Non-collaborative coexistence mechanism for WPAN in unlicensed spectrum

Department of Communication Technology
Aalborg University

June 3rd, 2004

Mobile Communications Group 1097, the 10th Semester
Abstract

The unlicensed spectrum, which is available at no cost and does not require lengthy and cumbersome process of getting license, stimulates the growth of various wireless applications, including WPAN and WLAN. Such growth tends to pollute the unlicensed spectrum with high level of interference. Therefore, it is essential for the wireless applications to employ certain coexistence mechanisms to enable sharing of the unlicensed spectrum. Such mechanisms are intended to make them more robust against interference and, at the same time, to avoid causing interference to other systems operating in the same band.

This project focuses on the non-collaborative coexistence mechanisms, which do not require interfering entities to explicitly exchange information to achieve mutual coordination, for WPAN in the unlicensed spectrum. Here we consider the Bluetooth wireless technology as a particular instance of WPAN, with two major sources of interference, namely frequency-static interference and self-interference. We analyse three different non-collaborative coexistence techniques, termed AFH, DAFH and FR, and propose necessary modifications for them to work with both types of interference. A C++ discrete event Bluetooth simulator is developed to investigate the performance of our proposals and impacts of their parameters. The DAFH and FR mechanisms exhibit excellent performance, in terms of average goodput, for a practically large value for the number of collocated piconets.

Participant: Huan Cong Nguyen

Supervisors: Hiroyuki Yomo, Petar Popovski

Publications: 8

Number of pages: 97

Finished: June 3rd, 2004
Preface

This Master thesis is written by Huan Cong Nguyen at the Department of Communication Technology, Institute of Electronic Systems, Aalborg University. It is the result of the 10th semester project, conducted from January to June 2004.

Report Structure

This thesis intends to take the reader through the problem analysis phase to the implementation and performance evaluation of proposed solutions. Its structure, therefore, is divided into 5 parts, as follows:

- Chapter 1: Introduction
- Chapter 2: Bluetooth wireless technology
- Chapter 3: Coexistence mechanisms
- Chapter 4: Simulation results
- Chapter 5: Conclusions and future works

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Hiroyuki Yomo and Petar Popovski, for their thorough assistance and guidance during this project. And special thanks to my wife and daughter, Lan My Tran and Huong Lan Nguyen, for their love, encouragement and support.

Huan Cong Nguyen
Contents

1 Introduction

1.1 The unlicensed spectrum and coexistence problem .......................... 1
1.2 Wireless Personal Area Network ................................................. 3
   1.2.1 WPAN characteristics ................................................. 4
   1.2.2 WPAN standards .................................................. 5
1.3 Scope of project ....................................................................... 7
   1.3.1 Problem identification .............................................. 7
   1.3.2 Objectives of the project ........................................... 9
1.4 Summaries ............................................................................. 10

2 Bluetooth wireless technology ...................................................... 11

2.1 Radio Frequency Layer ........................................................... 11
   2.1.1 Frequency Hopping .................................................... 12
   2.1.2 Modulation ............................................................. 12
   2.1.3 Power emission and control ....................................... 13
2.2 Baseband Layer ........................................................................ 14
   2.2.1 Bluetooth Device Address ............................................. 14
   2.2.2 Master, slave, piconet and scatternet .......................... 14
   2.2.3 Physical link between master and slave ...................... 15
   2.2.4 Selection of frequency hopping pattern ...................... 18
   2.2.5 Bluetooth packet structure ........................................ 19
   2.2.6 Exchange of control information ................................. 22
2.3 Probability of packet collision for Bluetooth piconets ................. 23
   2.3.1 Collision induced by self-interference ............................ 24
   2.3.2 Self-interference and frequency division strategy .......... 26
   2.3.3 Collision induced by frequency-static interference .......... 26
   2.3.4 Frequency-static interference model ............................ 28
2.4 Summaries ............................................................................. 28
# CONTENTS

## 3 Coexistence mechanisms

3.1 Adaptive Frequency Hopping (AFH) .................................................. 31
   3.1.1 Overview of AFH .......................................................... 31
   3.1.2 Implementation of the reference AFH mechanism ....................... 34

3.2 Dynamic Adaptive Frequency Hopping (DAFH) ............................... 35
   3.2.1 Customisation of DAFH superset ....................................... 36
   3.2.2 Operation of DAFH mechanism ......................................... 40
   3.2.3 Variants of DAFH mechanism .......................................... 41
   3.2.4 Advantages and disadvantages of DAFH ............................... 42

3.3 Frequency Rolling (FR) .............................................................. 43
   3.3.1 Operation of FR mechanism .............................................. 43
   3.3.2 Frequency Rolling with Probing (FR-WP) .............................. 45
   3.3.3 Frequency Rolling with Frequency Group Removal (FR-FGR) .......... 47

3.4 Summaries .................................................................................... 48

## 4 Simulation results

4.1 Simulation model ......................................................................... 51
   4.1.1 Bluetooth simulator ......................................................... 51
   4.1.2 Simulation scenario ......................................................... 52
   4.1.3 Simulation parameters for coexistence mechanisms ................... 54

4.2 Result analysis ............................................................................ 55
   4.2.1 Performance of the AFH mechanism ...................................... 57
   4.2.2 Analyses of DAFH simulation results ..................................... 60
   4.2.3 Analyses of FR simulation results ......................................... 68
   4.2.4 Comparison of coexistence mechanisms ................................. 75

4.3 Summaries .................................................................................... 76

## 5 Conclusions and future works

5.1 Conclusions .................................................................................. 79
   5.1.1 The AFH mechanism ......................................................... 80
   5.1.2 The DAFH mechanism ....................................................... 80
   5.1.3 The FR mechanism ............................................................ 81

5.2 Future works ................................................................................ 82

Bibliography......................................................................................... 83

A List of acronyms .............................................................................. 85

B List of symbols ............................................................................... 89

C Design of FCC-compliant FR-WP algorithm ................................. 93
   C.1 Fast Frequency Rolling with Probing (FFR-WP) algorithm .......... 93
   C.2 Performance of FFR-WP mechanism ....................................... 95
   C.3 Conclusions .............................................................................. 97
# List of Figures

1.1 WPAN in comparison with other types of wireless networks


2.1 Bluetooth ad-hoc network: (a) Point-to-point piconet, (b) Point-to-multipoint piconet, and (c) Scatternet [5] ............................... 15

2.2 piconet in operation with ACL links [5] ............................... 16

2.3 piconet in operation with a SCO link [5] ............................... 17

2.4 The Bluetooth basic frequency hop selection kernel ............................... 18

2.5 Slot timing for multi-slot packets [2] ............................... 19

2.6 Bluetooth packet: (a) Generic packet structure, (b) ACL Payload, and (c) SCO Payload [2] ............................... 20

2.7 Example of packet overlapping in the case of asynchronous piconets ............................... 24

2.8 Dependency of collision probability on number of frequency groups ............................... 27

2.9 Bluetooth channel allocation in comparison with the IEEE 802.11b DSSS spectrum ............................... 28

3.1 The Bluetooth adaptive frequency hop selection kernel ............................... 33

3.2 The original DAFH superset ............................... 36

3.3 The customised DAFH superset for 802.11b non-overlapping channel allocation ............................... 38

3.4 The rolling and random jump mechanisms in FR ............................... 44

3.5 The state-machine diagram for the FR-WP mechanism ............................... 46

3.6 The block interleaving scheme with $N_G = 4$ for FR-WP mechanism ............................... 47

4.1 States and events for a Bluetooth piconet ............................... 51

4.2 State-machine diagram for Bluetooth simulator ............................... 52

4.3 Normalised average goodput for AFH mechanism (without and with FSI) ............................... 58

4.4 Collision probability for AFH mechanism (without and with FSI) ............................... 58

4.5 Frequency occupancy for AFH mechanism (without and with FSI) ............................... 59

4.6 Normalised average goodput for DAFH mechanism (without FSI) ............................... 61

4.7 Collision probability for DAFH mechanism (without FSI) ............................... 61

4.8 Frequency occupancy for DAFH mechanism (without FSI) ............................... 62

4.9 Normalised average throughput for DAFH mechanism (with FSI) ............................... 63

4.10 Collision probability for DAFH mechanism (with FSI) ............................... 64

4.11 Frequency occupancy for DAFH mechanism (with FSI) ............................... 64

4.12 Effect of parameter $\alpha$ on DAFH-AT (Normalised average throughput, with FSI) ............................... 65
4.13 Effect of packet corruption probability (Normalised average throughput, with FSI) ................................................. 66
4.14 Effect of packet corruption probability (Probability of collision, with FSI) ......................................................... 66
4.15 Effect of DAFH-AT triggering threshold (Normalised average throughput, with FSI) ......................................................... 68
4.16 Normalised average goodput for FR mechanisms (without FSI) .......................................................... 69
4.17 Collision probability for FR mechanisms (without FSI) .......................................................... 69
4.18 Normalised average goodput for FR mechanisms (with FSI) .......................................................... 70
4.19 Collision probability for FR mechanisms (with FSI) .......................................................... 71
4.20 The effect of $J_c$ of FR-WP mechanism (Normalised average goodput, with FSI) ......................................................... 72
4.21 The effect of $P_{j}^{FR-WP}$ of FR-WP mechanism (Normalised average goodput, with FSI) ......................................................... 73
4.22 The effect of $H$ of FR-WP mechanism (Normalised average goodput, with FSI) ......................................................... 73
4.23 The effect of $T_p^{FR-WP}$ of FR-WP mechanism (Normalised average goodput, with FSI) ......................................................... 74
4.24 Normalised average goodput of coexistence mechanisms (without FSI) ......................................................... 75
4.25 Normalised average goodput of coexistence mechanisms (with FSI) ......................................................... 76

C.1 Fast frequency rolling mechanism ......................................................... 94
C.2 FFR-WP average goodput, with and without FSI ......................................................... 96
C.3 FFR-WP probability of collision, with and without FSI ......................................................... 97
List of Tables

1.1 Selected unlicensed frequency bands in the world [16] ............... 2

2.1 Transmitting power classes [2] ............................................ 13
2.2 Different types of Bluetooth packet [2] .................................. 21

3.1 A original superset of DAFH mechanism ............................... 37
3.2 A proposed superset for DAFH mechanism ............................. 39

4.1 Summaries of simulation parameters ...................................... 56
Chapter

Introduction

1.1 The unlicensed spectrum and coexistence problem

When Guglielmo Marconi formed The Wireless Telegraph & Signal Company Limited in July 1897 and became the first to commercialise wireless service, he could hardly imagine that the Radio Frequency (RF) spectrum, which was freely available to his company at that time, would quickly turn into a scarce and valuable public resource. Following Marconi’s wireless telegraph, various wireless services, from radio and television broadcast, paging and cellular telephony to satellite communication, have been introduced and popularised, which create a huge demand for the radio spectrum.

To ensure orderly access and maximise efficiency in its usage, the RF spectrum has been put under control of governments. Although the spectrum itself is infinite, not all of it is equally useful, depending on the radio propagation properties and the cost of technologies required. Therefore, frequencies above 300 GHz presently are not regulated, while frequencies below 300 GHz fall into two regulation categories, licensed and unlicensed [16]. To coordinate users and avoid potential interference, government regulators throughout the world require users to obtain license for the use of most frequencies. Such license gives its recipient the exclusive right to transmit in a block of spectrum in a given geographic region. While licensed service has a great advantage of protection from interference, the cost and administrative difficulties of obtaining a license for each transmitter are often prohibitive.

In addition to the licensed spectrum, government regulators have also set up several unlicensed frequencies, in which no individual license is granted [17]. Any device is allowed to transmit in unlicensed spectrum, provided that certain restrictions are met. Table 1.1 lists some of the most important unlicensed bands in the world and their properties. The table shows a general property that holds true for both licensed and unlicensed frequencies: The higher the frequency, the more bandwidth available for the service.

