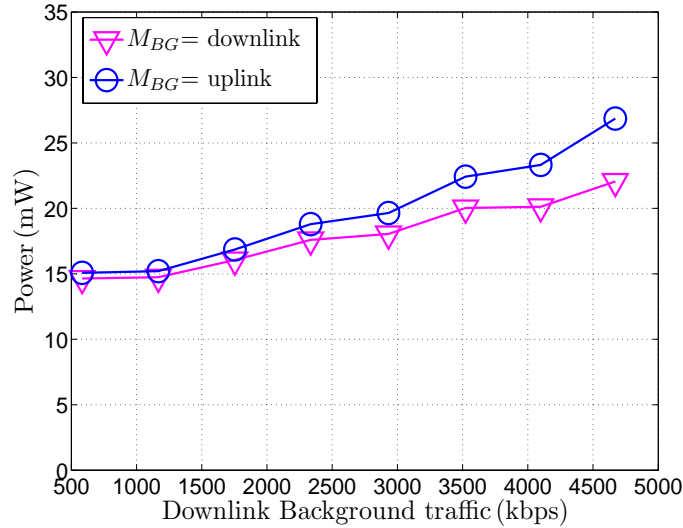


Figure 3.15: Power as a function of the  $BG_{\gamma_3}$  in the idle mode.

observations can be obtained.

- The effect over power consumption caused background traffic parameters are sorted in a decreasing order are  $BG_{\gamma_1}$ ,  $BG_{\beta}$ ,  $BG_{\gamma_2}$ , and  $BG_{\gamma_3}$ , which is correct for both uplink background traffic and downlink background traffic. And  $BG_{\alpha}$  is not valid with this observation, since the uplink  $BG_{\alpha}$  has much more effect against power consumption than downlink  $BG_{\alpha}$ . Uplink  $BG_{\alpha}$  ranks in the second position among all the uplink factors, while downlink  $BG_{\alpha}$  is located in the last position of all the downlink factors.
- With regard to the lowest data rate of background traffic, 583kbps, downlink fac-

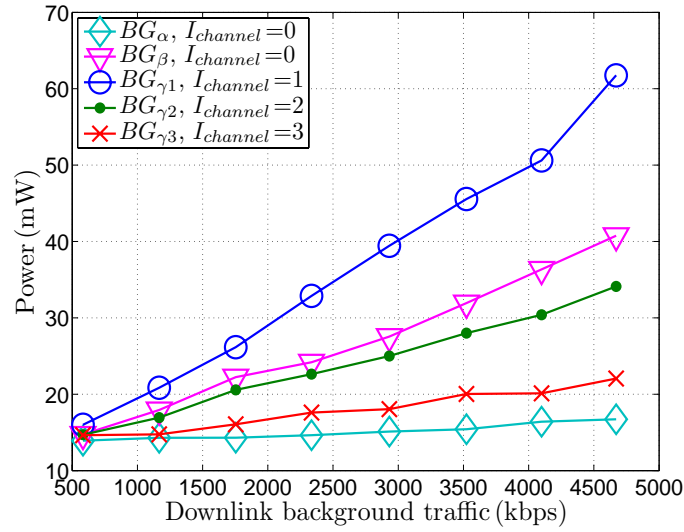


Figure 3.16: Power as a function of the downlink background traffic in the idle mode.

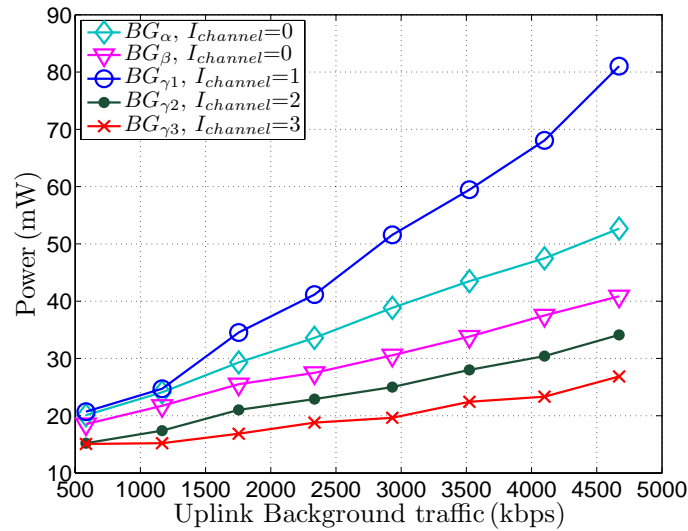


Figure 3.17: Power as a function of the uplink background traffic in the idle mode.

tors do not exhibit much difference over power consumption at this point, since the maximal deviation is just 2.08mW. However, the maximal deviation of uplink factors at this point is 5.63mW, almost two and a half times than the previous one.

- For all the background traffic factors, uplink factors always lead to more power consumption than downlink factors.

### B. Traffic mode

The relationship between  $BG_\alpha$  and power consumption in traffic pattern is displayed in Fig. 3.18 based on the test instruction of *TC\_2\_04\_Traffic\_BG $\alpha$* . All of the curves exhibits an upward trend against the increase of the data rate. And the gaps among these curves roughly remain the same, when the data rate varies from 583 kbps to 3523 kbps, then some of them sharply increase due to the change of data rate from 3523 kbps to 4099 kbps. Firstly, with regard to the foreground traffic, provided that the data rate is the

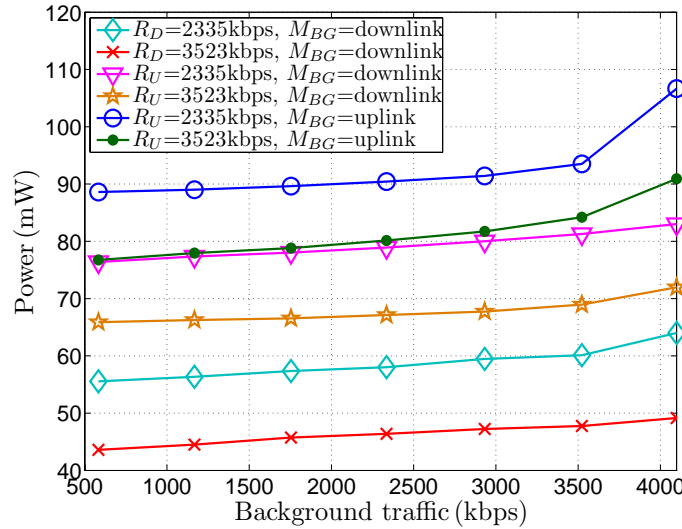


Figure 3.18: Power as a function of the  $BG_\alpha$  in the traffic mode.

same, the uplink causes more power consumption than the downlink. The reason is that the downlink traffic is required to be coordinated by the AP, while the uplink traffic is of an abrupt type initiated by a STA without any coordination involved. If the direction of the foreground traffic has been fixed, a higher data rate indicates less power consumption than a lower one. Secondly, regarding background traffic, if the data rate is fixed, the uplink causes more power consumption than the downlink. Finally, if both the background traffic and the foreground traffic are uplink, more power is wasted than others, which can be indicated by the first top two curves.

Fig. 3.19, which is based on the test instruction of *TC\_2\_05\_Traffic\_BG $\beta$* , displays the relationship between  $BG_\beta$  and power consumption in the traffic pattern. All the curves gradually increase against the increase of the data rate. The ascending slopes of all the curves with uplink foreground traffic are almost the same, which is higher than that with downlink foreground traffic. So based on this point, the uplink foreground traffic is the dominant factor for  $BG_\beta$ . Furthermore, a higher data rate of the foreground traffic indicates less power consumption, which is the same as the Fig. 3.18.

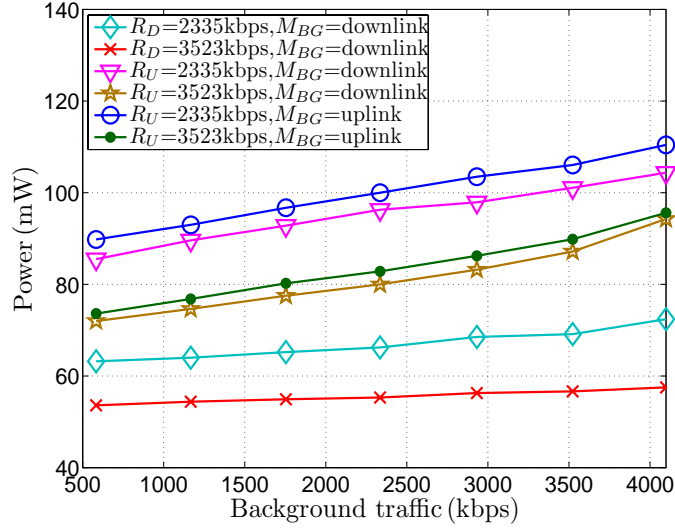


Figure 3.19: Power as a function of the  $BG_\beta$  in the traffic mode.

As shown from Fig. 3.20 to Fig. 3.23, the background traffic parameters, namely  $BG_\alpha$ ,  $BG_\beta$ ,  $BG_{\gamma_1}$ ,  $BG_{\gamma_2}$ , and  $BG_{\gamma_3}$ , are compared, when a foreground traffic is involved. Fig.3.20 shows the comparison, provided that the foreground traffic is downlink with the data rate of 2335kbps. All the curves exhibit upward trend when the data rate increases, and the ascending slopes are almost the same. Statistically,  $BG_{\gamma_1}$  leads to 8.35mW, 10.30mW, 18.32mW, 20.98mW more than  $BG_{\gamma_2}$ ,  $BG_\beta$ ,  $BG_\alpha$ , and  $BG_{\gamma_3}$ , respectively. Moreover,  $BG_{\gamma_2}$  and  $BG_\beta$  have almost the same effect on power consumption, only the deviation of 1.95mW remaining. And the same phenomenon occurs between  $BG_\alpha$ , and  $BG_{\gamma_3}$ , except that the deviation is 2.66mW.

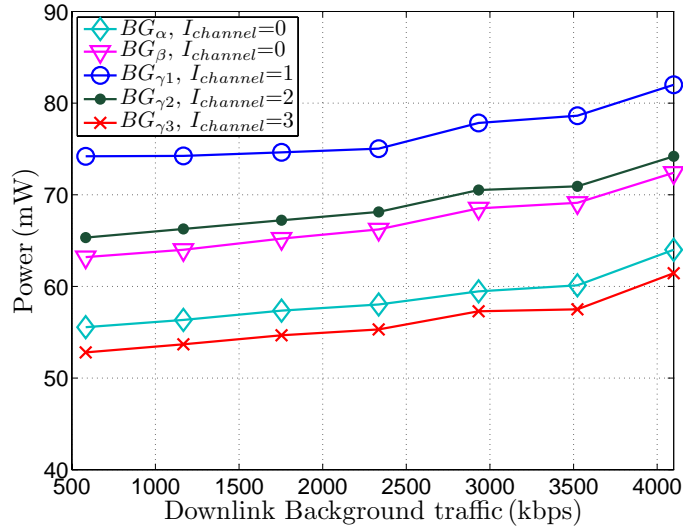


Figure 3.20: Power comparison ( $R_U = 2335$  kbps and  $M_{BG} = \text{downlink}$ ).

Considering the downlink foreground traffic with the data rate of 3523kbps, the comparison is shown in Fig.3.21. The curves displays almost the same ascending slopes, which means that there are roughly linear correlations among these curves. Statistically,  $BG_{\gamma_1}$  leads to 8.58mW, 13.15mW, 19.69mW, 22.326mW more than  $BG_{\gamma_2}$ ,  $BG_{\beta}$ ,  $BG_{\gamma_3}$  and  $BG_{\alpha}$ , respectively. Compared with Fig.3.20, a higher data rate of the foreground traffic indicates less power consumption, roughly 9.51mW drop in average. And one more observation is that a higher data rate of foreground traffic makes  $BG_{\alpha}$  consumes less power that a lower data rate.

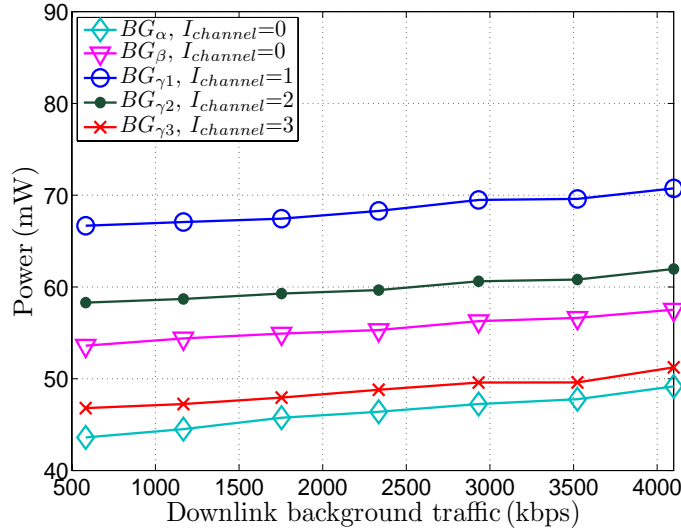


Figure 3.21: Power comparison ( $R_U = 3523$  kbps and  $M_{BG} = \text{downlink}$ ).

Fig.3.22 is based on the precondition that the direction and the data rate of the foreground traffic are set to uplink and 2335kbps, respectively. All of the curves shows upward trends along the increase of the data rate, but the ascending slope of the curve with  $BG_{\alpha}$  is 63.10% less than the average ascending slope of the curves with other background traffic parameters, which means  $BG_{\alpha}$  is more stable in affecting power consumption when the direction of the foreground traffic shifts from downlink to uplink. Statistically,  $BG_{\gamma_1}$  causes 7.27mW, 10.75mW, 17.96mW more than  $BG_{\gamma_2}$ ,  $BG_{\beta}$ , and  $BG_{\gamma_3}$ , respectively. And the gap between  $BG_{\alpha}$  and the average of other parameters increases from 11.26mW to 33.71mW.

Fig.3.23 compares different background traffic parameters with a 3523kbps uplink foreground traffic involved. The same phenomenon indicated by Fig.3.22 appears again, which means the ascending slope of the curve with  $BG_{\alpha}$  is much lower than that with other background traffic parameters. And the gap of power consumption caused by  $BG_{\alpha}$  and the average of other parameters increases from 8.28mW to 23.04mW. Compared

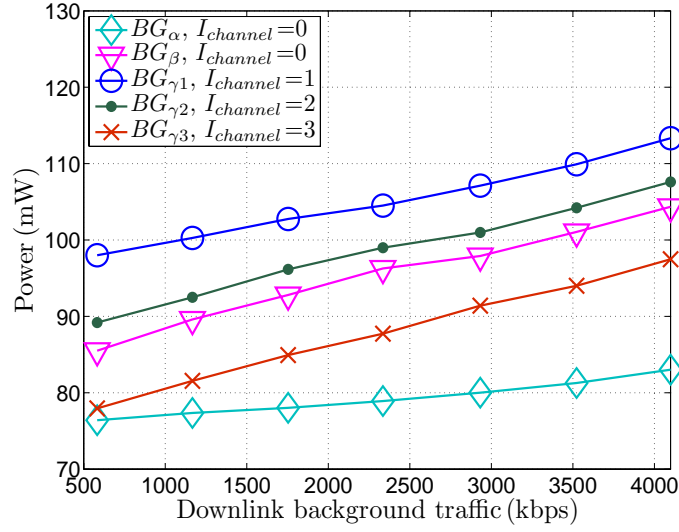


Figure 3.22: Power comparison ( $R_U = 2335$  kbps and  $M_{BG} = uplink$ ).