Recently, the unlicensed spectrum becomes more and more attractive due to several reasons. First of all, the unlicensed band is available at no cost and does not require the potentially lengthy and troublesome process of getting licenses, thereby stimulate its usages. This is in fact the original goal of having the unlicensed spectrum [14]. Secondly,
Table 1.1: Selected unlicensed frequency bands in the world

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Bandwidth</th>
<th>Geographic Availability</th>
<th>Power Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>900MHz</td>
<td>26MHz</td>
<td>North America</td>
<td>Up to 1W</td>
</tr>
<tr>
<td>2.4GHz</td>
<td>84MHz</td>
<td>United States</td>
<td>Up to 1W</td>
</tr>
<tr>
<td>2.4GHz</td>
<td>84MHz</td>
<td>Europe</td>
<td>Up to 100mW</td>
</tr>
<tr>
<td>2.4GHz</td>
<td>26MHz</td>
<td>Japan</td>
<td>Up to 10mW</td>
</tr>
<tr>
<td>5.2GHz</td>
<td>200MHz</td>
<td>Available in U.S. and</td>
<td>50 or 250mW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Europe for HIPERLAN only</td>
<td></td>
</tr>
<tr>
<td>5.8GHz</td>
<td>125MHz</td>
<td>North America</td>
<td>Up to 1W</td>
</tr>
<tr>
<td>24GHz</td>
<td>259MHz</td>
<td>North America</td>
<td>Up to 25W</td>
</tr>
<tr>
<td>60GHz</td>
<td>5GHz</td>
<td>North America and</td>
<td>500mW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hopefully Europe and</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Japan soon</td>
<td></td>
</tr>
<tr>
<td>&gt; 300GHz</td>
<td>Theoretically infinite but for practical use &lt; 50MHz</td>
<td>Limited by eye safety rules</td>
<td></td>
</tr>
</tbody>
</table>

an important advantage of the unlicensed spectrum is that it allows spectrum-sharing. For example, with the traditional licensed spectrum, spectrum sits idle at times when the license holder is not transmitting, which clearly is not efficient. With the unlicensed spectrum, many devices can share the same spectrum, so that when one is not transmitting, the other can. Such sharing greatly increases the potential for spectral efficiency, i.e. more traffic can be carried when the spectrum is shared [17]. Thirdly, many mobile applications, such as Wireless Personal Area Network (WPAN) and Wireless Local Area Network (WLAN), are ill-suited for licensed spectrum. Consider two or more laptops exchanging data via their own portable wireless network. With the unlicensed spectrum, users need not to obtain permission to operate such a network from any location where they might ever want to be. This is often considered as the key reason for using the unlicensed spectrum.

However, the usage of the unlicensed spectrum does come with a price. Because it is freely available, there is no limit on the number of users sharing a spectrum in a given location. Furthermore, any time a resource is shared by many users, no single user has strong incentive to conserve the resource. As such, there is no guarantee that performance under the license-free spectrum will be adequate. The interference level might be very strong and varying unexpectedly, especially in the outdoor environment, making correct reception difficult. This represents the most challenging problem for wireless system operating under the unlicensed band.

To address the problem, government regulators have imposed certain restrictions on the power level and access protocol for wireless devices to coexist in the unlicensed
1.2 Wireless Personal Area Network

band. The restrictions are different for different parts of the spectrum and for different geographical regions. For instance, a set of regulations for wireless devices operating at the 900MHz, 2.4GHz Industrial, Scientific and Medical (ISM) and 5.2GHz Unlicensed - National Information Infrastructure (U-NII) bands in North America is described in the Part 15 of Federal Communications Commission (FCC) [7]. However, such regulations only serve as fundamental requirements for coexistence, and further optimisation is left open for specific technology.

The anticipated growth of unlicensed spectrum usage in the near future has made the coexistence issue a very important research topic. The focus of this project is on the coexistence mechanisms for WPAN working in unlicensed spectrum. In the next section, the WPAN concept will be briefly introduced.

1.2 Wireless Personal Area Network

The proliferation of mobile computing devices in the last decade has created a great demand for ubiquitous and effortless interconnection to share resources and information. Today, it is not uncommon to see a person carrying a watch, cellular phone, personal stereo, and Personal Digital Assistant (PDA) that have multiple displays, input interfaces, speakers and microphone. Redundant data entry, duplication of information, hardware Input/Output (I/O) components, and software functions can be avoided by having common connectivity between these devices. Also, it is desirable to have some kind of connection for a cellular phone to exchange information, such as its address book, with a PDA or a notebook. One possible solution for interconnection is cable. However, the most vexing problem that many people have with cables is hooking up and hiding them. Cables are also awkward to carry around. Fortunately, the recent introduction of WPAN has significantly simplified the scenario.

By definition, WPAN is a network that connects devices centred around an individual person, in which the connection medium is wireless. The primary goal of the WPAN is to remove all cumbersome cables needed for sharing information and resources between closely located devices. Undoubtedly, freedom from cables opens vast opportunities for WPANs: For example, WPAN is going to replace the untidy cables between keyboard, mouse and PC, or those that connect loud-speakers with the modern stereo system. In the sports/entertainment industry, sporting equipment and athletic apparel of the future can be equipped with WPAN-enabled sensors that communicate performance information back to an user display, perhaps in a wrist watch form factor. Medical monitoring is another interesting application for WPAN: Vital signs of patient can be easily monitored, even if he/she is in motion, with the help of WPAN-supported medical sensors. As a result, the WPAN market is expected to experience very strong growth in the next few years.
1.2.1 WPAN characteristics

A WPAN system possesses of several important characteristics, which dictate technology choice. First, WPAN typically permits communications in a very short range, i.e. within about 10 meters. This coverage area, defined as a Personal Area Network (PAN), is the next domain in the Wide Area Network (WAN) and Local Area Network (LAN) hierarchy (see Figure 1.1). Some WPAN technologies are often referred to as the “last 10 meters solution”, as they are designed to provide in-building short-range Internet hotspot services.

![Diagram of WPAN, WLAN, and WWAN with coverage distances]

Figure 1.1: WPAN in comparison with other types of wireless networks

Secondly, WPAN module should be small, inexpensive and low power consumption. To integrate into wrist watch or wearable sensors, the connection module must be lightweight and small enough, so that it does not intrude on the design of those devices. If WPAN is to replace cable, it cannot be much more expensive than the cable or nobody will buy it. Because WPAN is designed for mobile devices, it must be able to run with very low power consumption, so as to conserve the batteries on the devices.

Thirdly, WPAN operates on ad-hoc basis, i.e. WPAN devices are self-organised and do not require a central base station or fixed network infrastructure for operation. This is due to the fact that WPAN connections are usually temporary and do not follow a fixed network configuration, as WPAN devices frequently come in and out of coverage area. Operating on ad-hoc basis, WPAN devices are able to quickly and flexibly form
1.2 Wireless Personal Area Network

a network on their own whenever they come close together. Ad-hoc network structure also makes WPAN more robust against network configuration changes.

Fourthly, the medium on which the WPAN operates is wireless. In contrast to cable-based communication, which is inherently secure, anyone could potentially listen into a wireless transmission. Therefore, sufficient security measure is required for WPAN.

Finally, to ensure that WPAN-enabled device can operate anywhere and anytime, the WPAN technique must utilise the globally-available unlicensed spectrum, such as the ISM or U-NII band. Another good reason for WPAN to work on the license-free spectrum is that WPAN users generally do not want to pay for their local communications. As discussed earlier, device operating under unlicensed spectrum might be subjected to very high level of interference. Thus, WPAN must employ certain techniques to make it more robust against interference, and at the same time, to induce minimal interference to other system operating at the same band. Such techniques are referred to as coexistence mechanisms, and are the focus of the project.

1.2.2 WPAN standards

To enjoy ubiquitous usage, standardisation of WPAN technology is required. Currently, there are several WPAN standards, which are discussed in this section.

Bluetooth wireless technology

Bluetooth wireless technology, which is initially developed by Swedish mobile phone maker Ericsson in 1994 as a way to link their mobile phones with computer and other accessories, is the pioneer in WPAN concept. The technology is named after Harald Blåtand, a tenth-century Danish Viking king who united and controlled Denmark and Norway. The name was selected because Bluetooth wireless technology is expected to unify the telecommunications and computing industries [3].

The Bluetooth Special Interest Group (SIG), founded in September 1998, is a group of companies working together to define and promote the Bluetooth specifications. By granting free licence to build products using Bluetooth wireless technology, the Bluetooth SIG has made Bluetooth an open specification, rather than keeping it restricted and proprietary. Therefore, Bluetooth has become the most widely-accepted WPAN standard, with up to 2,000 member companies representing tens of thousands of individuals around the world [4].

Bluetooth wireless technology operates at 2.4GHz in the globally-available, license-free ISM band. It employs Frequency Hopping Spread Spectrum (FHSS) technique to mitigate interference and fading, as well as to make the transmission more secure. Bluetooth

provides full-duplex operation using Time Division Duplex (TDD). The maximum operation range is approximately 100 meters, but the normal range is about 10 meters. Bluetooth protocol supports both circuit- and packet-switching for audio and data transmission, respectively. The data rate can be up to 723.2kbps in either direction while permitting 57.6kbps in the return direction (asymmetric channel), or 433.9kbps in both directions for symmetric channel [2]. More detailed discussion about Bluetooth specifications can be found in the Chapter 2.

IEEE 802.15 WPAN standards

To develop a common set of standards and to address various issues related to WPAN, the Institute of Electrical and Electronic Engineers (IEEE) 802.15 Working Group was established in January 1999 as a part of the IEEE 802 Local and Metropolitan Area Network Standards Committee of the IEEE Computer Society. At present, the 802.15 Working Group is comprised of four subcommittees or Task Groups, whose activities include publishing standards, recommended practices, or guides that have broad market applicability and dealing effectively with the issues of coexistence and interoperability with other wired and wireless networking solutions [4].

At the time of writing, there are several WPAN standards being adopted and/or developed by the IEEE 802.15 Working Group. They are briefly introduced hereafter:

- Published in June 2002, the IEEE 802.15.1 standard was the first recommendation from the IEEE 802.15 Working Group. This standard was based upon portions of the Bluetooth specification version 1.1. Since the Bluetooth SIG has introduced the version 1.2 of their specification, the IEEE 802.15 Working Group is reviewing necessary changes and will reflect them on the 802.15.1a standard.

- The IEEE 802.15.2 standard, which is approved in June 2003, discusses the issue of coexistence of WPAN and WLAN, which operate in the same unlicensed band. Particularly, it provides several coexistence mechanisms that can be used to facilitate coexistence of the IEEE WPAN 802.15.1 in the presence of the IEEE 802.11 and 802.11b WLAN.

- The IEEE 802.15.3 draft standard defines the Medium Access Control (MAC) and Physical Layer (PHY) specifications for the High Rate - Wireless Personal Network (HR-WPAN). The goal of this standard is to achieve a low complexity, low cost, low power consumption and high data rate wireless connectivity for devices within the Personal Operating Space (POS). The data rate will be high enough, 20Mbps or more, to satisfy a set of consumer multimedia needs for WPAN communications. The draft standard also addresses the Quality of Service (QoS) capabilities required to support multimedia data types [3].
1.3 Scope of project

1.3.1 Problem identification

The Bluetooth wireless technology operates in the unlicensed 2.4GHz ISM band. Extending from 2,400MHz to 2,483.5MHz for most countries in the world, this band might also be used by the other ISM emitters, namely:

- **The IEEE 802.11**: This standard defines two versions of the PHY layer for WLAN. The FHSS PHY uses 1MHz channel separation and hops pseudo-randomly over 79 channels, while the Direct Sequence Spread Spectrum (DSSS) PHY uses 22MHz channel and may support up to 3 non-overlapping channels in the ISM spectrum. Both of them support 1- and 2-Mbps data rates.

- **The IEEE 802.11b DSSS**: This WLAN standard, also known as “Wi-Fi”, is extended version of the IEEE 802.11 DSSS PHY. It provides two new data rates of 5.5- and 11-Mbps using the Complementary Code Keying (CCK) technique.

- **The IEEE 802.11g**: This is a new WLAN standard, which can achieve a maximum data rate of 54Mbps using Orthogonal Frequency Division Multiplex (OFDM) technique. It is also backward compatible with 802.11b standard.

- **The IEEE 802.15.3 High-Rate and 802.15.4 Low-Rate**: These are draft standards for WPAN, which are mentioned in section [1.2.2](#).

- The other short-range proprietary wireless techniques (such as car security, cordless headphone, etc.) and random noise generators (e.g. microwave ovens and sodium vapour street lamps) [5].
As a result, Bluetooth wireless technology is subjected to mutual interference with these ISM transmitters, which may result in severe performance degradation. The level of interference depends on many factors, namely, number of and distance between interfering devices, transmitting power level, data rate and amount of data traffic flowing over each of the wireless links. And also, different types of information being sent over the wireless networks have different levels of sensitivity to interference. For example, a voice link may be more sensitive to interference than a data link being used to transfer a data file [3]. From the point of view of the Bluetooth wireless technology, interference sources can be classified into the following categories:

- **Noise**: The noise is uncorrelated in time and yields uniform error over the whole set of frequencies. Therefore, it is not possible to combat the effect of noise by just employing some intelligent FHSS patterns [19].

- **Frequency-static interference (FSI)**: This type of interference often occupies a group of frequencies for a period that is much longer than the Bluetooth packet duration. The WLAN IEEE 802.11b DSSS is a canonical example of FSI for Bluetooth [6].

- **Frequency-dynamic interference**: In contrast to the FSI, frequency-dynamic interference occurs in a group of frequencies only for a short period, which is less or approximately equal to the Bluetooth packet duration. Bluetooth self-interference, which occurs when two or more collocated Bluetooth networks simultaneously select the same frequency for their transmission, falls into this category. As the WPAN proliferation gains momentum, the probability of having many Bluetooth networks in a close proximity of each others is increased, and so is the level of self-interference.

To alleviate the impacts of interference, Bluetooth wireless technology needs to employ some forms of coexistence mechanisms. The IEEE 802.15.2 is the first published standard on the coexistence problem between Bluetooth and WLAN, and it also recommends several coexistence mechanisms to minimise such mutual interference. According to this standard, there are two categories of coexistence mechanisms: **collaborative** and **non-collaborative**. In collaborative mechanisms, the interfering entities explicitly exchange data to achieve mutual coordination, while non-collaborative mechanisms consider that data exchange is not available. The collaborative techniques can be implemented only when Bluetooth and WLAN are collocated in the same physical unit, or a communication link exists between these devices [3]. The non-collaborative mechanisms are of our interest in this project.

The Adaptive Frequency Hopping (AFH) is one of non-collaborative coexistence mechanisms recommended by the IEEE 802.15.2 standard. The Bluetooth SIG currently believes that the AFH is best suited for scenarios where the Bluetooth device is not
1.3 Scope of project

Physically located with the device with which it is interfering. As such, it has been incorporated in the Bluetooth specification version 1.2. The AFH algorithm dynamically changes the frequency hopping sequence of Bluetooth device by removing those “bad” frequencies (which have high Packet Error Rate (PER) for instance) from its hopset for a certain period. This strategy is intended to alleviate interference resulting from the operation of the Bluetooth devices in the presence of frequency-static or slow-hopping WLAN devices. It is not, however, designed to deal with the self-interference between Bluetooth networks. In fact, the IEEE 802.15.2 standard indicates that the pseudo-random frequency hopping employed by Bluetooth wireless technology is responsible for mitigating the self-interference [2]. Nevertheless, by reducing hopset to avoid FSI, AFH tends to increase level of self-interference between Bluetooth networks, resulting in degradation of goodput.

As Bluetooth wireless technology is expected to be present in many commercial products, from head phones, mobile phones and PDAs to laptop and desktop computers, the problem of self-interference becomes more and more significant. In [19] and [18], two non-collaborative approaches, namely Dynamic Adaptive Frequency Hopping (DAFH) and Frequency Rolling (FR), have been introduced to mitigate the self-interference. Both of them have been proven to be effective against self-interference with drastic improvement on throughput performance. It is desirable to analyse these approaches and to propose necessary modifications for them to work with both FSI and self-interference.

1.3.2 Objectives of the project

Under the title “Non-collaborative coexistence mechanism for WPAN in unlicensed spectrum”, this project investigates the possible approaches for WPAN, particularly the Bluetooth wireless technology, to overcome the effects of both frequency-static and self-interference. The main objectives of the project are as follows:

- **To analyse the probabilities of packet collision for Bluetooth piconet due to frequency-static and self-interference.** Previous study on Bluetooth probabilities of packet collision, such as in [8], assumed that packet size is always equal to a slot duration. In this project, the probabilities of packet collision for arbitrary packet size are derived. These probabilities shall be used for validating the result of our Bluetooth simulator.

- **To implement a variant of the AFH mechanism and evaluate its performance under frequency-static and self-interference scenario.** The main structure for AFH has been defined in Bluetooth specification version 1.2. However, detailed implementation is left open for specific vendor. In this project, a variant of AFH mechanism is proposed, which acts as a reference for assessing performance of the other coexistence methods.
• To analyse the performance of DAFH mechanism under frequency-static and self-interference: Although DAFH mechanism is inherently robust against FSI, it has not been evaluated with such interference. This project studies the ability of DAFH in combating the FSI.

• To propose necessary modifications for FR mechanism to mitigate both frequency-static and self-interference, and investigate its performance: The FR mechanism is proposed in [18] for mitigating only self-interference. In this project, we are going to propose necessary modifications to make it also robust against FSI. Performance of the proposals are evaluated by means of simulation.

In this project, the FSI model will be the IEEE 802.11b DSSS WLAN. Among other ISM transmitters listed in section [1.3.1], the IEEE 802.11 FHSS and 802.11b DSSS are the most commonly-used standards in commercial products (due to their late presence, the IEEE 802.11g, 802.15.3 and 802.15.4 have not been widely deployed). According to the IEEE 802.15.2 recommendation, the IEEE 802.11b DSSS interference model is recommended because it represents a worse interferer than the IEEE 802.11 FHSS [3]. Other studies on coexistence mechanisms, such as [3] and [9], have also considered the IEEE 802.11b DSSS as their interference model.

1.4 Summaries

The unlicensed spectrum presents both opportunities and threats for the WLAN and WPAN technologies. While enjoying ubiquitous communication, WPAN, such as Bluetooth wireless technology, has to cope with high and varying level of interference coming from other wireless applications and from WPAN itself. Therefore, it is desirable for WPAN to employ coexistence mechanisms to mitigate the effect of interference and also to reduce its interference to others.

The main goals of this project are to propose necessary modifications for existing coexistence mechanisms for Bluetooth wireless technology to cope with both FSI and self-interference problems, and to evaluate their performance by means of simulation. The FSI model for investigation will be the IEEE 802.11b DSSS WLAN.
Bluetooth wireless technology

A key feature of Bluetooth specification is that it aims to allow different applications to effortlessly share information and resources between one another. To this end, Bluetooth does not just define a radio system, it also specifies a software stack to enable applications to find other Bluetooth devices in the area, discover what services they can offer, and use those services. Figure 2.1 illustrates the protocol stack of the Bluetooth wireless technology.

![Diagram of Bluetooth architecture](image)

Figure 2.1: Bluetooth architecture [11]

In this chapter, we limit ourselves to introducing only the RF and baseband layers of the Bluetooth specification, which are relevant to our project. Interested reader can refer to [2] for more detailed information of the Bluetooth protocol stack.

2.1 Radio Frequency Layer

The RF layer acts as a bridge between the baseband layer and the physical channel: It is responsible for receiving packets of information from the physical channel and delivering them to the baseband layer, and vice versa. The key characteristics of the RF layer are discussed in the following sections.
2.1.1 Frequency Hopping

As discussed earlier, Bluetooth devices operate in the unlicensed 2.4GHz ISM band, which is not a very stable or reliable medium. To cope with such hostile environment, Bluetooth specifically employs FHSS as a technique to combat interference and fading. Originally invented by the Austrian born actress Hedy Lamarr during World War II, the frequency hopping technique introduces both robustness and security to wireless communication, which are important for Bluetooth. In a frequency hopping system, the transmitter transmits a short burst of data over one frequency and then “hops” to a different frequency before sending another data burst. If the frequency hopping occurs at a rate that is faster than the information bit rate, it is called fast hopping system, otherwise it is referred to as slow hopping system. The transmission frequencies are determined by a hopping, or spreading, sequence. To properly receive the signal, the receiver must apply the same hopping sequence as the transmitter, which allows it to listen to the incoming signal at the right time and at the correct frequency. The Bluetooth wireless technology is a slow hopping system, in which one data packet is transmitted before the frequency hops to the next channel. The nominal hop rate of Bluetooth is 1600 hops per second. There are total 79 hopping channels in Bluetooth specification. These RF channels are spaced 1MHz and are ordered in channel number $q$ as shown in Eq. (2.1):

$$f_q = 2402 + q \quad (MHz) \quad q = 0, 1, \ldots, 78$$ (2.1)

where $f_q$ is the centre frequency of the $q^{th}$ RF channel, and $q$ is the corresponding channel number. The baseband layer, which will be discussed in the next section, employs special algorithm to calculate the common pseudo-random hopping sequence for a group of Bluetooth devices, so that they can exchange information. Such a group is referred to as a piconet. The employed algorithm ensures a maximum distance between adjacent hop channels in the sequence and a low correlation between sequences used by other piconets. The first is to mitigate the effect of fading and interference, while the latter is to minimise the collision probability between those Bluetooth piconets.

2.1.2 Modulation

The modulation scheme of Bluetooth is Gaussian Frequency Shift Keying (GFSK), which is selected to minimise transceiver complexity. In GFSK, the transmitted binary data is first filtered by a Gaussian filter in the baseband and then modulated with a simple frequency modulation. In GFSK, the modulated carrier frequency changes smoothly with

---

1Previously, Bluetooth specification version 1.0b defined only 23 hopping channels for France, Spain, Japan and few other countries due to the limitation of ISM spectrum at these geographical regions. However, Bluetooth SIG has negotiated with the 23-hop countries to allow the usage of 79-hop equipment, which leads to elimination of the 23-hop option from Bluetooth specification version 1.1.[13]
a Gaussian shaped envelope. This maintains continuous phase of the carrier frequency and reduces the emitted spectral sidelobes, allowing better spectral efficiency and less inter-symbol interference [3].

To obtain the most efficient use of bandwidth while still maintaining acceptable error probability, the digital bit stream is modulated using GFSK with a bandwidth-bit period product $BT = 0.5$ and a modulation index of between 0.28 and 0.35 [2]. The $BT$ product is the product of adjacent signal frequency separation (i.e. $0.5\text{MHz}$) and symbol duration ($1\mu s$). A $BT$ product of 0.5 corresponds to the minimum carrier separation to ensure orthogonality between signals in adjacent channels. The modulation index represents the strength of the peak frequency deviation ($f_d$) and can be expressed as $2f_dT$, where $T$ is the symbol duration [3]. The Bluetooth symbol duration is $1\mu s$, thus the modulation index can be translated to a frequency deviation range of $140\text{kHz}$ to $175\text{kHz}$. The Bluetooth specification gives $115\text{kHz}$ as an absolute minimum deviation [2].

### 2.1.3 Power emission and control

Table 2.1 presents three power classes defined in the Bluetooth specification. Power class 3 is the most common scheme adopted by manufacturers, and is also the lowest power consumption option. Power class 1 has a mandatory requirement for power control, while power class 2 and 3 make this optional. Nevertheless, for minimising power consumption and reducing interference level, power control is always preferable.

<table>
<thead>
<tr>
<th>Power Class</th>
<th>Maximum Output Power ($P_{\text{max}}$)</th>
<th>Power Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100mW (20dBm)</td>
<td>Mandatory: 4dBm to 20dBm, Optional: -30dBm to 4dBm</td>
</tr>
<tr>
<td>2</td>
<td>2.5mW (4dBm)</td>
<td>Optional: -30dBm to 4dBm</td>
</tr>
<tr>
<td>3</td>
<td>1mW (0dBm)</td>
<td>Optional: -30dBm to 0dBm</td>
</tr>
</tbody>
</table>

Power control operates by a receiver monitoring the Received Signal Strength Indicator (RSSI) and sending Link Manager Protocol (LMP) control command back to the transmitter, asking for their transmit power to be reduced if RSSI value is higher than what is required to maintain a satisfactory link. If the RSSI value drops too low, then the receiver may request the power to be increased. The specification requires power to be controlled in steps of $2\text{dB}$ to $8\text{dB}$.
2.2 Baseband Layer

The RF and baseband layers in Bluetooth correspond to the first layer, or the physical layer, of Open System Interconnection (OSI) model. The baseband is responsible for channel coding and decoding, low level timing control and management of the link within the domain of a single data packet transfer. There are control paths between the baseband and the RF layer, which allow the baseband to control the timing and frequency carrier of the radio block.

2.2.1 Bluetooth Device Address

Each Bluetooth device has an unique 48-bit MAC address, known as the Bluetooth Device Address \(BD\_ADDR\). The address is split into Non-significant Address Part (NAP), Upper Address Part (UAP) and Lower Address Part (LAP) as follows [2]:

- \(BD\_ADDR[31:24] = UAP[7:0]\) : Used to initialise the Header Error Check (HEC) and Cyclic Redundancy Checksum (CRC) calculations and for frequency hopping.
- \(BD\_ADDR[23:0] = LAP[23:0]\) : Used by the synchronization word generation and frequency hopping.

Similar to the Ethernet MAC address, the Bluetooth Device Address must be obtained from IEEE Registration Authority to ensure its uniqueness and to facilitate authentication process.