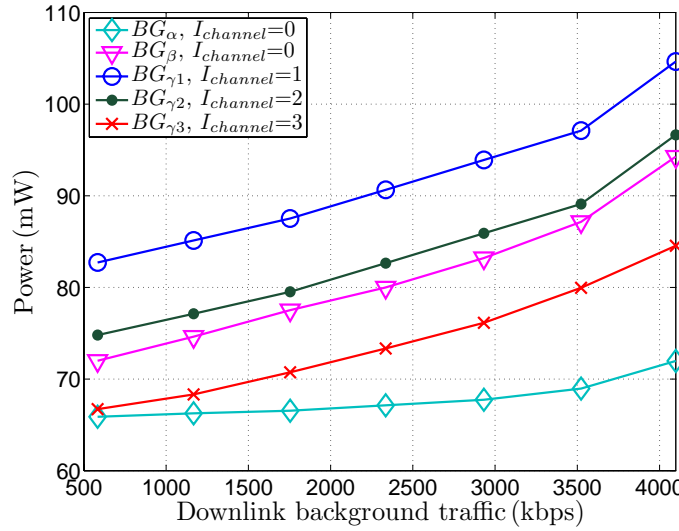


Figure 3.23: Power comparison ( $R_U = 3523$  kbps and  $M_{BG} = uplink$ ).

with Fig.3.22, in average, a net decrease, 12.16mW, is observed when the uplink data rate shifts from 2335kbps to 3523kbps.

Base on the figures from Fig.3.20 to Fig.3.23, several observations related to the traffic pattern are as follows.

- The effect over power consumption caused background traffic parameters are sorted in a decreasing order are  $BG_{\gamma1}$ ,  $BG_{\gamma2}$ ,  $BG_{\beta}$  and  $BG_{\gamma3}$ , which is correct for both uplink foreground traffic and downlink foreground traffic. And  $BG_{\alpha}$  ranks in the last position in all scenarios except that it ranks in last second position when the data rate of downlink foreground traffic is 2335kbps.  $BG_{\alpha}$  roughly can not cause

more power consumption compared with others.

- The ascending slope of the curve with  $BG_{\alpha}$  is almost the same as that with other background traffic parameters, provided that the direction of the foreground traffic is downlink. However, if the direction changes to uplink, the ascending slope greatly declines, which is lower than that with other background traffic parameters. Furthermore, the ascending slopes of  $BG_{\gamma_1}$ ,  $BG_{\gamma_2}$ ,  $BG_{\beta}$  and  $BG_{\gamma_3}$  remains the same, confronted with any direction of the foreground traffic.
- With regarding to the foreground traffic, the uplink mode consumes more power than the downlink mode. And a higher data rate introduces less power consumption than a lower data rate, provided that the foreground traffic mode is fixed, either uplink or downlink.

## Chapter 4

# WLAN Power Consumption Calculation

### 4.1 Power calculation in the idle mode

The calculation of power consumption for an idle mode is corresponding to the idle pattern experiments. A STA wakes up for beacon frames in order to check the existence of buffered packets destined for it. In this mode, no traffic is involved, so the STA transits to a sleep state at the end of each beacon packet. In practice, to avoid beacon frame lost, the STA has to wake up a little earlier before the arrival of any beacon packet.

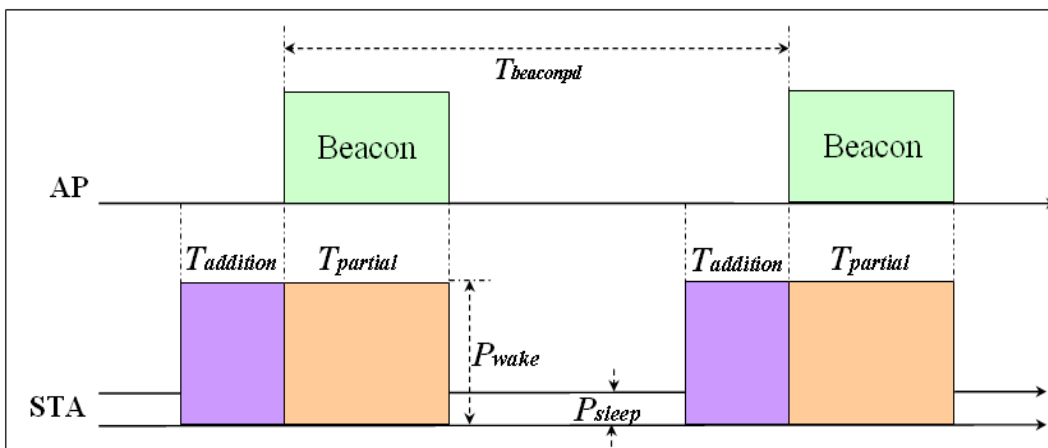


Figure 4.1: Interaction between a STA and an AP in the idle mode.

As shown in Fig.4.1,  $T_{addition}$  and  $T_{partial}$  denote the extra time before the arrival of a beacon packet, and the time for receiving a beacon frame. So the total active time,



denoted by  $T_{wake}$  during each beacon period is

$$T_{wakesingle} = T_{partial} + T_{addition}, \quad (4.1)$$

where  $T_{partial}$  depends on both the encapsulated frame size and the data rate of the physical layer, denoted by  $S_{PHY}$ , and  $R_{PHY}$ , respectively, shown as follows

$$T_{partial} = \frac{S_{PHY}}{R_{PHY}}. \quad (4.2)$$

And, as shown in Eq. 4.3  $S_{PHY}$  is composed with three components, the beacon packet size, the physical synchronization size, and the MAC frame header size, denoted by  $S_{beacon}$ ,  $S_{SYN}$ , and  $S_{Header}$ , respectively.

$$S_{PHY} = S_{beacon} + S_{SYN} + S_{Header}. \quad (4.3)$$

Moreover, without beacon lost probability ( $P_{BL}$ ) taken into account, the average power consumption during a beacon period is

$$\begin{aligned} P_{idle} &= \frac{P_{sleep}T_{sleep} + P_{wake}T_{wakesingle}}{T_{beaconpd}} \\ &= \frac{P_{sleep}(T_{beaconpd} - T_{wakesingle}) + P_{wake}T_{wakesingle}}{T_{beaconpd}}, \end{aligned} \quad (4.4)$$

where  $P_{sleep}$  and  $P_{wake}$  represent the power consumed for the sleep state and the active state. After substitute Eq. 4.1- Eq. 4.3 into Eq. 4.4, the power calculation in an idle mode without  $P_{BL}$  can be obtained

$$P_{idle} = P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconperiod}} \cdot (T_{addition} + \frac{S_{beacon} + S_{SYN} + S_{Header}}{R_{PHY}}). \quad (4.5)$$

After substitute the parameters with the default values in Tab.4.1,  $P_{idle} = 3.29$  mW. Finally, if  $P_{BL}$  is taken into account,

$$P_{idleBL} = P_{idle} \cdot (1 - P_{BL}) + P_{wake} \cdot P_{BL}, \quad (4.6)$$

where  $P_{idleBL}$  denotes the power consumption with  $P_{BL}$  considered. After substitute Eq. 4.5 into Eq. 4.6, the power in an idle mode considering  $P_{BL}$  can be calculated as

$$P_{idleBL} = P_{wake} \cdot P_{BL} + (1 - P_{BL}) \cdot \left[ P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot \left( T_{addition} + \frac{S_{beacon} + S_{SYN} + S_{Header}}{R_{PHY}} \right) \right]. \quad (4.7)$$

Since  $P_{BL}$  is involved in the power calculation for the idle mode, it can be obtained in separate different ways, either a mathematical deduction or an experimental method. With an average perspective, we get

$$T_{duration} \cdot P_{test} = \frac{T_{duration}}{T_{sleep} + T_{wake}} (1 - P_{BL}) (P_{sleep} T_{sleep} + P_{wake} T_{wake}) + \frac{T_{duration}}{T_{sleep} + T_{wake}} P_{BL} P_{wake} (T_{sleep} + T_{wake}), \quad (4.8)$$

where  $T_{duration}$  denotes the time for a testing scenario, while  $P_{test}$  represents the result shown on the multimeter. So the formula to calculate  $P_{BL}$  in idle mode is

$$P_{BL} = \frac{P_{wake} (T_{sleep} + T_{wake}) - (P_{sleep} T_{sleep} + P_{wake} T_{wake})}{(P_{wake} - P_{sleep}) T_{sleep}}. \quad (4.9)$$

On the other hand, a script, attached in Appendix, is created, which can be executed on the multimeter to roughly measure  $P_{BL}$ . Actually, the execution result refers to the ratio of the active time against the duration of a testing scenario,

$$R_{wakeratio} = \frac{T_{wake}}{T_{duration}} = \frac{\Sigma T_{normal} + \Sigma T_{BL}}{\Sigma T_{normal} + \Sigma T_{BL} + \Sigma T_{sleep}}, \quad (4.10)$$

Table 4.1: Default values for the power calculation in the idle mode

<i>Parameter</i>	<i>Default Value</i>
$S_{beacon}$	192 bytes
$S_{SYN}$	24 bytes
$S_{Header}$	34 bytes
$R_{PHY}$	1 Mbps
$T_{addition}$	2 ms
$P_{sleep}$	3.2 mW
$P_{wake}$	432 mW

where  $\Sigma T_{normal}$ , and  $\Sigma T_{BL}$  denote the active time cost for the normal wake up for beacon packets and the beacon frame lost, respectively. Moreover,  $\Sigma T_{normal}$  can also be expressed as

$$\Sigma T_{normal} = T_{wake} \cdot \frac{T_{duration} - \Sigma T_{BL}}{T_{beaconpd}}. \quad (4.11)$$

After substitute Eq. 4.11 into Eq. 4.10, we have

$$R_{wakeratio} = \frac{T_{wake} \cdot (T_{duration} - \Sigma T_{BL})}{T_{beaconpd} \cdot T_{duration}} + \frac{\Sigma T_{BL}}{T_{duration}}. \quad (4.12)$$

In addition, the total number of beacon packets and the number of lost beacon packets can be obtained with

$$N_{totalbeacon} = \frac{T_{duration}}{T_{beaconpd}}. \quad (4.13)$$

$$N_{BL} = \frac{\Sigma T_{BL}}{T_{beaconpd}}. \quad (4.14)$$

Taking Eq. 4.13 and Eq. 4.14 into account,  $P_{BL}$  can also be expressed as

$$\begin{aligned} P_{BL} &= \frac{N_{BL}}{N_{totalbeacon}} \\ &= \frac{\Sigma T_{BL}}{T_{duration}}. \end{aligned} \quad (4.15)$$

Considering Eq. 4.15, Eq. 4.12 can also be expressed as

$$P_{BL} = \frac{T_{wake} - T_{beaconpd} \cdot R_{wakeratio}}{T_{wake} - T_{beaconpd}}. \quad (4.16)$$

Substituting Eq.4.16 into Eq.4.7, the calculation of  $P_{idleBL}$  based on  $R_{wakeratio}$  is obtained

$$\begin{aligned} P_{idleBL} &= P_{wake} \cdot \frac{T_{wake} - T_{beaconpd} \cdot R_{wakeratio}}{T_{wake} - T_{beaconpd}} + \left[1 - \frac{T_{wake} - T_{beaconpd} \cdot R_{wakeratio}}{T_{wake} - T_{beaconpd}}\right] \cdot \\ &\quad \left(P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot T_{wake}\right). \end{aligned} \quad (4.17)$$

Fig. 4.2 presents the comparison among  $P_{idle}$ ,  $P_{idleBL}$  and the measurement  $P_{measure}$ , taking  $R_{wakeratio}$  and  $P_{BL}$  into account. The gap between the curves with  $P_{idle}$  and  $P_{measure}$  is acceptable when the beacon period is small, but it slightly rises with the increase of the beacon period to 4.25mW, accounting for 45.01% of the  $P_{measure}$ . On the other hand, the deviation between the curve with  $P_{idleBL}$ , which considers the effect

of  $R_{wakeratio}$ , and the curve with  $P_{measure}$  stay the same during the sample space, only 2.83mW occupying 24.22% of the measurement. So the effect of beacon lost has to be taken into account when calculating power consumption in idle pattern. Furthermore, the comparison between  $R_{wakeratio}$  and  $P_{BL}$  is also present in Fig.4.2. The curves almost are coincident especially when the beacon period is larger, and the average deviation is only 0.2%, occupying 14.14% of the average value of  $P_{BL}$ . So that means the assumption in Eq.4.15 is basically reasonable and acceptable.

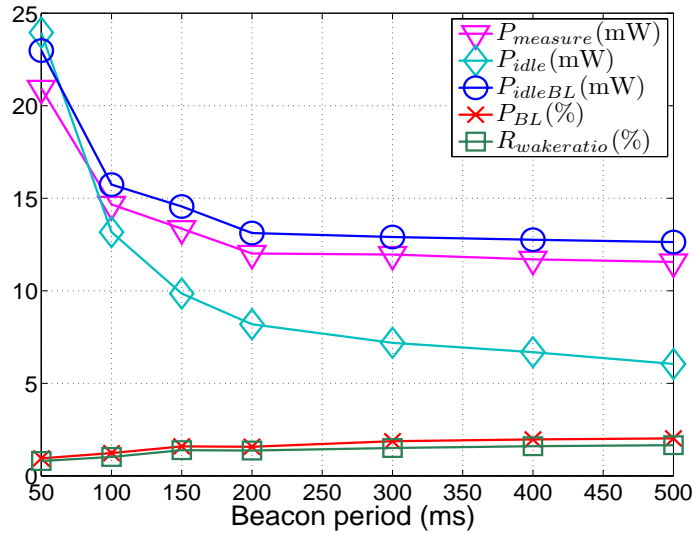


Figure 4.2: Validation of the theoretical calculation in the idle mode.

## 4.2 Power calculation in the traffic mode

The calculation of power consumption in a traffic mode can be categorized into two types, the transmitter mode and the receiver mode, since a transmitter and a receiver exhibit different actions during the interactions. And normally, the power for transmission, denoted by  $P_{TX}$  is higher than that for receiving, denoted by  $P_{RX}$ .