2.2.2 Master, slave, piconet and scatternet

A piconet is group of Bluetooth devices that share a common frequency hopping pattern. Within the piconet, one device provides the synchronization reference and is known as the \textit{master}. All other devices are referred to as \textit{slaves}. Up to seven slaves can be active in a piconet, i.e. they can be directly addressed by the master. Additionally, many more slaves can remain connected to the piconet in the parked state. These parked slaves can neither transmit data nor be addressed directly by the master. However, they remain synchronised to the master and wake up periodically to listen for broadcasts, which are used to bring them back to active life without going through the connection establishment procedure [2].
A Bluetooth device can act as a master or a slave at a particular time, but not simultaneously. Communication within a piconet can be point-to-point, if there are only one master and one slave, or point-to-multipoint, if two or more slaves are present (see Figure 2.2).

![Diagram](image)

**Figure 2.2: Bluetooth ad-hoc network:** (a) Point-to-point piconet. (b) Point-to-multipoint piconet, and (c) Scatternet

Two or more piconets connecting together form a *scatternet*, providing a larger coverage area and/or a greater number of devices in the network. In a scatternet, some Bluetooth devices will be members of more than one piconet: They could be master in one piconet and slave in the others, or function as slave in all piconets of which they are members.

### 2.2.3 Physical link between master and slave

The baseband layer specifies a TDD mechanism within a piconet, where the master and its slaves alternatively transmit data. The basic timing unit is $T_s = 625\mu$s, which is often referred to as a *time slot*. The master always starts transmitting at even time slots, while the slave uses the odd time slots for its transmission (see Figure 2.3). The term “slot pairs” is used to indicate two adjacent time slots starting with a master-to-slave transmission slot (i.e. even slot). In order to operate properly, all active slaves in the piconet need to be time- and frequency-synchronised to the master.

The Bluetooth specification version 1.2 defines five different types of link that can be established between a master and its slave(s), namely *Asynchronous Connection-Less (ACL)*, *Synchronous Connection-Oriented (SCO)*, *Extended Synchronous Connection-Oriented (eSCO)*, *Active Slave Broadcast (ASB)* and *Parked Slave Broadcast (PSB)*. In this section, we will briefly discuss the first two link types.
The ACL link

Figure 2.3 illustrates the operation of ACL links between a master and three slaves. The master transmits data packets to the slaves in sequence. The slave may only respond with an ACL packet in the next slave-to-master slot if it has been addressed in the preceding master-to-slave slot. In this case, as the slave 3 fails to decode the address in the packet header due to packet error, it does not know for sure whether it has been addressed or not and hence does not respond.

![Diagram of piconet operation with ACL links]

The ACL link exists between a master and a slave as soon as a connection has been established. A master may have a number of ACL links to a number of different slaves at any one time, but only one link can exist between any two devices. The ACL link provides a packet-switched connection where data is exchanged irregularly as and when data is available from higher layer. The choice of which slave to transmit to and receive from is decided by the scheduling policy at the master and implemented by polling on a slot-by-slot basis, and so both asynchronous and isochronous (time-bounded) services are possible. Most ACL packets facilitate error checking and retransmission to assure data integrity [5].

The SCO link

In contrast to the ACL link, a SCO provides a symmetric link between a master and a slave with reserved channel bandwidth and regular periodic exchange of data in the form...
of reserved slots. Therefore, the SCO link can be seen as a circuit-switched connection where data is regularly exchanged, and as such it is intended for time-bounded services, such as audio.

A master can support up to three SCO links to the same slave or to different slaves. A slave can support up to three SCO links from the same master. Due to time-bounded nature of SCO data, SCO packets are never retransmitted.\(^2\)

\[\text{Figure 2.4: piconet in operation with a SCO link}\]

The master will transmit SCO packets to the slave at regular intervals, defined by the parameter \(T_{SCO}\). This is referred to as the SCO interval and is counted in slots. The slave is always allowed to respond with a SCO packet in the reserved response slot, unless it correctly decodes the packet header and discovers that it had not been addressed as expected. If the packet is incorrectly decoded due to errors, then the slave can still respond as the slot is reserved (see Figure 2.4). Normally, in the reserved slot, the master is not allowed to transmit to other but the SCO slave. The only exception is the broadcast message, which would take precedence over the SCO link.

A SCO link is established from ACL link via Link Manager command from the master to the slave. This message will contain timing parameters to specify the reserved slots such as the SCO interval and the starting offset.

\(^2\)The Bluetooth specification version 1.2 further defines the eSCO link, where packets can be re-transmitted.\(^2\)
2.2.4 Selection of frequency hopping pattern

The most important role of the baseband layer is to control and select the frequency hopping pattern that is shared among Bluetooth devices within a piconet. The Bluetooth devices use the frequency hop selection kernel to determine the appropriate frequency channel for transmitting and receiving at any given point in time. As illustrated in Figure 2.5, the frequency channel is determined solely by the address and clock of the master. The Upper Address Part (UAP) and Lower Address Part (LAP) is used to decide the exact hop sequence to be used, while the clock simply determines the phase in that sequence [2]. The master’s address and clock are transmitted to slaves during an Inquiry process. During this process, the slave calculates the clock offset between its clock and that of the master. With this offset and the master’s address, the slave is able to synchronize its hop sequence with that of the master.

![Diagram](image)

Figure 2.5: The Bluetooth basic frequency hop selection kernel

The output of the frequency hop selection kernel is the RF channel index, constitutes a pseudo-random sequence (see Figure 2.5). The version 1.2 of Bluetooth specifications defines six types of hopping sequences, in which four are for the Paging and Inquiry states, and two for Connection state. The two hopping sequences for Connection state, which are important to this project, are summarised as following [2]:

- A basic channel hopping sequence has a very long period length, which does not show repetitive patterns over a short time interval, and which distributes the hop frequencies equally over the 79MHz during a short time interval. Channel hops are produced at a rate of one per slot, except during multi-slot packets. There is no hop during a multi-slot packet, although the sequence continues unhindered after the packet (see Figure 2.6).

- An adapted channel hopping sequence is derived from the basic channel hopping sequence which uses the “same channel mechanism” and may use fewer than 79 frequencies. This sequence is a part of the AFH coexistence mechanism, which will be discussed in details in the next chapter.

Detailed algorithm for constructing the basic and adapted channel hopping sequences can be found in [2]. As proved in [10], in the long run, the frequency hopping selection kernel of Bluetooth device can be approximated by a pseudo-random number generator.
2.2.5 Bluetooth packet structure

Bluetooth specification defines data packets which are 1, 3, and 5 slots long, and Figure 2.6 shows how the timing changes slightly for these multi-slot packets. The start of packets is always aligned with the slot start. All packets have the same amount of header and control data overhead, and therefore a multi-slot packet is more data efficient (i.e., the longer the packet, the lower the ratio between overhead information and data within one packet). On the other hand, the multi-slot packet has more bits to send, resulting in a higher probability of packet error due to channel impairment or interferences. As a result, the choice of packet size is the trade-off between data efficiency and reliability to achieve the optimal throughput.

![Diagram of slot timing for multi-slot packets](image)

The general packet format is shown in Figure 2.7, which consists of three entities: the access code, the packet header and the payload. Every packet must start with an access code. If a packet header follows, the access code is 72 bits long, otherwise it is 68 bits long and is known as a shortened access code. The access code is used for synchronization, DC offset compensation, detection of the presence of a packet and addressing the packet to specific devices. For example, slaves detect the presence of a packet by matching the access code against their stored copy of the master’s access code.

Following the access code is the packet header, which contains all control information associated with the packet and the link, such as the address of the slave for which the packet is intended. In total, the header has only 18 bits of information, including 8 bits of HEC to verify the integrity of the header. These bits are protected by a Forward Error Correction (FEC) code of 1/3. This encoder replicates the data 3 times so that 18 bits occupy 54 bit periods on air. This high level of redundancy and coding overhead
is employed because each field of the header is crucial to the correct operation of the link control protocol and thus the link. At the receiver, if the HEC does not match, the entire packet will be discarded.

Finally, the structure of payload field depends on the types of packet. For ACL packet, the payload consists of a payload header, payload data and a CRC field. The payload header contains the information for the logical link control, while the CRC field is used to validate the integrity of both received payload header and data. The payload data of ACL packet can have variable length, from 0 to 2712 bits. On the other hand, the payload of SCO packet is fixed in length (i.e. 240 bits), and there is no payload header or CRC field appended to the payload.

Table 2.2 lists several common packet types defined in the Bluetooth specifications and their particular characteristics.

- **ID packet**: The Identification (ID) packet consists of only the shortened access code and is used during “pre-connection” operation, where the link has not yet been established and the relative timing between devices may be unrelated. This packet acts as a highly robust signalling mechanism: The only information it carries is the access code of the device it is coming from or going to. It is detected and decoded simply by correlating against the desired access code [5].

- **NULL packet**: The NULL packet has no payload and consists of only the channel access code and packet header, which is 126 bits in total. It is used when the
### Table 2.2: Different types of Bluetooth packet

<table>
<thead>
<tr>
<th>Packet Type</th>
<th>Payload Header (bytes)</th>
<th>User Payload (bytes)</th>
<th>FEC rate</th>
<th>CRC</th>
<th>Symmetric Max. Rate (kb/s)</th>
<th>Asymmetric Max. Rate (kb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
</tr>
<tr>
<td>NULL</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
</tr>
<tr>
<td>POLL</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
<td>N/a</td>
</tr>
<tr>
<td>FHS</td>
<td>N/a</td>
<td>18</td>
<td>2/3</td>
<td>Yes</td>
<td>108.8</td>
<td>108.8</td>
</tr>
<tr>
<td>DM1</td>
<td>1</td>
<td>0-17</td>
<td>2/3</td>
<td>Yes</td>
<td>108.8</td>
<td>108.8</td>
</tr>
<tr>
<td>DM3</td>
<td>2</td>
<td>0-121</td>
<td>2/3</td>
<td>Yes</td>
<td>258.1</td>
<td>387.2</td>
</tr>
<tr>
<td>DM5</td>
<td>2</td>
<td>0-224</td>
<td>2/3</td>
<td>Yes</td>
<td>286.7</td>
<td>477.8</td>
</tr>
<tr>
<td>DH1</td>
<td>1</td>
<td>0-27</td>
<td>No</td>
<td>Yes</td>
<td>172.8</td>
<td>172.8</td>
</tr>
<tr>
<td>DH3</td>
<td>2</td>
<td>0-183</td>
<td>No</td>
<td>Yes</td>
<td>390.4</td>
<td>585.6</td>
</tr>
<tr>
<td>DH5</td>
<td>2</td>
<td>0-339</td>
<td>No</td>
<td>Yes</td>
<td>433.9</td>
<td>723.2</td>
</tr>
<tr>
<td>HV1</td>
<td>N/a</td>
<td>10</td>
<td>1/3</td>
<td>No</td>
<td>64</td>
<td>N/a</td>
</tr>
<tr>
<td>HV2</td>
<td>N/a</td>
<td>20</td>
<td>2/3</td>
<td>No</td>
<td>64</td>
<td>N/a</td>
</tr>
<tr>
<td>HV3</td>
<td>N/a</td>
<td>30</td>
<td>No</td>
<td>No</td>
<td>64</td>
<td>N/a</td>
</tr>
</tbody>
</table>

link control information carried by the packet header has to be conveyed between Bluetooth devices, but they do not have any data to send. The receiver of a NULL packet does not have to respond to it.

- **POLL packet:** This packet is very similar to the NULL packet. It does not have a payload. In contrast to the NULL packet, upon reception of a POLL packet, a slave must respond with a packet even when it does not have any information to send, unless the slave has scatternet commitments in that time slot. The POLL is not a part of the Automatic Repeat Request (ARQ) scheme. Thus, this packet can be used by the master in a piconet to poll the slaves for data or to check if a particular slave is still present in the piconet. Slaves are not allowed to transmit the POLL packet.

- **FHS packet:** The Frequency Hop Synchronisation (FHS) packet is a special control packet containing, among other things, the Bluetooth device address and the clock of the sender. Therefore, the FHS packet provides all information required by the recipient to address the sender in terms of timing, frequency hopping and access code. The payload contains 144 information bits plus a 16-bit CRC code. The payload is then coded with a rate 2/3 FEC to make a gross payload length of 240 bits.

- **DM# packet:** DM packet is a data packet with FEC of 2/3 to make the packet more robust against errors. In addition to FEC, the CRC is also employed to detect
errors in the payload. DM packet can be 1, 3 or 5 time slots long, which are denoted as DM1, DM3 or DM5, respectively.

- **DH# packet**: DH packet is also data packet, but without the FEC, such that the payload size is increased. Similar to DM packet, DH packet employs CRC for validating data integrity.

- **HV# packet**: HV packet is voice packet that spans only one time slot. Due to the time-bounded nature of voice services, HV packet will never be retransmitted. Therefore, it does not make sense to use CRC for HV packets. Instead, different levels of FEC are available for HV packets. HV1 packet carries a payload size of 10 bytes and uses rate 1/3 FEC. HV2 packet has a payload of 20 bytes and uses rate 2/3 FEC. HV3 packet does not use FEC and can carry 30 bytes in its payload.