In 802.11, Request to Send (RTS) and Clear to Send (CTS) are the control signals, which are supposed to decrease or avoid frame collisions by temporarily reserving the medium. The atomic interactions between a transmitter and a receiver are shown in Fig.4.3, which includes a RTS, a CTS, a data frame and an acknowledgement, separated by Short Interframe Spaces (SIFS). The arrows indicate transmission directions. And the solid frames represent transmissions with a power level of  $P_{TX}$ , while the dashed ones mean receipts with a power level of  $P_{RX}$ . After the Distributed Coordination function Interframe Space (DIFS), the back off mechanism is used for STAs to compete the idle medium to avoid

collisions.

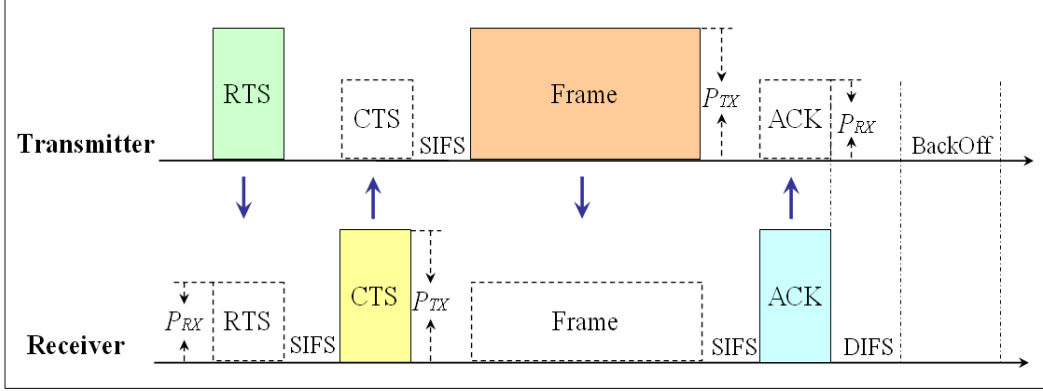


Figure 4.3: The atomic interactions between a transmitter and a receiver.

To calculate the average power consumed by the transmitter, the time for transmission, denoted by  $T_{TX,T}$ , and the time for receipt, denoted by  $T_{RX,T}$  have to be firstly defined. Referring to Fig.4.3, we get

$$T_{TX,T} = T_{RTS} + T_{Frame}, \quad (4.18)$$

$$T_{RX,T} = 3 \cdot T_{SIFS} + T_{ACK} + T_{CTS}, \quad (4.19)$$

where  $T_{RTS}$ ,  $T_{CTS}$ ,  $T_{frame}$ , and  $T_{SIFS}$  represent the time for RTS, CTS, frame transmission, and SIFS in the atomic interaction, respectively. And then the formula to calculate power consumption of the transmitter is

$$\begin{aligned} P_{transmitter} &= \frac{P_{TX}T_{TX,T} + P_{RX}T_{RX,T}}{T_{atomic}} \\ &= \frac{P_{TX}(T_{RTS} + T_{Frame}) + P_{RX}(3 \cdot T_{SIFS} + T_{ACK} + T_{CTS})}{T_{TX} + T_{RX} + T_{Frame} + T_{DIFS} + T_{backoff}}. \end{aligned} \quad (4.20)$$

where  $T_{atomic}$  specifies the period of the atomic interaction. In the same way, the formula of power calculation for the receiver can be obtained

$$\begin{aligned} P_{receiver} &= \frac{P_{TX}T_{TX,R} + P_{RX}T_{RX,R}}{T_{duration}} \\ &= \frac{P_{TX}(T_{CTS} + T_{ACK}) + P_{RX}(3 \cdot T_{SIFS} + T_{RTS} + T_{frame})}{T_{TX} + T_{RX} + T_{Frame} + T_{DIFS} + T_{backoff}}. \end{aligned} \quad (4.21)$$

If the parameters in Eq. 4.20 and Eq. 4.21 are assigned with the default values in Tab.4.2, the results are  $P_{transmitter} = 764$  mW and  $P_{receiver} = 504$ mW. Moreover, the traffic mode specifies the particular scenario that 1 Mega bytes data is downloaded or uploaded, fol-

Table 4.2: Default values for the power calculation in the traffic mode

<i>Parameter</i>	<i>Default Value</i>
$T_{DIFS}$	50 $\mu s$
$T_{SIFS}$	10 $\mu s$
$T_{RTS}$	640 $\mu s$
$T_{CTS}$	592 $\mu s$
$T_{ACK}$	592 $\mu s$
$T_{backoff}$	80 $\mu s$
$P_{TX}$	800 mW
$P_{RX}$	484 mW

lowed by an idle state for 30 seconds, then the process repeats afterwards. The formula to calculate power consumption for the scenario are as follows

$$P_{uplink} = \frac{P_{idle} \cdot T_{idle} + P_{transmitter} \cdot T_{traffic}}{T_{idle} + T_{traffic}}, \quad (4.22)$$

$$P_{downlink} = \frac{P_{idle} \cdot T_{idle} + P_{receiver} \cdot T_{traffic}}{T_{idle} + T_{traffic}}, \quad (4.23)$$

where  $P_{idle} = 3.29$  mW,  $T_{idle} = 30$  seconds,  $P_{transmitter} = 764$  mW,  $P_{receiver} = 504$  mW, and  $T_{traffic}$  is available in Tab.4.3. And then both of the formula of power calculation can be validated by comparing with the measurement results.

Table 4.3: Dynamic parameters in the traffic mode

<i>Sequence</i>	<i>PHY_Rate(Mbps)</i>	<i>UDP_Rate(kbps)</i>	<i>T_traffic(s)</i>
1	11	583	12.00
2	11	1167	6.00
3	11	1754	4.00
4	11	2335	3.00
5	54	2932	2.40
6	54	3523	2.00
7	54	4099	1.71
8	54	4671	1.5

The validation of the formula for the traffic mode is shown in Fig. 4.4. Both Eq. 4.22 and Eq. 4.23 can match the curves based on the measurement results regarding uplink and downlink respectively. With regard to uplink, at first the power consumption of  $P_{uplink}$  is a little higher than that of the measurement, denoted by  $M_{uplink}$ , followed by a cross point at 2335 kbps. And then  $M_{uplink}$  indicates more power consumption than  $P_{uplink}$ . However, the maximal deviation of power consumption along these two curves

is 5.02mW, only accounting for 8.66% of the experiment result. Furthermore, regarding downlink, the power consumption with  $M_{downlink}$  is a little larger than that with  $P_{downlink}$  along the whole range of the data rate, and the maximal deviation between these curves is 8.72mW, occupying 6.28% of the experiment result.

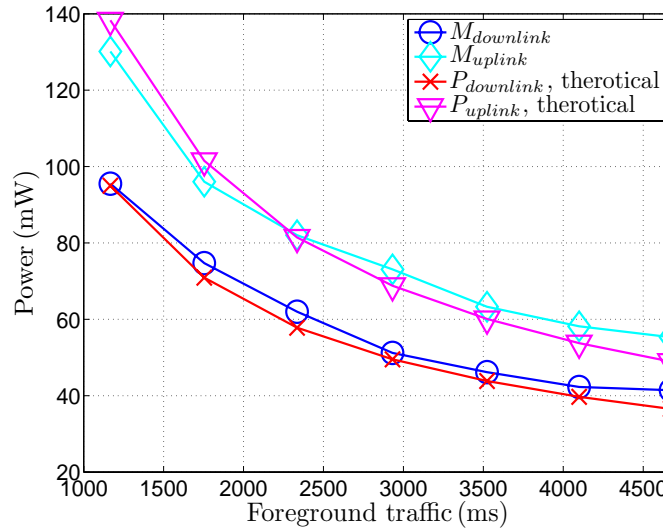


Figure 4.4: Validation of the theoretical calculation for the traffic pattern.

## Chapter 5

# WLAN Power Visualization Tool

The WLAN power visualization tool (WPVT) aims to facilitate the WLAN PCAF investigation in a smart way instead of the time consuming experiments. As long as a series of parameters are configured, the tool will present a curve of power consumption along the increase of a specific PCAF. Basically, mathematical calculations for both the idle pattern and the traffic pattern, discussed in Chapter 4, act as the foundations of WPVT.

### 5.1 Requirement analysis

#### 5.1.1 Functional requirement

The functional requirements specifies the basic functionalities a system/application has to support, which exactly defines what the system/application can do. WPVT, as a illustration tool of WLAN power consumption, has to cover all the PCAFs and aspects, which are discussed in Chapter 3 and Chapter 4. So the main features provided by the WPVT is as follows.

- It has to support the illustration of WLAN power consumption for both idle pattern and traffic pattern. The power consumption in both of these pattern has been investigated in both a experimental way and a mathematical way, since these pattern is the most common WLAN activities in practice. This is the basic feature required for the WPVT.
- Background traffic has to be taken into account during a illustration, since it makes the performance of power save management more susceptible compared with other PCAFs based on the experimental results. Moreover, as shown in the experiments, the effect over power consumption of background traffic depends on both the di-



rection and the data rate. So the options of both the mode and the data rate of background traffic should also be available in WPVT.

- A power consumption curve, where the function is power while the independent variable is one of the PCAFs, represents the relationship between power consumption and a specific PCAF. Which PCAF acts as the independent variable along the x axis can be specified by a user.
- After the the curve is rendered, a couple of text fields give the details of the configuration information and the power illustration results, which correspond to the curve.

### 5.1.2 Non-functional requirement

Non-functional requirement is used to estimate the operation of a system/application, which specifies the other aspects instead of basic functionalities provided by a system/application, such as reliability, security, etc. With regard to WPVT, the non-functional requirements include

- Reserve interfaces for extension or further development, which lies in the architecture of a system/application. It is an essential requirement for almost all system-/applications, due to the limitation of time, budget, and workload, etc. Generally, further research or investigation produces more interesting observations. The reserved interfaces are used for further development to take more investigations into account. The interface of extension in WPVT is the abstract class, *WlanCalculation*, which defines the basic functions for calculations in different mode.
- Exception handling. An exception in a system/application probably leads to an error or even a crash, and it occurs less frequently and in a unknown way. So exception handling is used to catch any exception and make the performance stable. Since WPVT is a interactive tool, where a series of parameters are required to be assigned with specific values, an abnormal configuration may indicate an exception. So an regular expression is used to check the validation of any configuration. For the illegal input, an prompt pops up that require the user to retry.
- Avoid violating copyright. It is required to respect copyrights, including everything in software development, such as source code, class diagram, framework etc. Ideas can be generate according to the available, but can not be cited without declaration, otherwise it is classified as a plagiarism.

## 5.2 Design specification

With respect to the design issue for WPVT, Model-View-Controller (MVC) [2], shown in Fig.5.1, is adopted to create the architecture of the tool, which separates data model, user interface and logic control into three distinct parts. It aims to decrease the coupling among these three parts, which means the modification of any part can only exert minimal impact on the other parts. The basic logics in MVC is that an event initiated by a user triggers the controller, which will affect view or data model accordingly. On the other hand, the view depends on the data stored the data model, and the view will be updated as long as the data model is changed.

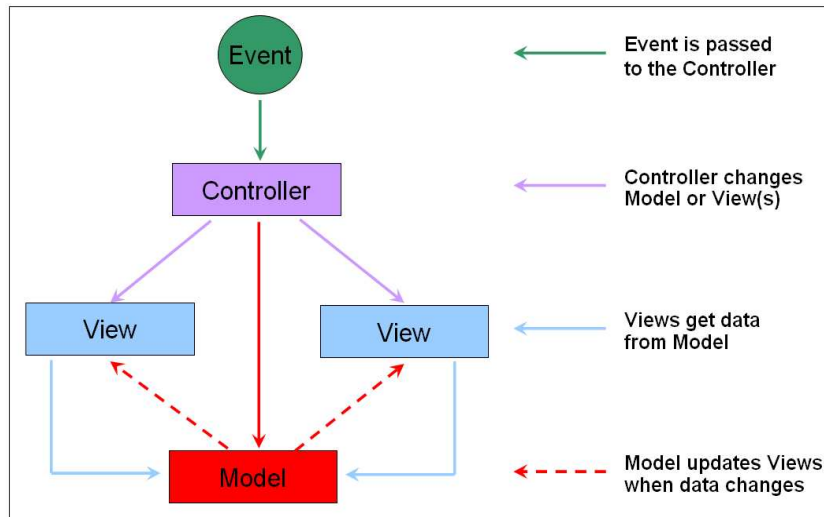


Figure 5.1: MVC architecture [2].

Applying the MVC into WPVT, an event action listener is triggered each time when a configuration is finished, and then the controller, which implements the main logics of the tool using the mathematical calculation, passes the configuration data and the calculated result of power consumption to the data model, which makes the WPVT render the curve according to the current data set. So each time, as long as a new configuration is submitted, the corresponding curve of power consumption will be rendered.

Based on the MVC structure, a UML class diagram [24] for MPVT is created to reflect the static structure of the system, with the relationships among these classes, such as reference, generalization, association, as shown in Fig. 5.2. The basic functions provided and roles played by each class are listed in Tab. 5.1.