Bluetooth specification employs a fast, unnumbered ARQ scheme for packet types with CRC-protected payload, such as DM or DH. These types of packet will be re-transmitted until an acknowledgement of a successful reception is returned by the destination, or a timeout is exceeded. The acknowledgement information is included in the header of return packet: The slave will respond in the slave-to-master slot directly following the master-to-slave slot, unless it has scatternet commitments in that time slot; while the master will respond at the next event addressing the same slave. For a packet reception to take part, at least the HEC must pass. In addition, the CRC must pass if present [2].

### 2.2.6 Exchange of control information

Control information, such as LMP or Logical Link Control and Adaptation Protocol (L2CAP) messages, can be passed between the master and its slaves in two different ways:

- **Unicast**: The master may send the control information to its slaves using point-to-point communication. The slave replies with an acknowledgement to let the master know whether or not the transmission has been received correctly. This scheme is very reliable, as the master can verify the correct reception of the control messages. However, the master will need to repeat the same messages for all slaves in the piconet, which can reduce the total throughput of the piconet.

- **Broadcast**: On the other hand, the master could broadcast the control information to all of its slaves simultaneously. The advantage of broadcasting is that this method might be faster in some situation and all the slaves could receive the control information at the same time. However, the slaves cannot acknowledge the receipt of broadcast packet, because if all of them reply at once, their transmission would interfere with each other and no data would get through. As a result, the master
does not know whether a broadcast packet has been received successfully or it has been wiped out by radio interference and should be retransmitted. To make the broadcast link reasonably reliable, the master must transmit each broadcast packet for a fixed number of time \((N_{BC})\) \([2]\). The number of broadcast retransmissions should be adjusted to the quality of the link and the target reliability (i.e. \(N_{BC}\) should be set to a larger value if the link quality deteriorates and/or a higher reliability is required).

### 2.3 Probability of packet collision for Bluetooth piconets

As discussed in the previous sections, the master and the slaves in a piconet share the wireless medium in time-division manner, so that the communication within the piconet is collision-free. However, packet collision is still possible for Bluetooth piconet, and the major causes of collision are the self-interference and the FSI. The packet collision due to self-interference happens if two or more piconets that are in the vicinity of each others simultaneously select the same hopping frequency for their transmission. In addition, packet collision also occurs if a piconet transmits using the same frequencies as those of the closely-located FSI, such as the IEEE 802.11b WLAN. This type of collision is referred to as collision induced by FSI. The purpose of this section is to derive the probabilities of packet collision for Bluetooth piconet due to either self-interference or FSI. These probabilities shall be used for validation of the Bluetooth simulator in the next chapter.

We consider the scenario where there are a number of piconets, \(N_p\), in close proximity of each others. Let denote the \(i^{th}\) piconet by \(\pi_i\), where \(i = 1, 2, \ldots N_p\). We assume that the distance between the devices within a piconet is small enough, so that the following approximation can be made: If at least one device in \(\pi_i\) is in transmission range of a device in \(\pi_j\), then each device in \(\pi_i\) is in transmission range of any device in \(\pi_j\), and vice versa. If this happens, \(\pi_i\) and \(\pi_j\) are said to be collocated, and they will perceive all transmission from the other as though they are coming from a single transmitter. Based on this assumption, a Bluetooth piconet can be abstracted as one transmitter with slotted channel (see Figure 2.8).

Several other assumptions are made to further simplify the scenario. First, we assume that the collocated piconets are all in Connection state. Secondly, they are fully loaded, i.e. they always have packets to send. Thirdly, packet header is assumed to be error-free, since it is very short and heavily protected with 1/3 rate FEC. Under second and third assumptions, slaves will always respond in the next slave-to-master slot, because they have packets to send and they can decode successfully their addresses in packet headers. If this is not the case, the probability of collision might be slightly reduced, as channel occupancy of the piconet is apparently lower. Finally, we assume that all piconets use
only 1-slot packet for their transmissions. The probabilities of packet collision in case of only 3- or 5-slot packets are used can be derived in similar manner.

### 2.3.1 Collision induced by self-interference

To derive the probability of collision induced by self-interference, we define $\pi_1$ as the “reference piconet”, and the other piconets are collocated with $\pi_1$ in this scenario. We assume that all piconets are asynchronous in a sense that their slot-starts are not coinciding, making a packet in $\pi_1$ overlapping with either one or two packets of other piconets (as illustrated in Figure 2.8). Suppose that $\pi_1$ is transmitting at frequency $f_{\pi_1}$. Since the current packet of $\pi_1$ overlaps with only one packet of $\pi_2$, the probability that $\pi_2$ will select the same frequency as $\pi_1$ for its transmission is $\frac{1}{M}$, where $M$ is the total number of frequency channels (i.e. 79 channels for Bluetooth). Thus, the probability of no collision for 1-packet overlapping between $\pi_1$ and $\pi_2$ is given by:

$$P_{\text{no col, si, 1-pk overlap, 2 piconets}} = 1 - \frac{1}{M} \quad (2.2)$$

Assume that all $N_p - 1$ collocated piconets overlap only one packet with the piconet of interest, then the probability of no collision becomes:

$$P_{\text{no col, si, 1-pk overlap, } N_p \text{ piconets}} = \left(1 - \frac{1}{M}\right)^{N_p-1} \quad (2.3)$$
And the probability of collision for 1-packet overlapping scenario is as follows:

\[ P_{\text{col}, \text{s}, 1\text{-pk overlap}, N_p \text{ picnets}} = 1 - P_{\text{no col}, \text{s}, 1\text{-pk overlap}, N_p \text{ picnets}} = 1 - \left(1 - \frac{1}{M}\right)^{N_p-1} \]  

\[ (2.4) \]

On the other hand, \( \pi_3 \) has two packets overlapping with the current packet of \( \pi_1 \). The probability of no collision in this case can be expressed as:

\[ P_{\text{no col}, \text{s}, 2\text{-pk overlap, 2 picnets}} = \left(1 - \frac{1}{M}\right)^2 \]  

\[ (2.5) \]

Similar to 1-packet overlapping case, the probability of collision at \( \pi_1 \) for 2-packet overlapping scenario with \( N_p - 1 \) collocated picnets is as following:

\[ P_{\text{col}, \text{s}, 2\text{-pk overlap, N_p picnets}} = 1 - P_{\text{no col}, \text{s}, 2\text{-pk overlap, N_p picnets}} = 1 - \left(1 - \frac{1}{M}\right)^{2N_p-2} \]  

\[ (2.6) \]

Apparently, the probability of collision in Eq. \( (2.6) \) is higher than that of Eq. \( (2.4) \). In general, the picnet of interest might experience both 1- and 2-packet overlapping situations at the same time. The probability of collision, therefore, must be expressed in terms of the conditional collision probabilities:

\[ P_{\text{col, s, N_p picnets}} = P_{1\text{-pk overlap}} \times P_{\text{col, s, 1-pk overlap, N_p picnets}} + P_{2\text{-pk overlap}} \times P_{\text{col, s, 2-pk overlap, N_p picnets}} \]  

\[ (2.7) \]

where \( P_{1\text{-pk overlap}} \) and \( P_{2\text{-pk overlap}} \) are probability that 1- and 2-packet overlapping scenarios will happen, respectively. These probabilities depend on the relative position of the transmitted packet of \( \pi_1 \) with the packets from the other collocated picnets. We denote the duration of the transmitted packet as \( T_{\text{on}} \) and the remaining slot time after packet transmission as \( T_{\text{off}} \). If \( T_{\text{on}} \) is greater than or equal to \( T_{\text{off}} \), then \( P_{1\text{-pk overlap}, T_{\text{on}} \geq T_{\text{off}}} \) and \( P_{2\text{-pk overlap}, T_{\text{on}} \geq T_{\text{off}}} \) are given as:

\[ P_{1\text{-pk overlap}, T_{\text{on}} \geq T_{\text{off}}} = \frac{T_s - (T_{\text{on}} - T_{\text{off}})}{T_s} \]  

\[ (2.8) \]

\[ P_{2\text{-pk overlap}, T_{\text{on}} \geq T_{\text{off}}} = \frac{(T_{\text{on}} - T_{\text{off}})}{T_s} \]  

\[ (2.9) \]

From Eq. \( (2.4) \), \( (2.6) \), \( (2.7) \), \( (2.8) \) and \( (2.9) \), we can derive the corresponding probability of collision induced by self-interference for Bluetooth:

\[ P_{\text{col, s, N_p picnets}, T_{\text{on}} \geq T_{\text{off}}} = \frac{T_s - (T_{\text{on}} - T_{\text{off}})}{T_s} \times \left[1 - \left(1 - \frac{1}{M}\right)^{N_p-1}\right] \]

\[ + \frac{(T_{\text{on}} - T_{\text{off}})}{T_s} \times \left[1 - \left(1 - \frac{1}{M}\right)^{2N_p-2}\right] \]  

\[ (2.10) \]
On the other hand, in the case where $T_{on}$ is smaller than $T_{off}$, $P_{2\text{pk overlap}, \ T_{on}<T_{off}}$ is always zero and:

$$P_{2\text{pk overlap}, \ T_{on}<T_{off}} = \frac{T_s - (T_{off} - T_{on})}{T_s} \quad (2.11)$$

Hence the probability of collision in this case is given as:

$$P_{\text{col, si, } N_p \text{ piconets}, \ T_{on}<T_{off}} = \frac{T_s - (T_{off} - T_{on})}{T_s} \times \left[ 1 - \left(1 - \frac{1}{M}\right)^{N_p-1} \right] \quad (2.12)$$

From Eq (2.10) and (2.12), we observe that if the duration of packet transmission, $T_{on}$, is equal to the duration of time slot, $T_s$, then the probability of collision is at its peak, which is equal to the case of 2-packet overlapping. When $T_{on}$ is equal or less than half of $T_s$, the probability of collision reduces to be equal to or lower than the probability shown in Eq (2.11). This coincides with the fact that the shorter the packet duration, the lower its probability of collision is.

### 2.3.2 Self-interference and frequency division strategy

Assume that $N_p$ collocated piconets hopping randomly over $M$ frequencies could be re-arranged equally into $N_G$ groups, so that $\frac{N_p}{N_G}$ piconets operate over smaller spectrum of $\frac{M}{N_G}$ frequencies. Such strategy is referred to as the frequency division strategy. In this case, the probability of collision for $T_{on} \geq T_{off}$ becomes:

$$P_{\text{col, si, } N_p \text{ piconets, } N_G \text{ groups}, \ T_{on}\geq T_{off}} = \frac{T_s - (T_{on} - T_{off})}{T_s} \times \left[ 1 - \left(1 - \frac{1}{M}\right)^{N_p-1} \right]$$

$$+ \frac{(T_{on} - T_{off})}{T_s} \times \left[ 1 - \left(1 - \frac{1}{M}\right)^{\frac{N_p}{N_G}-2} \right] \quad (2.13)$$

Figure 2.29 plots the probability of collision for the frequency division strategy for different values of $N_p$ and $N_G$, where $M = 79$ and $T_{on} = 366\mu s$. If the number of collocated piconets is less than or equal to the number of groups, the scenario is essentially collision-free. If the number of collocated piconets is greater than the number of available groups, the frequency division strategy is still able to reduce the probability of collision. It is important to note that the higher the number of frequency groups, the lower the probability of collision. Due to its efficiency against self-interference, the frequency division strategy is applied for coexistence mechanisms discussed in the next chapter.

### 2.3.3 Collision induced by frequency-static interference

Consider a frequency-static interferer $\Gamma$ that operates over a set of $M_\Gamma$ channels out of $M$ frequencies used by $\pi_1$. We assume that the frequency-static interferer is always present...
2.3 Probability of packet collision for Bluetooth piconets

![Dependency of collision probability on number of frequency groups (M=79, T_{on}=366µs)](image)

Figure 2.9: Dependency of collision probability on number of frequency groups

and its set of frequencies does not change with time. Therefore, \( \pi_1 \) will experience collision due to \( \Gamma \) whenever it selects a frequency that is belong to the frequency set of \( \Gamma \). The probability of collision due to \( \Gamma \) is given by:

\[
P_{\text{col}, \text{si}} = \frac{M_\Gamma}{M}
\]  \hspace{1cm} (2.14)

Since the collision events due to frequency-static and self-interference are independent from each other, the general probability of collision can be expressed as follows:

\[
P_{\text{col}, N_p \text{ piconets}} = 1 - \left( 1 - P_{\text{col}, \text{si}} \right) \times \left( 1 - P_{\text{col}, \text{si}, N_p \text{ piconets}} \right)
\]  \hspace{1cm} (2.15)

where \( P_{\text{col}, \text{si}, N_p \text{ piconets}} \) is the probability of collision induced by self-interference, which can take two forms as in Eq. (2.10) and (2.12). We observe that the probability of collision in Eq. (2.15) depends on several factors, namely the number of collocated piconets \( (N_p) \), the packet duration \( (T_{on}) \), the time slot \( (T_s) \), the number of available hopping channels \( (M) \) and the size of static-interference spectrum \( (M_\Gamma) \). For example, when there are more collocated piconets, the level of interference will, of course, be higher. If the total number of hopping channels decreases, the probability of collision will also increase for the same amount of collocated piconets. In addition, if the size of static-interference spectrum is large, it will make the probability of collision considerably higher.
2.3.4 Frequency-static interference model

As discussed in section 1.3.2 we consider IEEE 802.11b DSSS WLAN as the FSI model in this project. The DSSS PHY uses a 22MHz channel and may support up to three non-overlapping channels or seven overlapping channels in the unlicensed ISM band [1], Figure 2.10 shows the 802.11b channel allocation for Europe.

![Figure 2.10: Bluetooth channel allocation in comparison with the IEEE 802.11b DSSS spectrum](image)

We will pick the 802.11b DSSS non-overlapping channel allocation to demonstrate the effect of FSI. In this case, there are three possible regions for interference with Bluetooth, denoted as region I, II and III in Figure 2.10. The sizes of interfering spectrum, \( M_f \), are 22, 23 and 20 channels, respectively. In the next chapter, we are going to adapt the design of the proposed coexistence mechanisms for these interfering regions. Note that the same design principles can be applied for other shapes of interfering spectrum.

2.4 Summaries

This chapter provides an overview of the Bluetooth wireless technology, focusing on its RF and baseband layers. Bluetooth devices operate at the 2.4GHz unlicensed ISM band,
2.4 Summaries

where their frequencies hop pseudo-randomly over 79 channels. The hopping sequence is derived from the master’s address and clock, and it is shared among all Bluetooth devices of the same piconet.

The baseband layer employs TDD mechanism to facilitate collision-free communication between a master and its slaves. There are two basic types of physical link between the master and slaves, namely ACL and SCO. The ACL link acts like a packet-switched connection, while the SCO provides a circuit-switched connection for voice services. Bluetooth devices can transmit packet of 1, 3 and 5 slots long. The longer the packet size, the higher the data efficiency, but also the lower the reliability. Depending on packet type, FEC or ARQ or both schemes can be employed to protect the information.

Control information can be exchanged between a master and its slave(s) by two different approaches. Unicast is more reliable, but it might consume a large portion of system throughput. Broadcast can be faster in certain situation, but it does not guarantee that information has been passed successfully to recipient(s). Usually, a broadcast message is repeated for several times to achieve reasonable reliability.

In this chapter, we have derived the theoretical probability of collision for Bluetooth piconet. The probability is governed by the number of collocated piconets, the packet duration, the time slot, the number of available hopping channels, as well as the size of spectrum interfered by FSI. We also study the efficiency of frequency division strategy against self-interference, and this strategy will be used later on for coexistence mechanisms.
Coexistence mechanisms

3.1 Adaptive Frequency Hopping (AFH)

3.1.1 Overview of AFH

The AFH is defined in the IEEE 802.15.2 recommendation and the Bluetooth specification version 1.2 as a non-collaborative mechanism to enable the coexistence of Bluetooth devices with frequency-static devices in the unlicensed band, such as IEEE 802.11b WLAN. The AFH mechanism dynamically changes the frequency hopping sequence in order to avoid or minimise the interference seen by the Bluetooth device. The AFH only works in the Connection state, where a connection has been established between the master and the slave. The operation of AFH can be divided into 3 main parts, which are discussed hereafter.

Channel classification

The purpose of channel classification is to determine the quality of each frequency channel with respect to interference. A high-interference channel is classified as a “bad” channel, while an interference-free (or low-interference) channel is marked as “good” channel. There are several classification methods recommended in [3], which may be used separately or together:

- Classification based on RSSI: The RSSI is often used to evaluate the channel condition and thus it can be employed to classify the channel. For example, if the RSSI is high and a packet error is detected, the channel is likely to suffer from interference. Alternatively, Bluetooth device may monitor RSSI in idle time slots. The average RSSI for each channel is recorded over time. If this value is higher than a pre-defined threshold, the device can declare such channel as “bad”.

The advantage of RSSI classification is that it is possible to distinguish interference from channel propagation error, hence making classification more accurate. However, it may be more complicated and consume higher power than the other classification methods.
Classification based on PER: The quality of a particular frequency channel can be determined by the PER. A packet is considered lost if Bluetooth device fails to synchronize the access code, or encounters HEC or CRC error [3]. By measuring the rate of erroneous packets to received packets, it is possible to compile a list of PERs for all hopping channels. Periodically, Bluetooth device checks these PERs against a pre-defined threshold. The channel is denoted as “bad” if its PER is too high.

The advantage of the classification method based on PER is that it is simple and can be completely implemented at the link layer. However, it alone cannot directly distinguish whether the “bad” channel is due to interference or some other channel adverse conditions. Therefore, it is recommended that the PER should be used in conjunction with other method(s) to better serve the coexistence mechanisms.

Classification based on carrier sensing: The carrier sensing could be used to determine if particular spectrum is occupied by WLAN, such as the IEEE 802.11b. Periodically, the Bluetooth device listens to the channel for 802.11b signal. If the signal is present, a set of spectrum used by the signal will be marked as “bad”.

The carrier sensing method requires at least a simplified version of 802.11b receiver to perform the clear channel assessment procedure, making Bluetooth device more complex. Furthermore, it will not be able to detect other types of interferences, except the IEEE 802.11b.

The channel assessment procedure could be done at the master side or at the slave side. The master may integrate the channel classification information returned from its slaves. Once the master has compiled the list of “good” and “bad” channels, it will distribute the list to all of its slaves via LMP commands and enable the adaptive hopping mode at slaves if this has not been done before.

It is important to note that the Bluetooth specification and the IEEE 802.15.2 standard do not specify the channel classification procedure in details. It is up to specific vendor to decide the channel assessment method(s), as well as threshold(s) for classification.

Modification of hopping sequence

In adaptive hopping mode, a piconet uses the adapted channel hopping sequence, instead of the basic one, for communication. The adapted channel hopping sequence is derived from the basic channel hopping sequence, using the diagram shown in Figure 3.3.

The AFH channel map input is the list of “good” and “bad” channels, which indicates which frequency channels should be used and which should be unused. Due to government regulations for the 2.4GHz unlicensed band, the Bluetooth specification requires
3.1 Adaptive Frequency Hopping (AFH)

![Diagram of AFH setup](image)

Figure 3.1: The Bluetooth adaptive frequency hop selection kernel

that the minimum number of used channels be $N_{\text{min}} = 20$. That means the AFH scheme can remove maximum 59 “bad” channels from the original 79 frequencies.

The *AFH re-mapping function* is used to re-map the unused RF channel to one of those used channels. If the RF channel determined by the basic frequency hop selection kernel is already in the set of used RF channels, no adjustment will be made. As a result, the basic channel hopping sequence is identical to the adaptive channel hopping sequence on all locations where the used RF channels are present. This property is an important advantage of AFH scheme, because it facilitates non-AFH slaves remaining synchronised while other slaves in the piconet are in the AFH mode. Detailed information on the AFH re-mapping function can be obtained at [2].

Another feature of the AFH mode is the “same channel mechanism”. In basic hopping mode, slaves respond using a different frequency channel other than the one used by the master (see section 2.2). With AFH, however, both master and slave communicate over the same frequency channel in any slot pairs. This is done to avoid instances where the master transmits on a “good” channel while the slave responds on a “bad” one (or vice versa), as this would lead to several retransmissions [12]. Furthermore, the “same channel mechanism” is beneficial for channel assessment, because essentially it makes channel classification at the slaves redundant. By employing the built-in Acknowledgement (ACK) mechanism, the master can record the PER level experienced by itself and its slaves on any specific hopping channel. The receipt of a packet with the Negative Acknowledgement (NACK) bit set in the header indicates that the previous packet sent by the master was lost. Similarly, the master can conclude that a packet error has happened at either the uplink or the downlink, if it does not receive the expected ACK. Owing to the “same channel mechanism”, in both cases, the master can determine the quality of that hopping channel.

Since the master and the slave now transmit on the same frequency, the channel hopping rate is reduced by 50 percent to just 800 hops per second. While this can render Bluetooth devices more sensitive to channel fading and FSI, the benefit derived from doing so far outweighs this minor drawback [12].

Similar to section 2.3.1 we can derive the theoretical probability of collision induced by self-interference for the same channel method. If $T_{\text{on}}$ is greater than or equal to $T_{\text{off}}$, 


then the probability is given by:

\[
P_{\text{coex, si}, N_p, \text{piconets}, T_{on} \geq T_{off}} = \frac{2T_s - (T_{on} - T_{off})}{2T_s} \times \left[1 - \left(1 - \frac{1}{M}\right)^{N_p-1}\right] + \frac{(T_{on} - T_{off})}{2T_s} \times \left[1 - \left(1 - \frac{1}{M}\right)^{2N_p-2}\right]
\]

(3.1)

If \(T_{on}\) is less than \(T_{off}\), the probability of collision for the same channel method is exactly the same with the probability in Eq. (2.12):

\[
P_{\text{coex, si}, N_p, \text{piconets}, T_{on} < T_{off}} = \frac{T_s - (T_{off} - T_{on})}{T_s} \times \left[1 - \left(1 - \frac{1}{M}\right)^{N_p-1}\right]
\]

(3.2)

We note that the probabilities of collision in Eq. (3.1) and (3.2) depend on the total number of available hopping channels \(M\). As an AFH-enabled Bluetooth piconet reduces its hopset to avoid frequency-static interference, it actually increases the chance of having collisions due to self-interference.

**Channel recovery**

Due to the unpredictability of the wireless medium, the AFH mechanism is required to periodically re-evaluate the quality of all channels. This is important as the interfering source, such as IEEE 802.11b, may have turned off or appeared somewhere else since the last channel assessment. If the interference is no longer there, it is desirable for Bluetooth piconets to expand their hopsets, so as to minimise the level of self-interference between themselves. A good channel recovery scheme should allow the piconet to reuse spectrum that has been marked “bad” whenever it becomes “good” again.

A procedure for channel recovery is left open for the specific vendor to determine. The Bluetooth specification only defines the tool for channel recovery, which is LMP commands for updating the AFH channel map.

### 3.1.2 Implementation of the reference AFH mechanism

In this section, we propose a variant of the AFH mechanism, which is going to act as a reference for evaluating other coexistence methods in the next section. This variant bases solely on the average PER to classify the frequency channels.

A piconet starts with hopping pseudo-randomly over the full hopset (i.e. 79 channels). For a duration of \(T_u^{AFH}\), the piconet maintains records for the number of in-error packets and the number of received packets to calculate the PER statistics of every channel on which it operates. The duration \(T_u^{AFH}\) is referred to as AFH updating interval.
After each AFH updating interval, the master removes “bad” channels from the current hopset based on the collected PER statistics. The channels are sorted in order of their PER. “Bad” channels with PER exceeding threshold \( P_{AFH} \) are removed, until all “bad” channels are removed or the minimum number of used channels, \( N_{\text{min}} \), is reached. The parameter \( P_{AFH} \) is referred to as the \( \text{AFH removing threshold} \). The final list of used channels is distributed to all of the slaves. The distribution method can be unicast, which requires up to 14 successful packets for maximum of 7 active slaves (i.e. for each slave, one packet is for AFH channel map updating command and the other for slave acknowledgement). The updating of AFH channel map costs a portion of available system throughput, and therefore should be done only when necessary. To save the throughput, the master may not send any AFH channel map updating command unless the channel list has changed between two updating interval. By default, the slaves continue to hop over the current hopset if no updating command is issued from the master.

After updating process, the piconet starts to hop over new hopset for another AFH updating interval. Again, it keeps record of the PER for all channel in the current hopset, so that it could decide which channel should be removed at the end of updating interval.

The removed channel shall be reused after a period of \( T_{AFH} \). This is denoted as \( \text{AFH reuse period} \). Each removed frequency is associated with a timeout counter. This counter is set to be equal to \( T_{AFH} \) at the moment the frequency is removed. The counter decreases over time, and when it reaches zero, that frequency can be added to the current hopset for re-usage. To avoid sending too many AFH channel map updating command, the reused frequency will be included to the current hopset at the following channel updating event.