In addition to the UML class diagram, which specifies the basic structure in terms of classes, a series of UML sequence diagrams [24] are also required to discover interac-

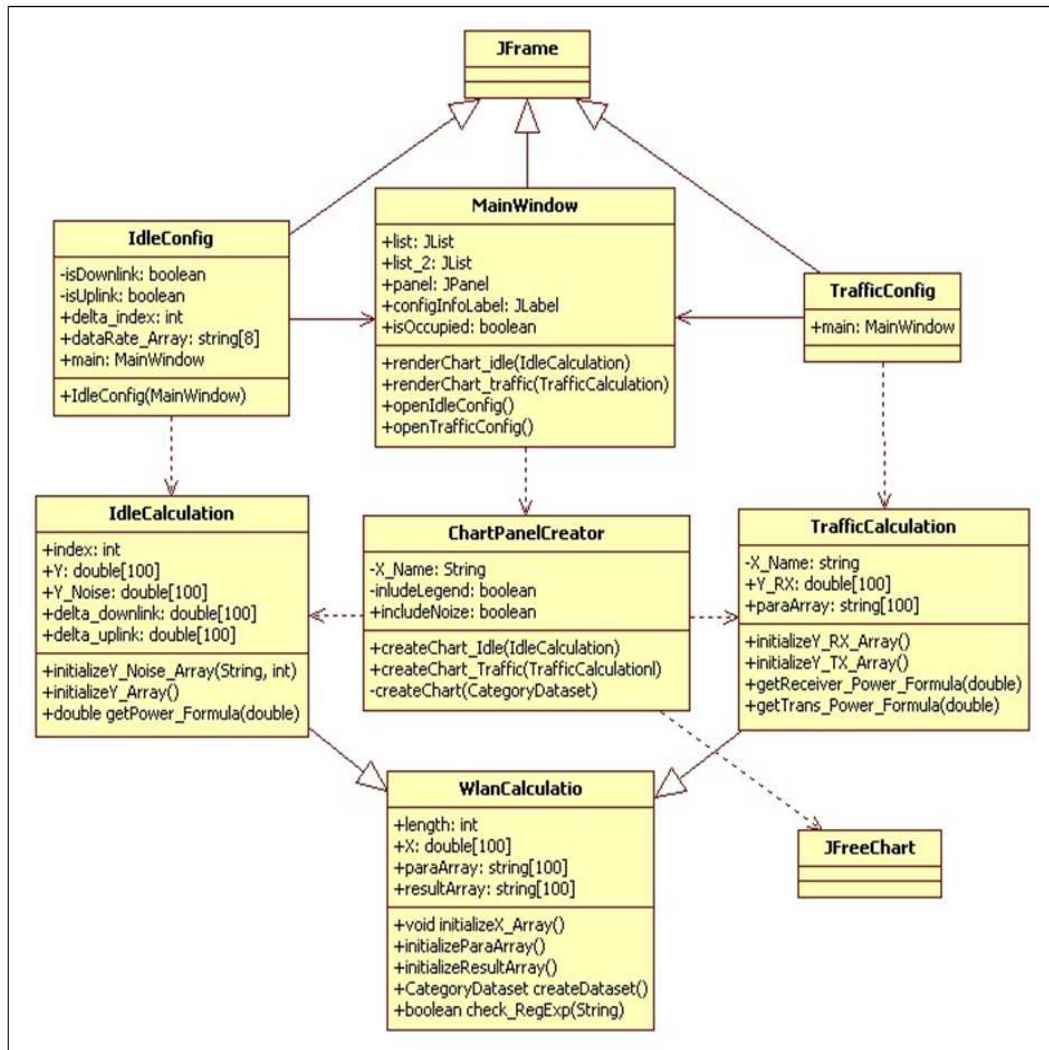


Figure 5.2: UML class diagram.

tions among specific objects. In this way, the methods for the classes can be discovered to implement the behaviors provided by the classes.

Table 5.1: Functions and descriptions of the Java classes

<i>Class</i>	<i>Description</i>	<i>Function</i>
<i>JFrame</i>	A standard class in Swing component	Creating windows
<i>IdleConfig</i>	Extended from <i>JFrame</i>	Creating the interface for idle configuration
<i>TrafficConfig</i>	Extended from <i>JFrame</i>	Creating the interface for traffic configuration
<i>MainWindow</i>	Extended from <i>JFrame</i>	Main entry of the system
<i>WlanCalculation</i>	An abstract class	Defining the basic calculation for all patterns
<i>IdleCalculation</i>	Extended from <i>WlanCalculation</i>	Handling the calculation for idle pattern
<i>TrafficCalculation</i>	Extended from <i>WlanCalculation</i>	Handling the calculation for traffic pattern
<i>JFreeChart</i>	A standard java chart library	Rendering the curve
<i>ChartPanelCreator</i>	Coordination class	Invoke <i>JFreeChart</i> based on different pattern

Fig. 5.3 presents a normal interaction sequence among objects for the idle mode. Firstly, *MainWindow* asks *IdleConfig* to display the configuration panel by invoking the method of *openIdleConfig()*. And then with the configuration information provided by a user, *IdleConfig* initiates the data mode in a couple of steps. The first step is that after the independent variable is distinguished, in order to specify the range and the interval of the independent variable, it invokes the methods of *setX\_lower()*, *setX\_upper()*, *setInterval()* and *initializeX\_Array()* provided by *IdleCalculation*. The second step is that when *initializeX\_Array()* is called, *IdleCalculation* invokes its own method of *getPower\_Formular()* to initialize the function value along the y axis. The final step is used to update the information which is also shown corresponding to the curve, by calling *initializeParaArray()* and *initializeResultArray()*. After the data model is initialized, the coordination object of *ChartPanelCreator* is called by *MainWindow* with *createChart\_Idle()*, and then it fetches the data from data model by invoking *createDataset()*. As long as the data set used for a curve is available, *ChartPanelCreator* asks *JFreeChart* to render the curve by invoking *createChart()*. The interaction sequence for traffic pattern is almost the same as that for idle pattern, except that the power calculations for both the receiver and the transmitter are needed, which indicates different data models and method invocations.

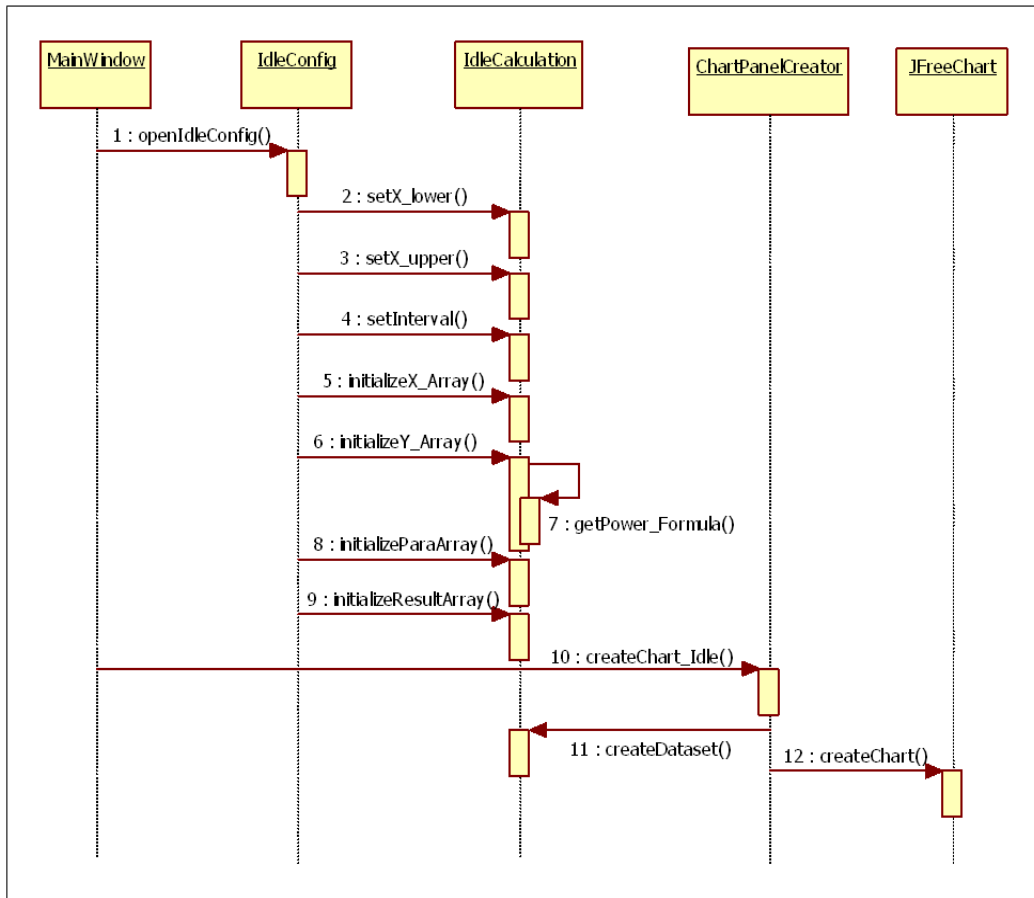


Figure 5.3: UML sequence diagram for the idle mode.

### 5.3 Implementation

The implementation part is categorized into Model Driven Development(MDD) [25]. The concept of MDD means an abstract model of a system is created to represent the basic structure of the system, which generally can not be executed directly, and then an executable model is transformed from the abstract model. IBM Rational Software Architect [4] is chosen as the development environment for WPVT, since it supports the feature of model transformation from a class diagram to java source codes. The transformation processes are shown in Fig.5.4.

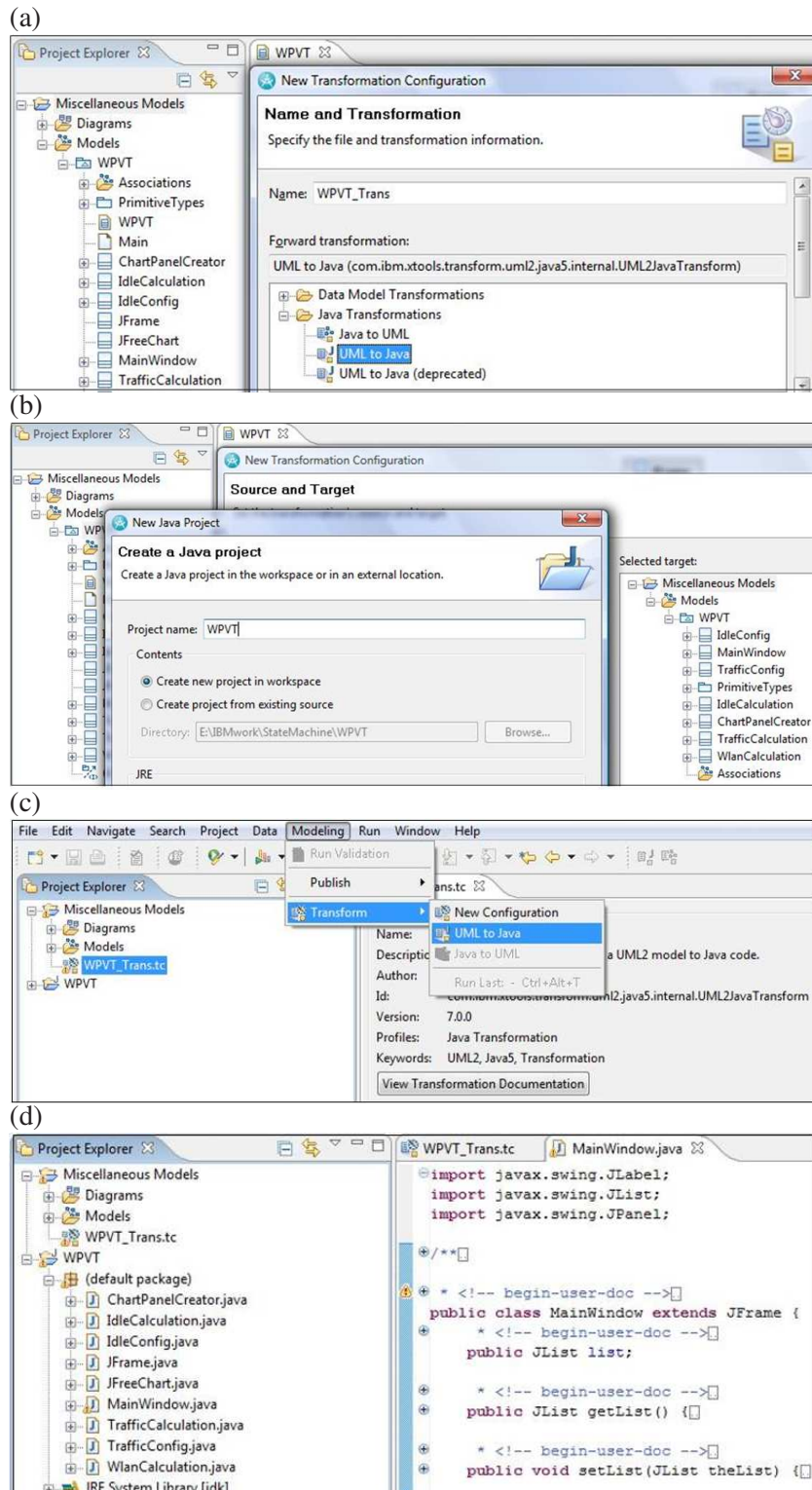


Figure 5.4: Model transform from the class diagram to java codes.



JFreeChart [26] is adopted as the main solution to display a curve of power consumption, which is a professional chart library written in java, and provides a series of open application programming interfaces (APIs) for users. Since JFreeChart is not a standard library predefined in Java Development Kit (JDK), it is required to import the external jar library into the development environment, as shown in Fig.5.5. And then the APIs provided by JFreeChart are free to be used in the same way as the predefined standard libraries.

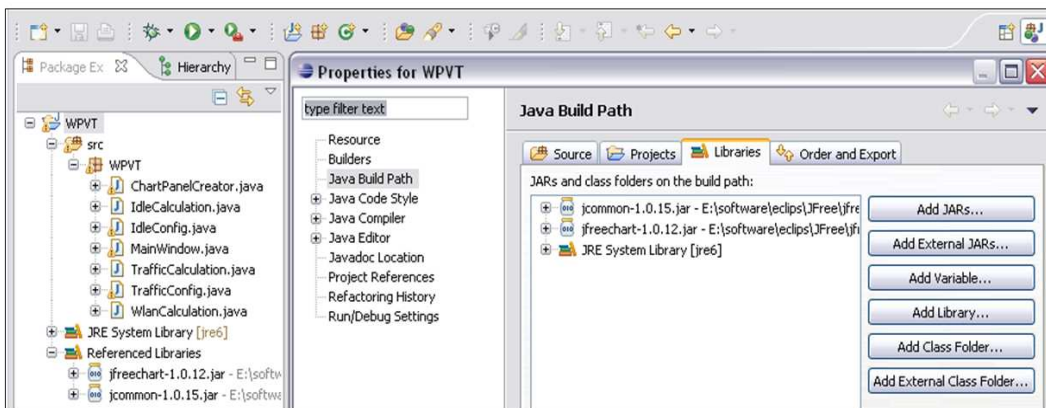


Figure 5.5: Add external jar library for JFreeChart.

After implementing WPVT, taking portability into account, it is advisable to generate an executable application of WPVT. The normal way to solve the problem is creating an executable Java ARchive (JAR) file [27], which can merge multiple files into a single file. Eclipse development platform provides a way to generate an executable JAR file based on a java project, but it does not work for the imported external Jar library. Finally, Fat Jar Eclipse plug-in [28], version 0.0.13, is adopted to solve the problem, which supports imported external Jar libraries. The basic operations to generate the executable JAR file, *WPVT.jar*, are shown in Fig.5.6.

(a)



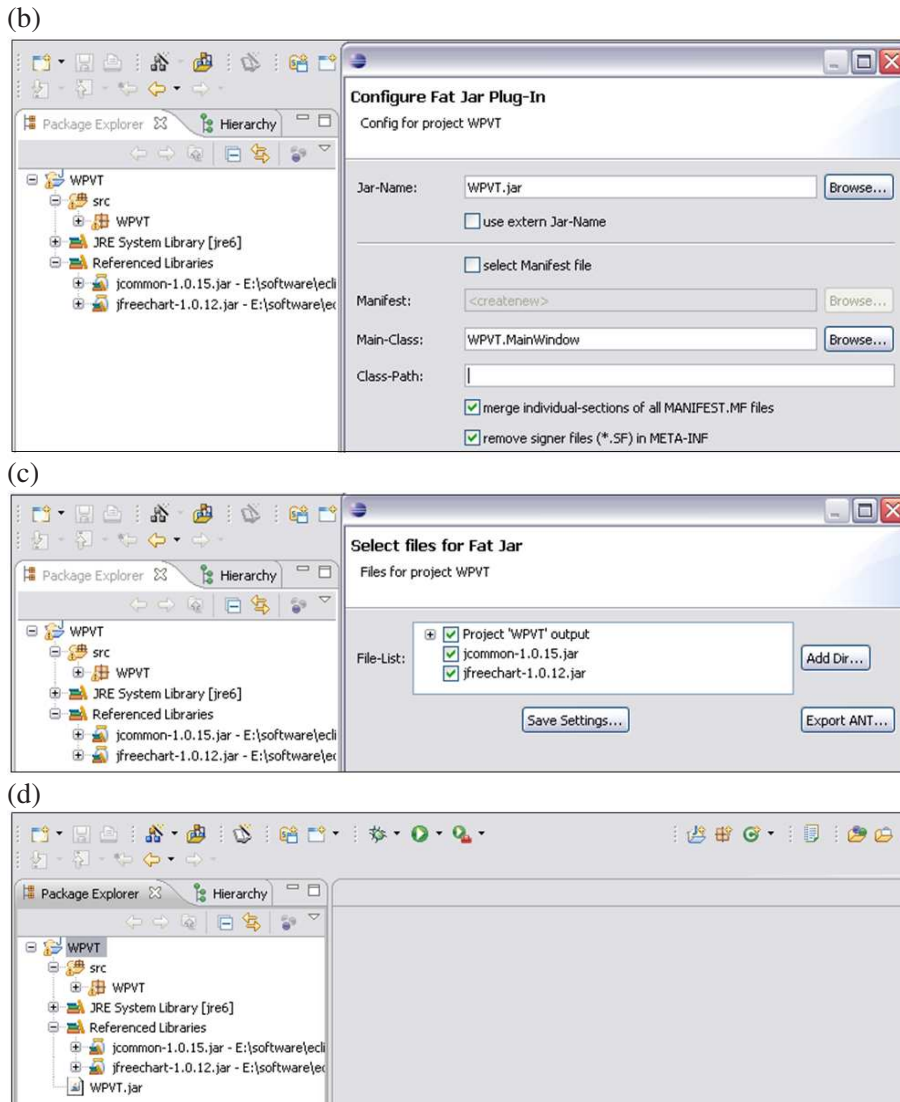


Figure 5.6: Export an executable jar file using Fat Jar eclipse plug-in.

## 5.4 Testing

The testing part is used to check whether WPVT has covered all the functions or features specified in the section of requirement analysis.

### 5.4.1 Testing of WPVT for the idle mode

Three pairs of sub-figures in Fig.5.7 displays the results of power consumption illustration for idle pattern.



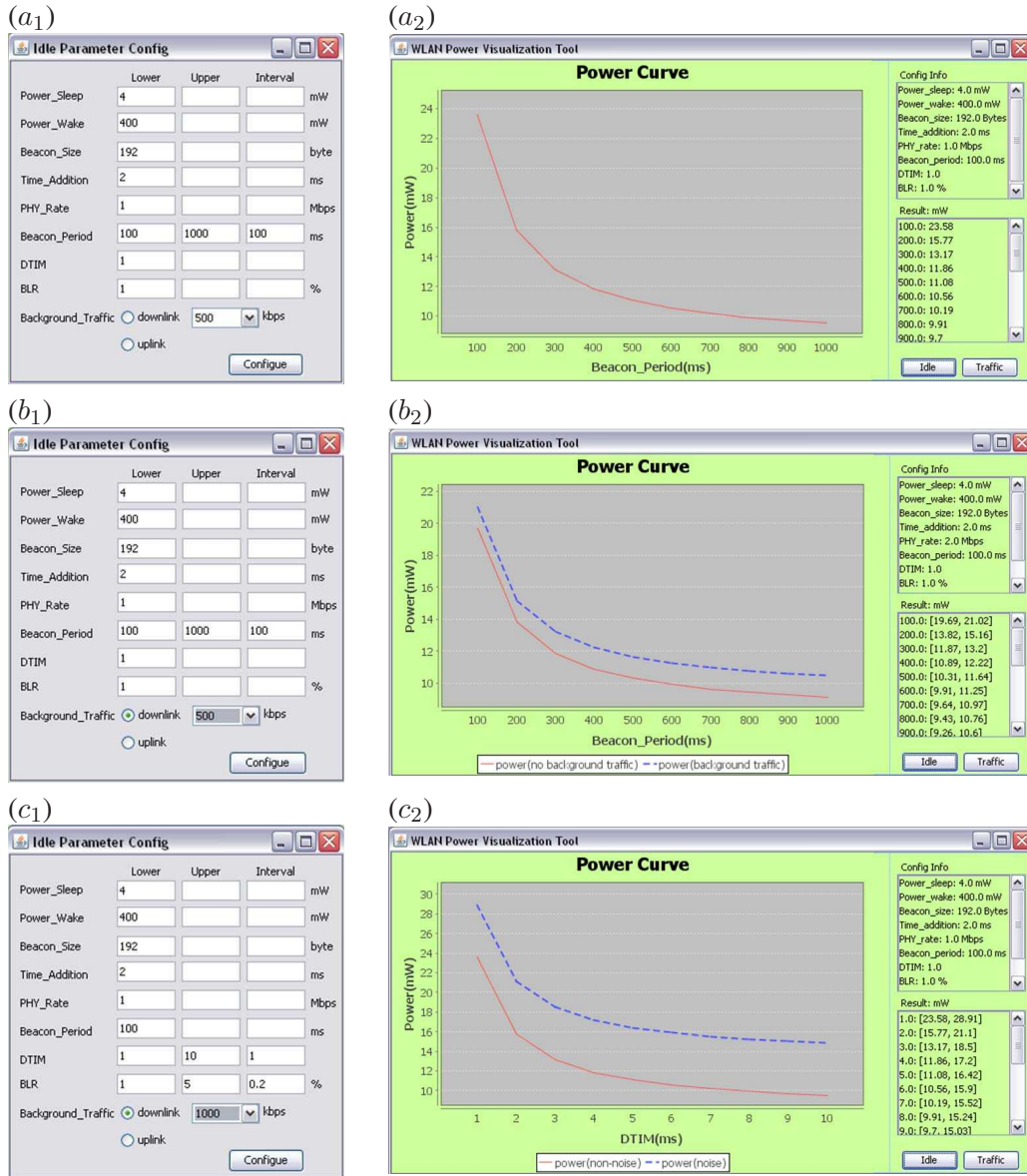


Figure 5.7: Testing for WPVT illustration in the idle mode.

○ No background traffic effect is involved. The solid curve in Fig. 5.7(a<sub>2</sub>) displays the relationship between power consumption and beacon period, based on the configuration in Fig. 5.7(a<sub>1</sub>), where the beacon period is appointed as the independent variable from 100ms to 1000ms with the interval of 100ms. Beside the curve, the configuration information and the accurate values of power consumption samples are displayed in the right top area and the right bottom area, respectively.

- Background traffic effect is taken into account. The testing shown in Fig.5.7(b) is almost the same as that shown in Fig.5.7(a) except that the background traffic effect is taken into account. In Fig.5.7(b<sub>2</sub>), the solid line represents the power curve only based on theoretical calculation, while the dashed one considered the effect of background traffic with a specific direction and a specific data rate. Actually the effect of background traffic is based on the experimental results of  $BG_{\gamma 1}$ , which affects power saving the most. So the normal testing results are supposed lie in the area between these two curves. Moreover, the configuration information and the power consumption range are correctly shown.
- If multiple parameters are specified as independent variables without intention, how the collision is solved? The parameter, which appears above in the configuration panel, is supposed to be the independent variables, shown in Fig.5.7(c<sub>1</sub>)-(c<sub>2</sub>).

### 5.4.2 Testing of WPVT for the traffic mode

A couple of sub-figures in Fig.5.8 represent the testing results of power consumption illustration for traffic mode.

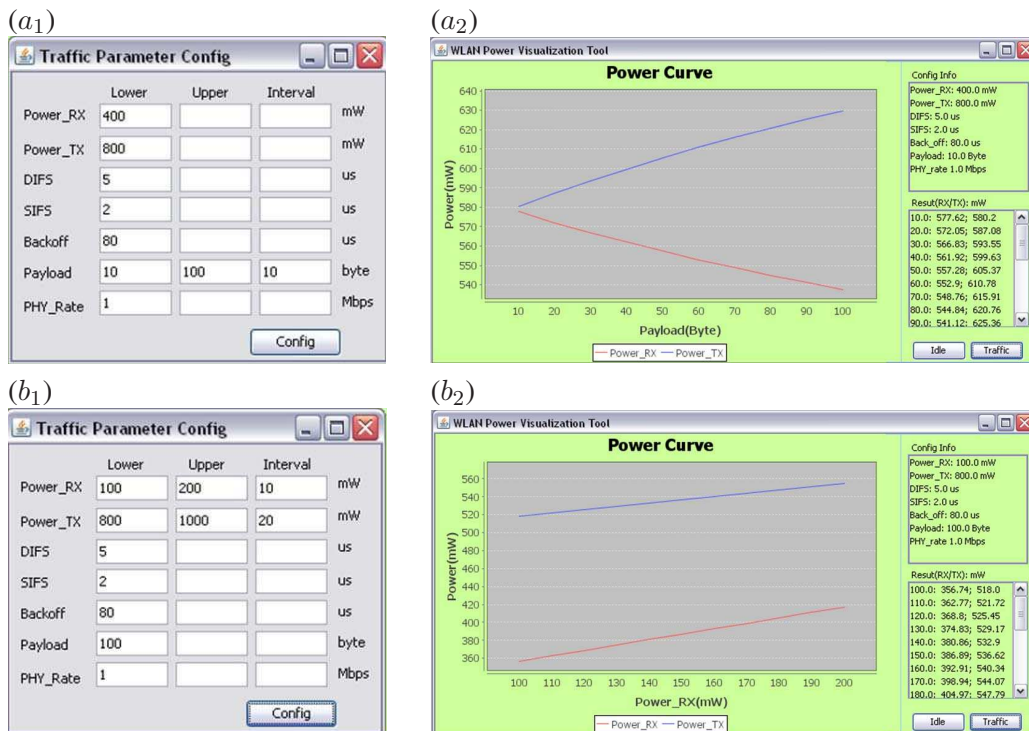


Figure 5.8: Testing for WPVT illustration in the traffic mode.

- Testing for traffic model based on the normal configuration. As shown in Fig.5.8(a<sub>1</sub>)

and Fig.5.8(a<sub>2</sub>), based on the normal configuration, two curves are displayed in the curve area. The top one illustrates the power consumed by the transmitter, while the other is for the receiver. Moreover, the size of payload plays the role of the independent variable from 10 bytes to 100 bytes with the interval of 10 bytes. In addition, the configuration information and the power consumption values for both the transmitter and the receiver are correctly displayed.

- If multiple parameters are specified as independent variables without intention, in the same way as for idle mode, the parameter, which appears above in the configuration panel, acts as the independent variables, shown in Fig.5.8(b<sub>1</sub>)-(b<sub>2</sub>).

### 5.4.3 Validation of configuration

Normally, there are two types of invalid configurations, as shown in Fig.5.9. The first type, shown in Fig.5.9(a), is that a parameter value is not categorized into a real number, such as a letter. And the validation process of a real number is based on a regular expression. The second type, shown in Fig.5.9(b), is the missing of configuration for one or more parameters. Any kind of invalid configuration leads to a prompt, which requires a user to re-configure the corresponding parameters.

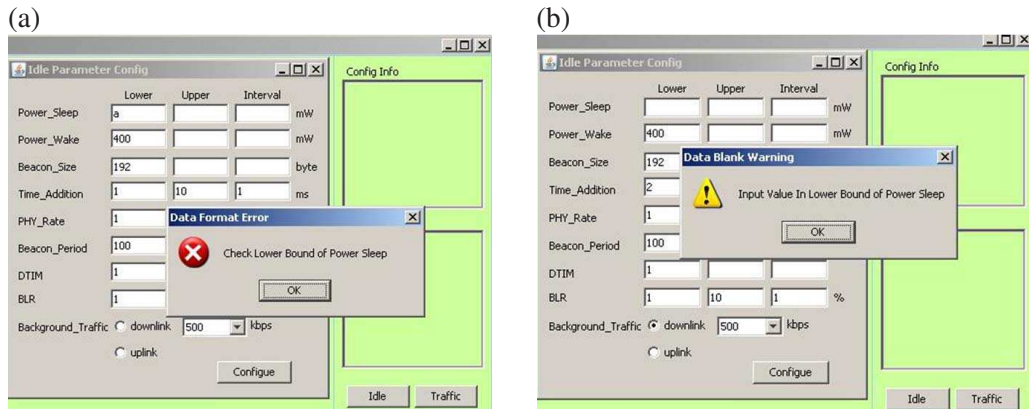


Figure 5.9: Configuration validation.

## Chapter 6

# The Proposed Algorithm for Power Saving in WLANs

### 6.1 Background

Based on the experiment and the mathematical calculation, discussed before, the power management mechanism especially makes sense when a STA is in idle state. If a STA is always involved in transmitting/receiving data, the power consumed by the STA absolutely lies in a higher level. As specified in the traffic mode, the most common WLAN activity on a hand held device is transmitting/receiving a specific amount of data, followed by an idle state for a period, and then the process repeats. Even in the traffic mode, power conservation has to rely on the idle state. In order to achieve less power consumption, the doze time has to be maximized while the awake time has to be minimized, provided that the throughput does not degrade in an acceptable extent.

A beacon frame is a very important element in 802.11 WLANs, which not only advertises the services provided by the BSS, but also achieves the transitions of a STA from a doze state to an awake state, or versus visa. Regarding the legacy polling scheme, beacon frame plays an essential role to coordinate the interactions among an AP and all the associated STAs. Normally, a beacon frame informs a STA about the arrival time of the next beacon frame, so the STA has to wake up earlier,  $T_{addition}$ , than the expected arrival time in order to capture the beacon frame, and then it transits to sleep state after parsing all the information included in the beacon frame. On the contrary, if the STA wakes up later than the arrival time of the beacon frame, it has to stay in awake state for the whole beacon period until the following beacon frame can be captured. Since  $T_{beaconpd}$  is 50 times larger than  $T_{wake}$ , so a beacon lost introduce much more power consumption.

Synchronization (SYN) clock plays a very important role in a distributed system, such as a 802.11 WLAN, since without the help of a common reference, all the equipments can not interact with each other only relying on their local clocks. Beacon frame bridges the interactions among an AP and all the associated STAs, since both the current time stamp of the AP and the expected arrival time of next AP are broadcasted. Due to physical characters, any clock definitely produces time drift [29], forward or backward. The time drift of an AP makes the real arrival time of next beacon frame forward or backward, which means the reliability of the AP decreases.