As we can see, the AFH updating interval, \( T_{AFH} \), should be long enough to make the collected PER statistics meaningful and to avoid frequent exchange of channel updating commands. On the other hand, it should not be too long, which make the algorithm less robust against the variation of wireless medium. The AFH reused period, \( T_{AFH} \), depends on the rate at which the FSI is changing. For example, if the interference spectrum does not change with time, a long AFH reused period is more advantageous than a short one, since the piconet tries to reuse the “bad” channels less often.

### 3.2 Dynamic Adaptive Frequency Hopping (DAFH)

In [19], a non-collaborative coexistence mechanism, termed DAFH, is proposed for avoidance of frequency-dynamic self-interference among collocated WPANs. The idea behind DAFH algorithm is to make collocated piconets dynamically allocate themselves into orthogonal hopsets, so that they do not interfere each others. The algorithm uses only
PER measurement as its input, and does not differentiate whether the interferer is another WPAN or a non-WPAN entity. Therefore, although it is designed to combat self-interference, the DAFH is inherently immune from FSI. In this section, we explain the fundamental operation of DAFH mechanism, and propose necessary modification to adapt the original scheme to the IEEE 802.11b interference.

### 3.2.1 Customisation of DAFH superset

The first and also the most essential step of the DAFH algorithm is to define the superset, which consists of all possible hopping sets that can be used by the piconets. In general, hopsets in the superset are divided into $L$ levels, and at the $l^{th}$ level there are $M_l$ hopsets. The $k^{th}$ hopset at level $l$ is denoted by the admissible utilisation vector $u(l, k)$, where $l = 0, 1, \ldots L$ and $k = 0, 1, \ldots M_l$. The minimal allowable size of a hopset is $N_{\text{min}}^{\text{DAFH}}$.

#### The original DAFH superset

In [19], a DAFH superset is defined using binary division strategy. Figure 3.2 illustrates this principle: One hopset at level $l_1$ is divided into left and right half to produce two other hopsets at level $l_2 = l_1 + 1$. For example, level 1 has two hopsets, $u(1,0)$ and $u(1,1)$, which are derived from the hopset $u(0,0)$ of level 0. In the original superset, the number of hopset at level $l$ is $M_l = 2^l$.

![Figure 3.2: The original DAFH superset](image)

Table 3.1 shows all hopsets in the original superset in details. It is important to note that the sizes of the hopsets are not equal, due to the fact that only 79 frequencies are available. The minimal allowable size of a hopset, in this case, is 4.
### Table 3.1: A original superset of DAFH mechanism

<table>
<thead>
<tr>
<th>Level</th>
<th>Hopset</th>
<th>Channels</th>
<th>Number of Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 0</td>
<td>u(0, 0)</td>
<td>0 ... 78</td>
<td>79</td>
</tr>
<tr>
<td>Level 1</td>
<td>u(1, 0)</td>
<td>0 ... 39</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>u(1, 1)</td>
<td>40 ... 78</td>
<td>39</td>
</tr>
<tr>
<td>Level 2</td>
<td>u(2, 0)</td>
<td>0 ... 19</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>u(2, 1)</td>
<td>20 ... 39</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>u(2, 2)</td>
<td>40 ... 59</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>u(2, 3)</td>
<td>60 ... 78</td>
<td>19</td>
</tr>
<tr>
<td>Level 3</td>
<td>u(3, 0)</td>
<td>0 ... 9</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 1)</td>
<td>10 ... 19</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 2)</td>
<td>20 ... 29</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 3)</td>
<td>30 ... 39</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 4)</td>
<td>40 ... 49</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 5)</td>
<td>50 ... 59</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 6)</td>
<td>60 ... 69</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>u(3, 7)</td>
<td>70 ... 78</td>
<td>9</td>
</tr>
<tr>
<td>Level 4</td>
<td>u(4, 0)</td>
<td>0 ... 4</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 1)</td>
<td>5 ... 9</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 2)</td>
<td>10 ... 14</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 3)</td>
<td>15 ... 19</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 4)</td>
<td>20 ... 24</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 5)</td>
<td>25 ... 29</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 6)</td>
<td>30 ... 34</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 7)</td>
<td>35 ... 39</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 8)</td>
<td>40 ... 44</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 9)</td>
<td>45 ... 49</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 10)</td>
<td>50 ... 54</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 11)</td>
<td>55 ... 59</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 12)</td>
<td>60 ... 64</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 13)</td>
<td>65 ... 69</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 14)</td>
<td>70 ... 74</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>u(4, 15)</td>
<td>75 ... 78</td>
<td>4</td>
</tr>
</tbody>
</table>
The modified DAFH superset

The superset introduced in [19] is not designed with FSI in mind, which might lead to inefficient usage of available spectrum if the interference like IEEE 802.11b DSSS is present. As we can see from Figure 3.2 if the interference is located in region II, it might affect both hopsets at level 1, making the DAFH-enabled piconet unable to stay at that level. In this section, we are going to propose a customised superset for DAFH, which aims at eliminating such inefficiency. We select the 802.11b non-overlapping channel allocation shown in Figure 2.10 as an example of FSI spectrum, but the same design principles can be applied for any other interference spectrum patterns.

![Diagram of the customised DAFH superset for 802.11b non-overlapping channel allocation](image)

Figure 3.3 illustrates three possible regions of 802.11b interference and the customised DAFH superset. Level 0 has only one hopset, $u(0,0)$, which covers all 79 hopping frequencies. Level 1 consists of two hopsets, $u(1,0)$ and $u(1,1)$, which are basically two halves of the hopset $u(0,0)$. These hopsets are specially designed so that if interference is present at either region I, II or III, at least one hopset is interference-free. In similar manner, hopsets of level 2, 3 and 4 are formed by halving the hopsets of preceding levels. We restricted the minimal allowable size of a hopset, $N_{DAFH}^{\text{min}}$, to 5. This choice is made so that we can accommodate as many non-interfering piconets as possible, while making the hopset relatively robust against interference. As a result, level 4 is the lowest level, as its hopsets are too small for further division. Due to the same reason, hopsets $u(3,2)$ and $u(3,3)$ at level 3 are not divided. These hopsets, together with hopsets at level 4, are referred to as non-divisible hopsets. Details of the proposed superset is shown in Table 3.2.

There are only 12 hopsets at the level 4 compared to 16 of the original superset in [19]. Therefore, the number of collocated piconets that this superset can accommodate is expected to reduce.
### Table 3.2: A proposed superset for DAFH mechanism

<table>
<thead>
<tr>
<th>Level</th>
<th>Hopset</th>
<th>Channels</th>
<th>Number of Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 0</td>
<td>( u(0,0) )</td>
<td>0 ... 78</td>
<td>79</td>
</tr>
<tr>
<td>Level 1</td>
<td>( u(1,0) )</td>
<td>0 ... 28 and 52 ... 58</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>( u(1,1) )</td>
<td>29 ... 51 and 59 ... 78</td>
<td>43</td>
</tr>
<tr>
<td>Level 2</td>
<td>( u(2,0) )</td>
<td>0 ... 21</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>( u(2,1) )</td>
<td>22 ... 28 and 52 ... 58</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>( u(2,2) )</td>
<td>29 ... 51</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>( u(2,3) )</td>
<td>59 ... 78</td>
<td>20</td>
</tr>
<tr>
<td>Level 3</td>
<td>( u(3,0) )</td>
<td>0 ... 10</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>( u(3,1) )</td>
<td>11 ... 21</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>( u(3,2) )</td>
<td>22 ... 28</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>( u(3,3) )</td>
<td>52 ... 58</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>( u(3,4) )</td>
<td>29 ... 40</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>( u(3,5) )</td>
<td>41 ... 51</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>( u(3,6) )</td>
<td>59 ... 68</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>( u(3,7) )</td>
<td>69 ... 78</td>
<td>10</td>
</tr>
<tr>
<td>Level 4</td>
<td>( u(4,0) )</td>
<td>0 ... 5</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>( u(4,1) )</td>
<td>6 ... 10</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( u(4,2) )</td>
<td>11 ... 16</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>( u(4,3) )</td>
<td>17 ... 21</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( u(4,4) )</td>
<td>29 ... 34</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>( u(4,5) )</td>
<td>35 ... 40</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>( u(4,6) )</td>
<td>41 ... 46</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>( u(4,7) )</td>
<td>47 ... 51</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( u(4,8) )</td>
<td>59 ... 63</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( u(4,9) )</td>
<td>64 ... 68</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( u(4,10) )</td>
<td>69 ... 73</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>( u(4,11) )</td>
<td>73 ... 78</td>
<td>5</td>
</tr>
</tbody>
</table>
3.2.2 Operation of DAFH mechanism

The operation of DAFH-enabled piconet begins with hopping pseudo-randomly over the full hopset, which is \( u(0,0) \). The piconet maintains records of the number of in-error packets and the number of received packets to calculate the PER statistics on the current hopset. Let us denote the received packet counter as \( \text{TotalCount} \), and the erroneous packet counter as \( \text{ErrorCount} \). When the \( \text{ErrorCount} \) reaches a pre-defined value of \( \text{MAX\_ERRORS} \), the piconet starts to estimate the average PER:

\[
P_e = \frac{\text{MAX\_ERRORS}}{\text{TotalCount}}
\]

and resets both \( \text{ErrorCount} \) and \( \text{TotalCount} \) to zero. If \( P_e \) is greater than the current threshold \( P_t^{DAFH}(l) \), then the piconet is triggered to avoid the source of error by selecting a smaller hopset. \( P_t^{DAFH}(l) \) is referred to as the \textit{triggering threshold} at the \( l \)th level of the superset. By limiting the \( \text{ErrorCount} \) instead of \( \text{TotalCount} \), the PER is always estimated with the same reliability, while the piconet tends to faster leave the hopsets that are characterised with high PER [19].

Once the piconet is triggered due to high PER, it selects either left or right half of the current hopset, if available. This is referred to as the \textit{binary hopset reduction}. In case the current hopset cannot be split further in left and right halves due to \( N^{DAFH} \) restriction, the piconet shall pick randomly another hopset from the list of non-divisible hopsets, which is called \textit{the random-pick hopset reduction}. The same procedure is repeated until the piconet reaches a hopset in which PER is less than the triggering threshold.

In DAFH, the piconet should work with a reduced hopset only if there is interference at the other frequencies, either from frequency-static interferer or from other collocated piconets. Therefore, a doubling mechanism is introduced, which allows piconets to regain frequencies that were excluded during the reduction process. Consider the current hopset is \( u(l_1, k_1) \), where \( l_1 > 0 \). If the piconet is not triggered for a \( T_d^{DAFH} \) period, then it will double its hopping frequencies by selecting new hopset \( u(l_2, k_2) \), where \( l_2 = l_1 - 1 \) and \( k_2 \) is chosen uniformly from \( \{0, 1, \ldots, M_{l_2}\} \). The \( T_d^{DAFH} \) is referred to as the \textit{DAFH doubling period}. After doubling, the piconet is required to remember the previous set, \( u(l_1, k_1) \). If the PER is greater than the current threshold when \( \text{ErrorCount} \) reaches \( \text{MAX\_ERROR} \) for the first time, the piconet should go back to the "last know good" state, which is \( u(l_1, k_1) \). If it is not triggered at the first time, the piconet "forgets" \( u(l_1, k_1) \) and proceeds with the usual randomised reduction, as mentioned above.

When a hopset change occurs due to either triggering or doubling, the master informs its slaves via unicast. The hopset updating process requires at most 14 successful packets for a piconet with 7 slaves (one information packet plus acknowledgement per each slave). This amount of overhead might significantly reduce the available throughput of the piconet. Therefore, a piconet should only be triggered when interference exists.
Since PER measurement is the only input, DAFH algorithm cannot differentiate between packet errors induced by channel noise and those due to interference. To prevent unnecessary reduction due to channel noise, the threshold $P_t^{DAFH}(l)$ should be equal to or higher than the estimated maximal probability of packet error induced by channel noise, $\hat{p}_n$ [19].

Regarding the selection of doubling period, if $T_d^{DAFH}$ is too short, then the effect of the orthogonalisation by reduction of hopset will be hindered and a large overhead will be introduced. On the other hand, $T_d^{DAFH}$ should not be too long, so that the piconet could quickly expand its hopset when the interference previously caused reduction of the hopset is no longer present.

### 3.2.3 Variants of DAFH mechanism

There are two variants of the DAFH mechanism, namely DAFH with Constant Threshold (DAFH-CT) and DAFH with Adaptive Threshold (DAFH-AT). The DAFH-CT has the same triggering threshold for all levels, that means:

$$P_t^{DAFH-CT}(l_i) = P_t^{DAFH-CT}(l_j) \quad \forall l_i, l_j$$

(3.4)

where $P_t^{DAFH-CT}(l_i)$ is threshold at the level $l_i$. When triggered, the DAFH-CT chooses the new hopset uniformly between the left and right half of the current hopset [19].