Taking time drift into account, the fixed  $T_{addition}$  has an obvious drawback. In order to capture beacon frames and avoid beacon lost,  $T_{addition}$  has to be larger to guarantee that a STA can wakes up earlier than the unpredicted arrival time. The larger value of  $T_{addition}$  can definitely increase the reliability of the AP, but causes more power consumption. And on the other hand the smaller value of  $T_{addition}$  can save more power, but may confront with a risk of beacon lost, which can also lead to more power consumption. Moreover, time drift is less obvious in an AP with higher quality than that with lower quality. If  $T_{addition}$  can be dynamically adjusted according to time drift, the beacon lost probability can decrease and meanwhile more power can be saved. The adaptive  $T_{addition}$  acts as a key role in the proposed algorithm, which will be addressed in the following sections.

## 6.2 Proposed algorithm

As shown in Fig. 6.1, there are three types of awake states for a STA. If no time drift of the AP appears, the beacon frame arrives at the expected time, so the STA can successfully capture the beacon frame. This is the most common awake state, which is defined as Awake Type 1. The other two awake types are defined for the time drift of the AP,

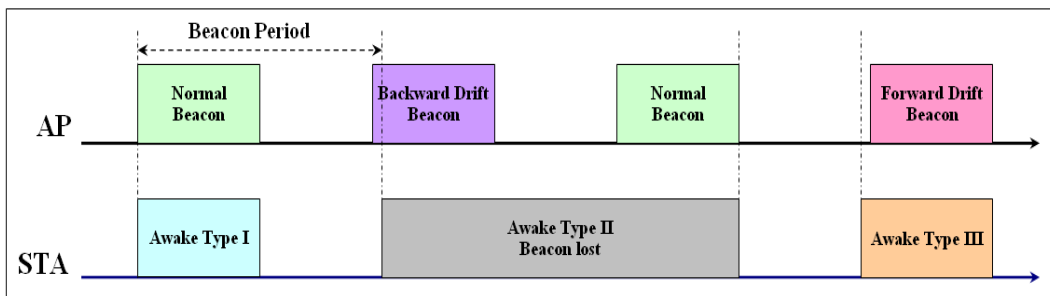


Figure 6.1: Awake types of a STA in the 802.11 WLAN.

which depends on the direction of the time drift. If the time drift is backward, which means the beacon frame arrive earlier than the waking up of the STA, the beacon frame

will be lost. And then the STA has to stay awake for the whole beacon period, which is defined as Awake Type 2. By contrast, Awake Type 3 indicates the forward time drift of the AP, which means the beacon frame arrives later than expected, so the STA can absolutely capture the beacon frame, which only causes a little power consumption more than Awake Type 1.

The proposed algorithm has to depend on these awake types, and trigger different actions based on different awake types. The rules of the algorithm are shown in Fig. 6.2, and described as follows.

- If the STA wakes up with Awake Type 1, nothing is required to do.
- If the STA wakes up with Awake Type 2, the SYN clock moves backward with the a varied time interval,  $2^{n-1} \cdot \Delta$ , where n represents the number of times for a series of consecutive adjustments, and  $\Delta$  denotes the time interval for each adjustment.
- If the STA wakes up with Awake Type 3, the SYN clock moves forward with the a varied time interval,  $2^{n-1} \cdot \Delta$ , where both n and  $\Delta$  denotes the same meaning as that for Awake Type 2.

There are a couple of limitations regarding the rules. Firstly, the algorithm provides a tiny adjustments of the SYN clock to make a platform dynamically adapt to the time drift of an AP, so the adjustment scale has to be in an acceptable bound, which means n can not be arbitrarily enlarged. Moreover, both the upper bound of n and  $\Delta$  have to depend on

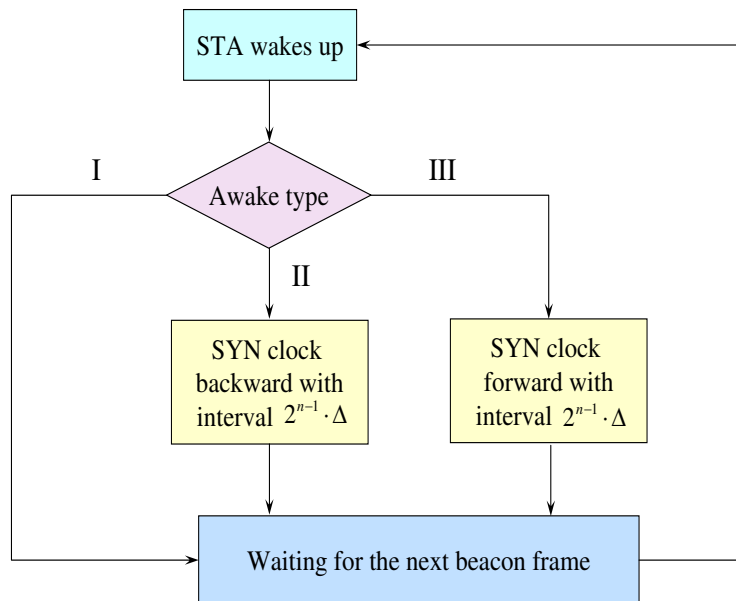


Figure 6.2: The flow chart of the proposed algorithm.

specific platforms. Finally, the adjustment of SYN clock is difficult to be implemented on specific platforms, and the adjustment may negatively affect other application which are also based on the same SYN clock.  $T_{addition}$ , which is a independent particular parameter for WLAN application, can be used to implement the adjustment according to the time drift. Moreover,  $T_{addition}$  is also involved in mathematical calculation of power consumption in Chapter 4, which can facilitate the mathematical comparison between the proposed algorithm and the previous one.

## 6.3 Performance evaluation

### 6.3.1 Time drift analysis

In order to prove the proposed algorithm is more efficient with respect to power conservation, the statistical analysis of the real time drift is supposed to be basis for the following mathematical calculations, which unfortunately is in a lack in the prior research. So general air logs for specific scenarios have to be generated, followed by extracting relevant information and analyzing a series of data using specific tools. The details about data collection and data analysis are described as follows.

Firstly, the time drift samples have to be collected for different testing scenarios. The first one is based on default configuration of the test bed and no background traffic is exerted, while the others are based on  $TC.1.05.Idle.BG_{\alpha}$ , which are configured with different direction and different data rate of the background traffic. Moreover, as explained before, it is of a special interest to study power conservation in idle mode, so all the scenarios mentioned in this subsection belong to idle mode.

In addition to the procedures specified in the test case, Wireshark has to be started at the beginning of each testing scenario, and sniff on the channel under test. The testing time has to be extended to guarantee more than 10000 beacon frames have been sniffed. After filtering all other kinds of frames except beacon frames, the log is exported and saved as a Comma Separated Value (CSV) file [30]. All the information included in each beacon frame except time stamp is removed with Excel, and then the real time intervals between consecutive beacon frames are calculated. So all the time drift can be got by subtracting beacon interval with real time interval. Then it is exported as a text file, which can be loaded into Matlab space. Based on the samples of the time drift, Matlab can generate a curve for the PDF of the time drift as shown in Fig. 6.3. For comparison, Fig. 6.3 also plots a theoretical normal distribution curve with the mean  $\mu$  and the standard deviation  $\sigma$  calculated according to the time drift samples. It can be seen from all subgraphs that



the actual PDF of the time drift approximates a normal distribution very well.

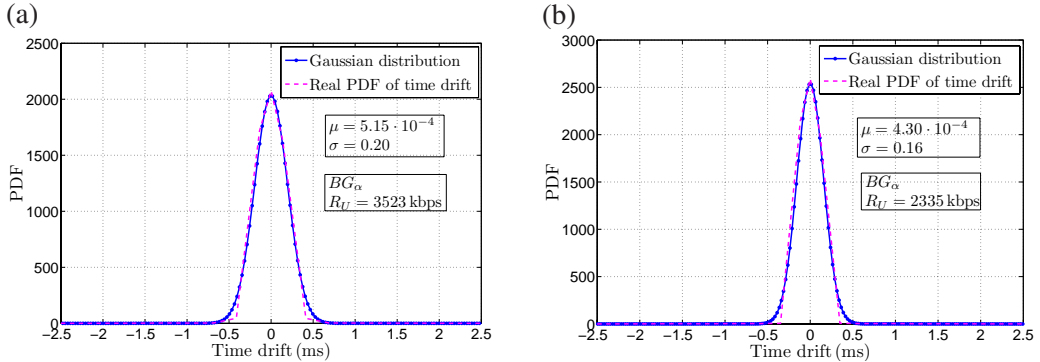
A Gaussian PDF is parameterized by the mean  $\mu$  and the variance  $\sigma^2$ . Such a PDF has several notable properties including the curve of the PDF is symmetric about the mean value  $\mu$  and the curve is more smooth with a larger variance  $\sigma^2$ . The decreasing variance from Fig. 6.3(a) to Fig. 6.3(e) indicates that the time drift samples are not highly centralized around the mean value, but more spread along the x axis compared with others. The curve with the 3523kbps uplink background traffic is more sharp than that with 2335kbps uplink background traffic, followed by 3523 kbps downlink background traffic, 2335kbps downlink background traffic, and no background traffic involved, which indicates the power consumption in a descending order. The conclusion is almost the same as the experiment results. And the time drift is the direct reason for more power consumption.

As shown in Fig. 6.3, the mean value of the normal distribution represents the average delay of a beacon frame. The mean for 3523 kbps uplink background traffic is the largest, followed by that with 2335 kbps uplink background traffic, 3523 kbps downlink background traffic, 2335 kbps downlink background traffic, and no background traffic. That means the uplink background traffic can cause a larger delay than the downlink background traffic, while a higher data rate of background traffic indicates a larger delay than the one with a lower data rate.

### 6.3.2 Mathematical argument

Based on the analysis in the previous subsection, the time drift of the SYN clock between an AP and a STA, can be described by the normal distribution

$$f_{T_{drift}}(t) = \frac{1}{\sqrt{2\pi} \cdot \sigma} \cdot e^{-\frac{(t-\mu)^2}{2\sigma^2}}. \tag{6.1}$$





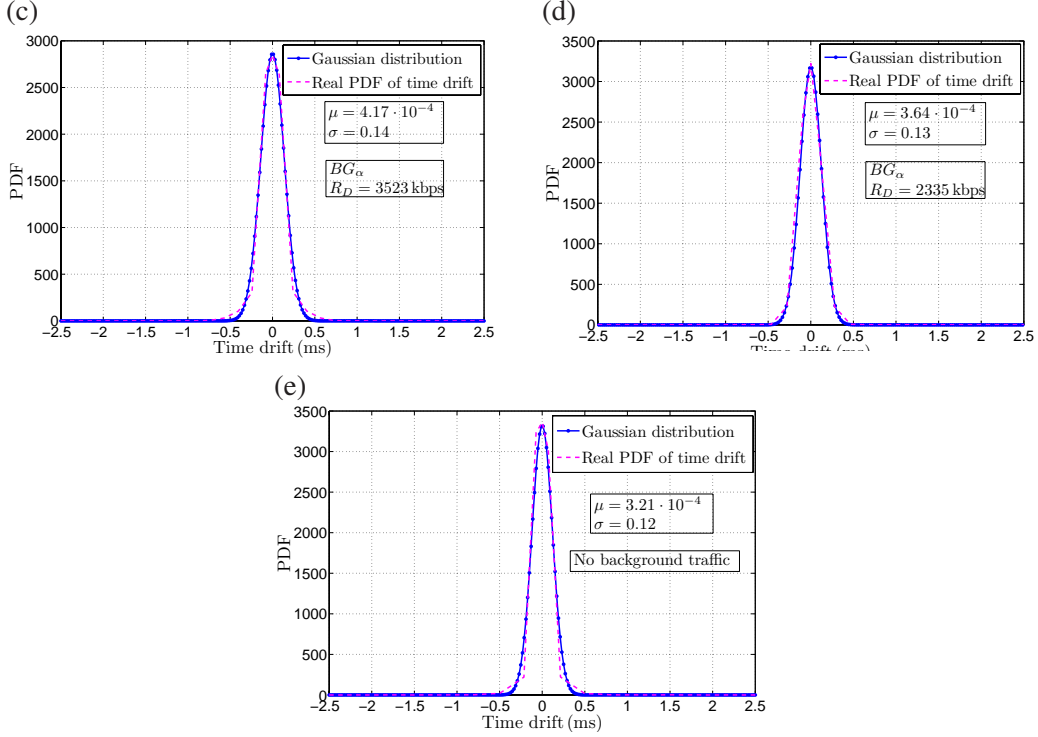


Figure 6.3: PDFs of the time drifts.

If the time drift moves backwards more than a given value of  $T_{addition}$ , a beacon frame has to be lost. Based on this point, the  $P_{BL}$  can be described as

$$P_{BL} = \int_{-\infty}^{-T_{addition}} f_{T_{drift}}(t) dt. \quad (6.2)$$

Taking the effect of the time drift into account and substituting Eq. 6.2 into the following equation, which has been discussed before,

$$P_{idleBL} = P_{wake} \cdot P_{BL} + (1 - P_{BL}) \cdot \left( P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{wakepartial} + T_{drift}) \right), \quad (6.3)$$

we have

$$P_{idleBL} = P_{wake} \cdot \int_{-\infty}^{-T_{addition}} f_{T_{drift}}(t) dt + (1 - \int_{-\infty}^{-T_{addition}} f_{T_{drift}}(t) dt) \cdot \left( P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{partial} + T_{drift}) \right). \quad (6.4)$$

Since the PDF of the time drift, denoted by  $f_{T_{drift}}(t)$ , is a normal distribution, we can further express Eq. 6.4 in a closed form as

$$P_{idleBL} = P_{wake} \cdot \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{\mu + T_{addition}}{\sqrt{2}\sigma}\right) + \left(1 - \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{\mu + T_{addition}}{\sqrt{2}\sigma}\right)\right) \cdot \left(P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{partial} + T_{drift})\right), \quad (6.5)$$

where  $\operatorname{erfc}(\cdot)$  represents the complementary error function. So the formula of power consumption for Legacy polling scheme is

$$P_{old} = \begin{cases} P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{partial} + T_{drift}), & -T_{addition} \leq T_{drift} \\ P_{idleBL}, & T_{drift} < -T_{addition}. \end{cases} \quad (6.6)$$

On contrary, the proposed new algorithm can minimize the negative effect of the time drift, since the SYN clock can dynamically adapt to the changes of the time drift. As shown in Fig. 6.3, the variance of the time drift is extremely small, so the each adjustment of SYN clock can obviously reduce the negative effect of time drift. In the ideal condition, it is assumed that there is no beacon frame lost due to the dynamically adaptive algorithm, based on which the power consumption for new mechanism can be roughly expressed as

$$P_{new} = \begin{cases} P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{partial}), & T_{drift} \geq 0 \\ P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{partial} + T_{drift}), & -T_{addition} \leq T_{drift} \leq 0 \\ P_{sleep} + \frac{P_{wake} - P_{sleep}}{T_{beaconpd}} \cdot (T_{addition} + T_{partial}), & T_{drift} < -T_{addition}. \end{cases} \quad (6.7)$$

After comparing Eq. 6.6 with Eq. 6.7, it is obvious that  $P_{new}$  is less than  $P_{old}$  when  $-T_{addition} \leq T_{drift}$ . On the other hand, if  $T_{drift} < -T_{addition}$ , the PDFs of the power  $P_{idleBL}$  computed according to Eq. 6.5 are depicted in Fig. 6.4 for different values of  $T_{addition}$ . For comparison, this figure also shows the PDF of the power for the new mechanism when the packet lost occurs. In this figure, it can be observed that the PDF of the power  $P_{idleBL}$  shifts to the left as increasing the value of  $T_{addition}$ . Such an observation implies that the mean value of the power decreases if increasing  $T_{addition}$ . When  $T_{addition}$  is larger than  $300 \mu s$ , the mean power  $P_{idleBL}$  determined by the old

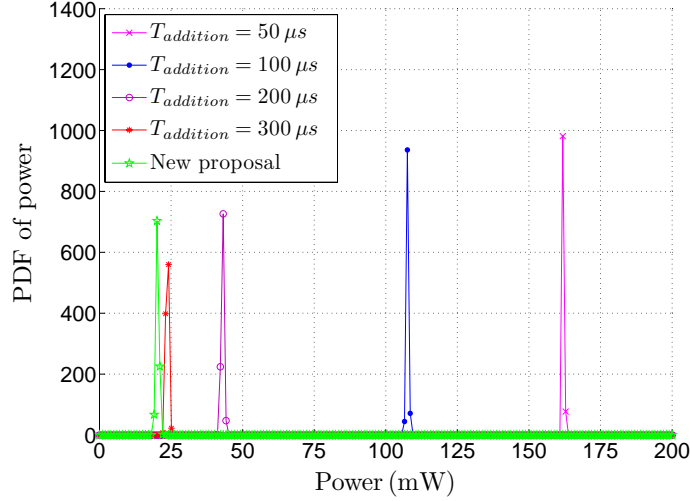


Figure 6.4: Comparison of the power PDFs.

algorithm approaches the average power obtained from the new mechanism. It can be explained easily by the monotonic decreasing characteristic of the complementary error function  $\text{erfc}(\cdot)$  and its fast convergence to zero. When  $T_{\text{addition}} > 300 \mu\text{s}$ , we have

$$\text{erfc}\left(\frac{\mu + T_{\text{addition}}}{\sqrt{2}\sigma}\right) \ll 0. \quad (6.8)$$

Therefore, the inequality

$$P_{\text{wake}} \cdot \frac{1}{2} \cdot \text{erfc}\left(\frac{\mu + T_{\text{addition}}}{\sqrt{2}\sigma}\right) \ll 0, \quad (6.9)$$

and the approximation

$$1 - \frac{1}{2} \cdot \text{erfc}\left(\frac{\mu + T_{\text{addition}}}{\sqrt{2}\sigma}\right) \approx 1, \quad (6.10)$$

hold. Finally, the power  $P_{\text{idleBL}}$  in Eq. 6.5 can be simplified as

$$P_{\text{idleBL}} \approx P_{\text{sleep}} + \frac{P_{\text{wake}} - P_{\text{sleep}}}{T_{\text{beaconpd}}} \cdot (T_{\text{addition}} + T_{\text{partial}} + T_{\text{drift}}). \quad (6.11)$$

We find that  $P_{\text{idleBL}} \rightarrow P_{\text{new}}$  if  $T_{\text{addition}} > 300 \mu\text{s}$ . Such a conclusion is also validated by the results presented in Fig. 6.4.

The purpose of this subsection is to confirm that the proposed algorithm outperforms the old mechanism from mathematic perspective. For this purpose, the theoretical equations for the corresponding power, denoted by  $P_{\text{old}}$  and  $P_{\text{new}}$ , respectively, are first presented. Taking the time drift  $T_{\text{drift}}$  is a Gaussian variable into account, the PDFs of the power  $P_{\text{old}}$  and  $P_{\text{new}}$  are illustrated in Fig. 6.4.

It can be seen in this figure that all PDF curves are symmetric and steep, which implies that all samples are highly concentrated. From this point of view, the symmetric point with a peak value can be regarded as the mean value of all the samples. The results presented in Fig. 6.4 show that for the old mechanism, the PDF of the power  $P_{old}$  shift to the left if the value of the time drift  $T_{drift}$  increases. That means the average power decreases as increasing  $T_{drift}$ . When  $T_{drift} > 300\mu s$ , the power PDFs for both algorithms are overlapped. Such an observation can be explained by the fact that the power  $P_{old}(P_{idleBL})$  approximates to  $P_{new}$  as  $T_{drift} > 300\mu s$  [see Eq. 6.8 - Eq. 6.11].

# Chapter 7

## Discussions

Chapter 7 discusses the problems mentioned in the subsection of problem statement.

### 7.1 Search and identify PCAFs

This master project thesis concentrates on the investigations how the PCAF affects the power consumption in 802.11 WLANs. Therefore, as a starting point, selecting and identifying proper PCAFs is of special importance. The PCAFs used in this thesis are chosen according to the 802.11 standard, relevant literature, the practical experience from Ericsson, and the experimental exploration. These PCAFs are selected in such a way so that the PCAF can be tested in the test-bed when they need to be identified. Moreover, the PCAFs must be adjustable using either hardware or software.

Several PCAFs have been taken into consideration in this thesis, known as the beacon period, the DTIM, the beacon size, the signal strength, the foreground traffic, and the background traffic. All these PCAFs mentioned above are identified for the purpose of the WLAN power consumption investigations. Specifically, both the beacon period and the DTIM are highly related with the legacy polling scheme as shown in the 802.11 standard [31], while the beacon size and the signal strength are suggested by Ericsson. In addition, the traffic PCAFs have been discussed in [14], and based on experimental exploration, the traffic PCAFs take a traffic direction, a data rate and a channel interval into account.

## 7.2 Experimental study

It is important to guarantee the validity and the accuracy of the collected data in experiments, since data plays an essential role to quantitatively analyze how PCAFs affect power conservation. To achieve this, a couple of points have been taken into account during the experiments.

- Since the test-bed is located in the working area of Ericsson, interferences from other existing 802.11 WLAN devices are inevitable, unpredictable and not under control, which can be monitored by the oscilloscope. To avoid interferences, the signals transmitted in the test-bed are shielded by RF cables, and most of the testing are carried on during evenings.
- Each measurement result is the average value based ten independent measurements for each testing instruction.
- During the traffic mode measurements, the result is regarded as invalid if a collision between a foreground traffic and a background traffic occurs.

Due to the time limitation, this thesis only considers the case when the channel interval ( $I_{channel}$ ) is no more than 3. It would be desirable to test the case when  $I_{channel} > 3$ , although it can be predicted, according to the theoretical knowledge and the obtained measurement results, that the case  $I_{channel} \leq 3$  consumes more power than that of the case when  $I_{channel} > 3$ . Moreover,  $I_{channel}=1$  costs the most power consumption among all kinds of background traffic. The interesting phenomenon means a transmission in another WLAN with  $I_{channel}=1$  leads to the most power consumption in the WLAN under test. The reason for this is left as one of the further works.

In addition, during the traffic mode measurements, where both the foreground traffic and the background traffic are involved, collisions between these traffics sometimes occur. The collision mainly relies on the hardware under test. Fig.7.1 displays the statistic information related to the collision appearance during the experiment. The downlink foreground traffic is very stable without any collision with others, while the uplink foreground traffic performs worse. Since the measurement result in face of a collision is classified as an invalid data, the information presented in the figures in Chapter 3 is less than what the test instructions specified.

In Chapter 4, analytical expressions are derived to calculate the power consumption of 802.11 WLANs (for both the idle mode and the traffic mode). The experimental data are also employed to validate these obtained equations. A good correspondence between the analytical results and the experimental results can be observed. From now on, the

AP	Channel Interval	FG		Downlink	Uplink
		BG			
Same	0	Downlink		✓	Collision
		Uplink		✓	✓
Downlink			✓	Collision	
Uplink			✓	✓	
Different	1	Downlink		✓	Collision
		Uplink		✓	Collision
	2	Downlink		✓	Collision
		Uplink		✓	Collision
	3	Downlink		✓	Collision
		Uplink		✓	Collision

Figure 7.1: Collision statistics in the traffic mode.

analytical derivations allows to theoretically predict the power consumption for different scenarios specified in 802.11 WLANs without involving any real testing.

### 7.3 WPVT

The development method for WPVT is model driven development. It is easy to transform from the class diagram to the java code. But a lot of details have to be taken into account when specifying the transformation configurations. If one of them is neglected or set up incorrectly, the transformation will fail.

Since the effect of background traffic is based on the experimental results, so there is a lack of generality to a certain extent. The experimental results regarding background traffic is based on default configuration of the WLAN under test, but the parameters' manual set up may not be classified into default configuration, which is one of the further tasks discussed afterwards.

### 7.4 New proposal

After analyzing the legacy polling scheme, which is also known as the traditional power save mechanism for 802.11 WLANs, a new proposal is derived based on the experiment results. The difference between the legacy polling scheme and the new proposal is that the new algorithm takes the time drift into account.

The power consumption mainly depends on beacon lost probability ( $P_{BL}$ ), which can be

indirectly affected by all the PCAFs investigated before. Moreover,  $P_{BL}$  depends on the time drift of SYN clock between an AP and a STA. After analyzing the air log by making use of Excel and Matlab, it can be concluded that the time drift follows a Gaussian distribution. Assuming the time drift is a Gaussian variable, the PDF curves of the power consumption corresponding to the Legacy polling scheme and the new proposal are plotted. It can be seen that the new proposal provides a lower power consumption compared with the legacy polling scheme.

In order to confirm the validity of the new proposal by simulations, it would be advisable if the new mechanism can be simulated with a professional network simulation tool, e.g., NS2.



## Chapter 8

# Conclusions and Future Work

### 8.1 Conclusion

In this master thesis, the focus has been on the 802.11 WLAN power consumption analysis. The work can be summarized as follows.

The experimental study of the impacts of different PCAFs on the power consumption of 802.11 WLANs is presented. The identified and confirmed PCAFs include the beacon period, the DTIM, the beacon size, the signal strength, the foreground traffic, and the background traffic ( $BG_\alpha$ ,  $BG_\beta$ ,  $BG_{\gamma 1}$ ,  $BG_{\gamma 2}$ ,  $BG_{\gamma 3}$ ). Following the test instructions specified for both the idle mode and the traffic mode, measurement data for these two modes have been collected from the test-bed. According to the experimental study, it has been shown that all the PCAFs have different qualitative and different quantitative effect on the power consumption of 802.11 WLANs.

- The power consumption exhibits a negative proportional relation to both the beacon period and the DTIM, while it displays a positive proportional relation to both the beacon size and the signal strength.
- The traffic PCAFs consume more power compared with the non-traffic factors. The reason is that the presence of the traffic factors can significantly reduce the effectiveness of a power saving mechanism. The foreground traffic is involved in a transmission or a reception, which is supposed to consume more energy. However, the background traffic is a passive PCAF, which means the under test WLAN is affected by an activity from another WLAN.
- A downlink traffic indicates more power consumption than a uplink traffic, due to the coordination by the AP. A higher data rate of a background traffic leads to a more congested medium, which further causes a more severe time drift. Therefore,

more power is consumed. On the other hand, a higher data rate of a foreground traffic indicates less time cost for a transmission or a reception. In a sense, less power is consumed.

- In an idle mode, the effects of background traffic factors on the power management sorted in a descending order are  $BG_{\gamma_1}$ ,  $BG_{\beta}$ ,  $BG_{\gamma_2}$ , and  $BG_{\gamma_3}$ , while in a traffic mode, the descending order changes to  $BG_{\gamma_1}$ ,  $BG_{\gamma_2}$ ,  $BG_{\beta}$  and  $BG_{\gamma_3}$ . However, the effect caused by  $BG_{\alpha}$  is not stable confronted with background traffics with different directions, or different data rates compared with others. In addition to the observations, the quantitative relationship among PCAFs and the power consumption are illustrated in Chapter 3.

Mathematical formulas for the 802.11 WLN power consumption have been derived for both the idle mode and the traffic mode, respectively. Furthermore, accuracy of these analytical results have been verified by the experimental data measured from the test-bed. The average difference between the theoretical calculation and the measurement is 2.83 mW for an idle mode. The maximum deviations are 5.02 mW for a uplink traffic and 8.72 mW for a downlink traffic, i.e., 8.66% and 6.28% of the measurement results, which can be tolerated in most practical applications. Thus, the formula can be referenced with an acceptable precisions.

A visualization tool, known as WPVT, has been implemented by Java. The basic functionality of WPVT is that power consumption curves can be displayed with given PCAF configurations. All the theoretical curves shown in WPVT are plotted according to the analytical formula, the accuracy of which have been demonstrated by experimental results. The advanced feature of the tool is that the background traffic can be taken into account during illustrations.

A new algorithm, by taking a time drift of SYN clock into account, has been proposed for the purpose of improving a power conservation. An analytical expression for the power consumption has been obtained for the new mechanism. Also, by analyzing the airlog captured by Wireshark, it can be concluded that a time drift is normal distributed. Furthermore, considering the time drift as a Gaussian variable, PDF curves of the power consumption have been presented for both the Legacy polling scheme and the new mechanism. It turns out that the new mechanism outperforms the Legacy polling scheme with respect to the power consumption.

## 8.2 Contributions

The major contributions of this thesis can be summarized as follows.

- According to the experimental study, it turns out that the background traffic is the most powerful PCAF negatively affecting the power conservation. And it depends on three parameters, known as the direction, the data rate, and the channel interval. To the best of author's knowledge, the power consumption investigation incorporating these three parameters is less known. The investigation of the background traffic in terms of these parameters can absolutely supplement the existing research.
- The formulas for the power consumption have been derived for both the idle mode and the traffic mode, which provides a more formal form to describe the relationship between PCAFs and the power consumption. Moreover, they have been validated by the experimental results, so the accuracy can be guaranteed. In addition, a beacon lost probability, described by  $P_{BL}$ , is a new finding in the thesis, which is the most direct factor affecting the power consumption. Actually, all the PCAFs firstly affect  $P_{BL}$ , which accordingly influences the power consumption.
- The new algorithm has been proposed in the thesis, which depends on the time drift of the SYN clock between an AP and a STA. Based on mathematical demonstrations, it is shown that the new proposal provides an optimization solution compared with the 802.11 legacy polling scheme. It contributes the most in the thesis due to the novelty of the methodology and the conclusion. In numerous publications, algorithms proposed for the power conservation are validated by network simulation tools such as NS2. This thesis achieves validations using mathematics, which provides a new method to demonstrate proposed algorithms.
- WPVT has been implemented as a supplement to achieve power consumption illustration for 802.11 WLANs. Compared with the experimental study, it provides a much less time consuming method to investigate PCAFs against the power conservation. What WPVT requires is only a configuration of PCAFs instead of constructing a test-bed, configuring hardware, testing with multimeter, etc., included in the traditional method.

## 8.3 Further Work

The experimental study has uncovered the effect of each PCAF over power conservation in 802.11 and many meaningful observations have been obtained. But there are a couple of points, required to be taken into account, for the further work.

- Both  $AP_{DUT}$  and  $AP_{REF}$  are supposed to be changed with different styles or brands in order to get more persuasive observations.
- Although the shielded cable can get rid of interferences from other WLANs, it provides a totally different fading medium from the air. If the condition allows, it is better to place the test bed in an anechoic chamber to simulate the more real WLAN environment.
- It is advisable to do measurements with  $BG_{\gamma n}$ , where  $3 < n < 12$ , since the interference from different channels can be thoroughly analyzed based on the experiments. From this point of view, the background traffic illustration in WPVT can also be supplemented.
- It is of a special interest that the background traffic eliminates the effectiveness of power management the most when the channel interval equals 1. But the reason for this strange phenomenon is not known. It is advisable to look into this issue with the theoretical knowledge in the physical layer.
- The limitation of WPVT, discussed before, is that the effect of a background traffic depends on the measurement results for a default configuration, so there is a lack of generality. In order to supplement the current solution, more measurements are required for a background traffic, where different configurations of the test-bed have to be taken into account.
- It is preferable for WPVT to support the illustration for a hybrid traffic mode, which mixes an idle state and a transmitting/receiving state. The frequency, the duration, and the appearance order of these states can be specified in advance by a user. The implementation of the hybrid traffic mode can be implemented by extending the parent class, *WlanCalculation*.
- The improvement of the proposed algorithm has been demonstrated by the mathematical deductions. Although it provides a novel way to analyze the new algorithm, it has to rely on a specific assumption, that the algorithm perfectly works to exactly adapt to the time drift. In order to get more persuasive simulation results, it is suggested to simulate the new algorithm in a simulation tool, such as NS2, which can reflect a more real networking environment.

## Appendix A

# Agilent Script

```
var CLSID_AgtIOManager = "AgtIOUtils.AgtIOManager";  
// OLE Class ID of Agilent measurements library  
  
var myInstrument="Multimeter";           // Name of measurement device  
  
var AgtIOManager = new ActiveXObject(CLSID_AgtIOManager);  
// An instance of agilent measurements library is created  
  
var device = AgtIOManager.ConnectToInstrument(myInstrument);  
// An instrument (multimeter) is selected from the library  
  
var reading = 0;           // A single measurement: current consumption of the chip  
  
var max_cnt = 15 * 60 * 5;  
// Number of measurements. Measurements rate is 15 samples per second.  
  
var threshold = 0.005;     // If current is higher than threshold, chip is awake.  
  
var i;                     // Measurement sample counter  
var nawk = 0, nslp = 0;    // Counters for "chip is awake" and "chip is in sleep".  
  
device.Output("CONFIGURE:VOLTage:DC 1,.0001"); // Starting measurements
```

## APPENDIX A. AGILENT SCRIPT

---

```
for(i = 0; i < max_cnt; ++i)
{
reading = new Number(device.Query("MEASURE?")); // A single sample
if(reading > threshold) // Check if reading is above threshold or not
{
nawk ++; // Increment "awake" counter
}
else
{
nslp ++; // Increment "sleep" counter
}
}
var rateawk = nawk / max_cnt * 100; Calculate awake rate
WScript.Echo("Chip is awake: " + rateawk + "% of time."); // And print it.
```

## Appendix B

# Cisco AP Configuration File

```
!  
version 12.4  
no service pad  
service timestamps debug datetime msec  
service timestamps log datetime msec  
service password-encryption  
!  
hostname AP-1-1  
!  
enable secret 5 $1$.fnS$FS2xRDEfSyqLhkx2irVvp1  
!  
no aaa new-model  
!  
!  
dot11 mbssid dot11 vlan-name UNSECURE1 vlan 101  
dot11 vlan-name UNSECURE2 vlan 102  
dot11 vlan-name UNSECURE3 vlan 103  
dot11 vlan-name UNSECURE4 vlan 104  
dot11 vlan-name UNSECURE5 vlan 105  
dot11 vlan-name UNSECURE6 vlan 106  
dot11 vlan-name UNSECURE7 vlan 107  
dot11 vlan-name UNSECURE8 vlan 108  
!  
dot11 ssid LiSSID
```

## APPENDIX B. CISCO AP CONFIGURATION FILE

---

```
vlan 101
authentication open
mbssid guest-mode
!
!
!
username Cisco password 7 106D000A0618
username admin privilege 15 password 7 073C385F780C0B
!
bridge irb
!
!
interface Dot11Radio0
no ip address
no ip route-cache
!
ssid LiSSID
!
antenna transmit right
antenna receive right
power local -1
channel 2412
station-role root
no dot11 qos mode
bridge-group 1
bridge-group 1 block-unknown-source
no bridge-group 1 source-learning
no bridge-group 1 unicast-flooding
bridge-group 1 spanning-disabled
!
interface Dot11Radio0.101
encapsulation dot1Q 101
no ip route-cache
bridge-group 101
bridge-group 101 subscriber-loop-control
bridge-group 101 block-unknown-source
no bridge-group 101 source-learning
```



```
no bridge-group 101 unicast-flooding
bridge-group 101 spanning-disabled
!
interface Dot11Radio0.102
encapsulation dot1Q 102
no ip route-cache
bridge-group 102
bridge-group 102 subscriber-loop-control
bridge-group 102 block-unknown-source
no bridge-group 102 source-learning
no bridge-group 102 unicast-flooding
bridge-group 102 spanning-disabled
!
interface Dot11Radio0.103
encapsulation dot1Q 103
no ip route-cache
bridge-group 103
bridge-group 103 subscriber-loop-control
bridge-group 103 block-unknown-source
no bridge-group 103 source-learning
no bridge-group 103 unicast-flooding
bridge-group 103 spanning-disabled
!
interface Dot11Radio0.104
encapsulation dot1Q 104
no ip route-cache
bridge-group 104
bridge-group 104 subscriber-loop-control
bridge-group 104 block-unknown-source
no bridge-group 104 source-learning
no bridge-group 104 unicast-flooding
bridge-group 104 spanning-disabled
!
interface Dot11Radio0.105
encapsulation dot1Q 105
no ip route-cache
bridge-group 105
```

```
bridge-group 105 subscriber-loop-control
bridge-group 105 block-unknown-source
no bridge-group 105 source-learning
no bridge-group 105 unicast-flooding
bridge-group 105 spanning-disabled
!
interface Dot11Radio0.106
encapsulation dot1Q 106
no ip route-cache
bridge-group 106
bridge-group 106 subscriber-loop-control
bridge-group 106 block-unknown-source
no bridge-group 106 source-learning
no bridge-group 106 unicast-flooding
bridge-group 106 spanning-disabled
!
interface Dot11Radio0.107
encapsulation dot1Q 107
no ip route-cache
bridge-group 107
bridge-group 107 subscriber-loop-control
bridge-group 107 block-unknown-source
no bridge-group 107 source-learning
no bridge-group 107 unicast-flooding
bridge-group 107 spanning-disabled
!
interface Dot11Radio0.108
encapsulation dot1Q 108
no ip route-cache
bridge-group 108
bridge-group 108 subscriber-loop-control
bridge-group 108 block-unknown-source
no bridge-group 108 source-learning
no bridge-group 108 unicast-flooding
bridge-group 108 spanning-disabled
!
interface Dot11Radio1
```

```
no ip address
no ip route-cache
shutdown
no dfs band block
channel dfs
station-role root
no dot11 qos mode
bridge-group 1
bridge-group 1 subscriber-loop-control
bridge-group 1 block-unknown-source
no bridge-group 1 source-learning
no bridge-group 1 unicast-flooding
bridge-group 1 spanning-disabled
!
interface FastEthernet0
no ip address
no ip route-cache
duplex auto
speed auto
bridge-group 1
no bridge-group 1 source-learning
bridge-group 1 spanning-disabled
!
interface FastEthernet0.101
encapsulation dot1Q 101
no ip route-cache
bridge-group 101
no bridge-group 101 source-learning
bridge-group 101 spanning-disabled
!
interface FastEthernet0.102
encapsulation dot1Q 102
no ip route-cache
bridge-group 102
no bridge-group 102 source-learning
bridge-group 102 spanning-disabled
!
```

```
interface FastEthernet0.103
encapsulation dot1Q 103
no ip route-cache
bridge-group 103
no bridge-group 103 source-learning
bridge-group 103 spanning-disabled
!
interface FastEthernet0.104
encapsulation dot1Q 104
no ip route-cache
bridge-group 104
no bridge-group 104 source-learning
bridge-group 104 spanning-disabled
!
interface FastEthernet0.105
encapsulation dot1Q 105
no ip route-cache
bridge-group 105
no bridge-group 105 source-learning
bridge-group 105 spanning-disabled
!
interface FastEthernet0.106
encapsulation dot1Q 106
no ip route-cache
bridge-group 106
no bridge-group 106 source-learning
bridge-group 106 spanning-disabled
!
interface FastEthernet0.107
encapsulation dot1Q 107
no ip route-cache
bridge-group 107
no bridge-group 107 source-learning
bridge-group 107 spanning-disabled
!
interface FastEthernet0.108
encapsulation dot1Q 108
```

## APPENDIX B. CISCO AP CONFIGURATION FILE

---

```
no ip route-cache
bridge-group 108
no bridge-group 108 source-learning
bridge-group 108 spanning-disabled
!
interface BV11
ip address dhcp client-id FastEthernet0
no ip route-cache
!
ip http server
no ip http secure-server
ip http help-path http://www.cisco.com/warp/public/779/smbiz/prodconfig/help/eag
bridge 1 route ip
!
!
!
line con 0
line vty 0 4
login local
!
end
```

# Bibliography

- [1] M. S. Gast, *802.11 Wireless Networks: The Definition Guide*. O'REILLY, 2005.
- [2] Model-view-controller pattern. MARKUP LANGUAGE. [Online]. Available: <http://www.enode.com/x/markup/tutorial/mvc.html>
- [3] Wi-fi alliance home page. Wi-Fi Alliance. [Online]. Available: <http://www.wi-fi.org/>
- [4] Ibm rational software architect. IBM. [Online]. Available: <http://www-01.ibm.com/software/awdtools/architect/swarchitect/>
- [5] Ieee 802.11. Wikipedia. [Online]. Available: <http://en.wikipedia.org/wiki/802.11>
- [6] *IEEE Std 802.11a: High-speed Physical Layer in the 5 GHz Band*, IEEE Computer Society LAN MAN Standards Committee, Std., 1999.
- [7] *IEEE Std 802.11b: Higher-Speed Physical Layer Extension in the 2.4 GHz Band*, IEEE Computer Society LAN MAN Standards Committee, Std., 1999.
- [8] *IEEE Std 802.11g: Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band*, IEEE Computer Society LAN MAN Standards Committee, Std., 2003.
- [9] *IEEE P802.11n/D7.0: Amendment 5: Enhancements for Higher Throughput*, IEEE Computer Society LAN MAN Standards Committee, Std., 2008.
- [10] *IEEE Std 802.11e: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements*, IEEE Computer Society LAN MAN Standards Committee, Std., 2005.
- [11] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. IEEE INFOCOM*, 2001.

## BIBLIOGRAPHY

---

- [12] A. Mahersi and V. Vardham, “Power consumption breakdown on modern laptop,” in *Proc. the 4th workshop on Power-Aware Computer Systems*, 2004.
- [13] Atheros\_power\_phitepaper. Atheros. [Online]. Available: [http://www.atheros.com/pt/whitepapers/atheros\\_power\\_whitepaper.pdf](http://www.atheros.com/pt/whitepapers/atheros_power_whitepaper.pdf)
- [14] Y. He, R. Yuan, X. Ma, J. Li, and C. Wang, “The ieee 802.11 power saving mechanism: An experimental study,” in *Proc. IEEE WCNC.*, 2008.
- [15] Wlan traffic visualizer. SourceForge org. [Online]. Available: <http://sourceforge.net/projects/wlantv/>
- [16] Wireshark go deep. The Wireshark team. [Online]. Available: <http://www.wireshark.org/>
- [17] B. Chen, K. Jamieson, H. Balakrishnan, and J. L. and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” in *Proc. MOBICOM*. Roma, Italy, Jul. 2001, pp. 85–96.
- [18] Y. Xu, J. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Proc. MOBICOM*. Roma, Italy, Jul. 2001, pp. 70–84.
- [19] Y. He, R. Yuan, X. Ma, J. Li, and C. Wang, “Scheduled psm for minimizing energy in wireless lans,” in *Proc. ICNP.*, 2007.
- [20] E. S. Jung and N. H. Vaidya, “An energy efficient mac protocol for wireless lans.” in *Proc. INFOCOM*. New York, NY, USA, Jun. 2001, pp. 1756–1764.
- [21] Cisco iso software configuration guide for cisco aironet access point. Cisco Systems, Inc. [Online]. Available: [http://www.cisco.com/en/US/docs/wireless/access\\_point/12.2.15\\_JA/configuration/guide/s15local.html](http://www.cisco.com/en/US/docs/wireless/access_point/12.2.15_JA/configuration/guide/s15local.html)
- [22] U.s.robotics wireless maxg adsl gateway user guide. U.S. Robotics Corporation. [Online]. Available: <http://www.usr.com/support/9108/9108-ug/index.htm>
- [23] Python programming language. Python Software Foundation. [Online]. Available: <http://www.python.org/>
- [24] M. Fowler and K. Scott, *UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition*. Addison-Wesley Professional, 2003.
- [25] D.Milicev, *Model-Driven Development with Executable UML*. Wrox, 2009.

## BIBLIOGRAPHY

---

- [26] Jfreechart. Object Refinery Limited. [Online]. Available: <http://www.jfree.org/jfreechart/>
- [27] Jar file specification. SUN Microsystems. [Online]. Available: <http://java.sun.com/j2se/1.3/docs/guide/jar/jar.html>
- [28] Fat jar eclipse plug-in. feri. [Online]. Available: <http://fjep.sourceforge.net/>
- [29] H. Marouani and M. R. Dagenais, "Internal clock drift estimation in computer clusters," Tech. Rep. 5, 2008.
- [30] How to use csv files. International Monetary Fund. [Online]. Available: <http://www.imf.org/external/help/csv.htm>
- [31] *IEEE Std 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Computer Society LAN MAN Standards Committee, Std., 1999.