On the other hand, the DAFH-AT employs adaptive triggering threshold for each level, as shown in Eq. (3.5). The rationale for this adaptive threshold is that a piconet at level $l$ should not be triggered by less than $M_l$ piconets at level 0, since there are enough hopsets at level $l$ to accommodate $M_l$ piconets, including itself [19]. Therefore, the piconet should not reduce its hopset, but let the others do.

$$P_t^{DAFH-AT}(l) = 1 - \left(1 - P_t^{DAFH-AT}(0)\right)\left(1 - \frac{1}{M}\right)^{2(M_l-1)}$$

(3.5)

In addition, the adaptive threshold helps DAFH to gain immunity towards the errors from channel noise. In case of DAFH-CT, if the threshold is below the PER level due to the channel noise, denoted by $p_n$, the piconet is continuously triggered and introduces unnecessary overhead which reduces throughput. For DAFH-AT, the smaller the hopset, the higher the threshold and hence the more difficult it is for a piconet to be triggered. Hence, it is likely that the piconet will find a level $l_1$ for which $P_t^{DAFH-AT}(l_1) > p_n$ and settle there.

Another feature for the DAFH-AT is the biased random selection. As discussed earlier, DAFH-CT chooses uniformly the left or right half of the current hopset, if triggered. This random selection is an excellent choice if the cause of packet errors is other collocated piconets with the same hopset, because the probability of interfering piconets to select
the same half is minimal. However, if the FSI or other collocated piconets with smaller hopsets is the problem, then choosing the half with less errors would be a wiser choice. The DAFH-AT employs biased random selection to account for both situations. The probability to select the left half of the current hopset upon triggering is [19]:

\[ p_L = 2^{-\left(\frac{L}{R_e}\right)\alpha} \]  

(3.6)

where \( L_e \) and \( R_e \) are the number of packet errors at the left and the right half of the current hopset, respectively. \( \alpha \) is bias selection parameter. When \( \alpha \rightarrow 0 \), the probability \( p_L \rightarrow \frac{1}{2} \) and the selection process is unbiased by the ratio \( \frac{L}{R_e} \). When \( \alpha \rightarrow \infty \), the Eq. (3.6) represents the deterministic reduction (i.e. the piconet always selects the half with less packet errors).

3.2.4 Advantages and disadvantages of DAFH

Working as a non-collaborative method, the DAFH mechanism does not require interfering entities to explicitly exchange information to achieve mutual coordination. By letting collocated piconets dynamically allocate themselves into orthogonal hopsets, the DAFH algorithm succeeds in avoiding the self-interference and inherently avoids FSI. In [19], the DAFH mechanism is shown to be able to reduce the PER due to self-interference and to drastically improve the throughput performance, compared to the case where no coexistence mechanism is applied.

Nevertheless, there are several obstacles for implementation of the DAFH mechanism. First, the FCC regulation for the wireless applications operating under ISM unlicensed band requires that [27]:

Frequency hopping systems in the 2400-2483.5 MHz band shall use at least 15 non-overlapping channels. The average time of occupancy on any channel shall not be greater than 0.4 seconds within a period of 0.4 seconds multiplied by the number of hopping channels employed. Frequency hopping systems which use fewer than 75 hopping frequencies may employ intelligent hopping techniques to avoid interference to other transmissions. Frequency hopping systems may avoid or suppress transmissions on a particular hopping frequency provided that a minimum of 15 non-overlapping channels are used.

It is clear that the hopset reduction in DAFH is not in strict agreement with the current regulation for the frequency hopping system in the 2.4GHz unlicensed spectrum, although such situation could be justified if we consider the interference induced by a group of collocated DAFH-enabled piconets (referred to as a collective entity) to other
devices in the unlicensed spectrum [19]. Furthermore, the hop-set reduction minimises the robustness of frequency hopping systems. A small hop-set could fall completely under an interference spectrum, resulting in lost connection between the master and its slaves. This is referred to as the “deadlock” situation.

Secondly, the DAFH mechanism requires certain information about the wireless medium in which the device is operating, such as the estimated maximal probability of packet error due to channel noise or the shape of FSI spectrum. The first is needed for obtaining appropriate threshold, while the latter is for selecting an optimum superset. To obtain such knowledge, other channel assessment mechanisms, such as carrier sensing or RSSI, are required in conjunction with PER measurement.

3.3 Frequency Rolling (FR)

As mentioned above, the DAFH algorithm does not strictly conform to the FCC rules for ISM band. In this section, we discuss the FR coexistence mechanism, which aims at avoiding the self-interference among collocated piconets while satisfying the regulation.

3.3.1 Operation of FR mechanism

The FR mechanism is proposed in [18]. Similar to DAFH, the algorithm uses only PER statistics for making decisions, and it cannot distinguish the packet errors due to channel noise with those induced by self-interference or frequency-static interference. However, FR manages to achieve orthogonisation of hopsets of the collocated piconets in a different way compared to DAFH. The effect of FR in the long run is that the collocated piconets use all available channels in an implicit time-division manner.

At time slot $t_0$, a FR-enabled piconet starts to operate over a subset of 79 available frequencies. The subset is referred to as current hopset, and it can be completely described by the starting frequency $f_s(t_0)$ and its size $H$ (see Figure 3.4). In a small time interval, called rolling interval $T_{r1}^{FR}$, the piconet picks channels pseudo-randomly from the current hopset. During each rolling interval, the piconet records the number of received packet and the number of erroneous packet for PER calculation.

At the end of a rolling interval, the PER measurement is compared to a pre-defined jump threshold $P_{j}^{FR}$. If PER is lower than the jump threshold, the piconet changes the current hopset by shifting it $h_s$ step to the right. The mechanism is referred to as rolling. Note that the shifting is performed in modulo $M$, where $M$ is the total available hopping frequencies for Bluetooth [18]. In other words, the starting frequency of hopset for the next rolling period is given by:

$$f_s(t_0 + T_{r1}^{FR}) = \left( f_s(t_0) + h_s \right) \mod M$$  (3.7)
On the other hand, when the PER is higher than the threshold, the piconet assumes that self-interference has happened and it should find another hopset for operation. As such, the piconet shifts its hopset by \( J_k \) step to the right, where \( J_k \) is a pseudo-random integer generated by the master. This is called random jump, which aims at achieving the orthogonalisation of hopsets for the collocated piconets. Both rolling and random jump mechanisms are illustrated in Figure 3.4. The starting frequency of new hopset, in case of random jump, is calculated from the starting position of previous hopset as follows:

\[
f_s(t_0 + T_{ri}^{FR}) = (f_s(t_0) + J_k) \mod M
\]  

(3.8)

When a random jump occurs, the master must inform its slaves about the new offset \( J_k \). In [13], this is done by broadcasting the message for \( N_{BC} \) times. As discussed in section 2.2.6, value of \( N_{BC} \) depends on the target reliability and the quality of the transmission link, which is often difficult to estimate.

FR parameters, such as \( T_{ri}^{FR} \) and \( J_k \), are thoroughly designed in [13] so that it does not violate the FCC rules for \( h_s = 1 \). In addition, a necessary condition for the FR to meet FCC regulation is imposed: After a random jump, the piconet must continue with nominal rolling, without random jump, for at least 6 seconds [13]. However, this condition makes the piconet vulnerable to self-interference during that 6 seconds period, because the piconet is not allowed to perform random jump. If the step size \( h_s \) is increased, e.g. to be equal to the hopset size \( H \), the FCC regulation could be complied easier.
The original FR mechanism is designed to combat self-interference, but not FSI. In the next sections, we introduce two FR variants which are capable of mitigating both types of interference.

3.3.2 Frequency Rolling with Probing (FR-WP)

The idea behind Frequency Rolling with Probing (FR-WP) mechanism is to combine FR with AFH. While FR is effective against self-interference, AFH is very good at mitigating FSI. Generally speaking, a FR-WP enabled piconet must periodically check the spectrum for static-interference, remove all channels affected by such interference, and then perform frequency rolling only on those channels without static-interference.

In FR-WP scheme, a piconet starts to probe the channel condition for a period of $T_p^{FR-WP}$. Such probing period is designed to find and mark all channels with FSI as “bad”. During this period, the piconet always uses the basic hopping sequence, defined by Bluetooth specification, for its transmissions. The piconet keeps records of number of packet errors and number of received packets on each hopping channel for PER estimation. At the end of probing period, channels are sorted in order of their average PER. All channels having PER greater than a pre-defined threshold are marked as “bad”. The threshold is called removing threshold and denoted as $P_r^{FR-WP}$. The “bad” channels are removed, until there is none of them left or the minimum number of used channels, $N_{min}$ is reached. The final list of channels is referred to as used channels and will be employed in the next stage.

After probing, the piconet continues with a rolling period $T_r^{FR-WP}$. During the period, the piconet performs frequency rolling similar to the original FR over the list of used channels. The piconet selects channels pseudo-randomly from a subset of used channels, which is defined by a starting frequency $f_s(t)$ and its size $H$. The initial starting frequency after probing is a pseudo-random integer chosen by the master. While rolling, the piconet collects PER statistic on the current hopset. By the end of the period, such PER statistic is compared against the jump threshold $P_j^{FR-WP}$. If the PER is smaller than the jump threshold, the piconet will roll over the set of used channels with step size $h_s$ for another rolling period. The shifting operation is performed in modulo $M_u$, where $M_u$ is the number of used channels. It is possible for a piconet to stay in rolling state for several rolling interval, as long as its PER performance satisfies the jump threshold. Otherwise, the piconet decides that either self-interference or FSI has occurred in the current hopset. Hence, it returns to the probing state for one probing period before coming back to rolling. The random jump mechanism in original FR to combat self-interference has been replaced by switching back and forth between rolling and probing state in the FR-WP.

When the piconet alters its state from probing to rolling or vice versa, the master must coordinate such change with its slaves. If the piconet turns into rolling state, the list
of used channel, the initial starting frequency and the size of hopset and the rolling step size must also be distributed. Such information can be brought by unicast between master and slaves, which requires up to 14 successful packets for a piconet with at most 7 active slaves (one for information and one for acknowledgement per slave).

![State-machine diagram for the FR-WP mechanism](image)

We observe that the probing period, $T_{p}^{FR-WP}$, should be long enough to make the collected PER statistics meaningful and to avoid frequent changes of piconet state. However, it cannot be arbitrary long, because if the piconet is always under probing state, the performance gain due to rolling will be minimal. To take full advantage of frequency rolling scheme, the piconet should spend relatively longer time in rolling than in probing state. To this end, we can either prolong the rolling period $T_{r}^{FR-WP}$ or force the piconet to perform a number of rolling periods before returning to probing state. The latter could be done by keeping a counter $j$ for the number of times that PER is greater than jump threshold. The piconet shall return to probing state only when the counter has reached the maximum value of $J_c$. Otherwise, the piconet performs a random jump to avoid self-interference and increase the counter $j$ by one (see Figure 3.5). It is clear that the higher the value of $J_c$, the longer the piconet settles at rolling state. However, high value of $J_c$ causes FR-WP insensitive to the change of wireless medium.

In FR-WP scheme, we propose another modification from the original FR mechanism, which is to interleave the list of channels before rolling. By interleaving the channels, the “deadlock” situation, which occurs in the DAFH and the original FR scheme, can be avoided in FR-WP. As a FR-WP hopset is a combination of frequencies from different parts of the spectrum after interleaving, it is unlikely that all of them are blocked by FSI or channel fading. The interleaving scheme uses in this project is a simple block interleaving: All 79 available channels are divided into $N_G$ groups, which must be an even integer divisible by 80. The interleaved list of channels is created by mixing frequencies from all the odd groups with frequencies from the even groups, as shown
in Figure 3.6. In this case, the number of groups is 4. Remember that the last group always lacks one frequency, due to the fact that only 79 frequencies is available, but not 80. “Bad” frequencies, if any, are removed directly from the interleaved channel list. By interleaving, a rolling hopset of size $H$ will probably span several frequency groups, which makes it more robust against FSI and channel fading. When interleaving is used, the hopset is no longer identifiable by its starting frequency and hopset size. Instead, the starting index is used and the shifting is still performed in modulo $M_n$.

![Diagram](a) Before interleaving

![Diagram](b) After interleaving

**Figure 3.6: The block interleaving scheme with $N_G = 4$ for FR-WP mechanism**

A careful design of the FR-WP parameters is required to make FR-WP strictly conformed to FCC regulation. However, that is out of the scope of this project, and scheduled for future work.

### 3.3.3 Frequency Rolling with Frequency Group Removal (FR-FGR)

As mentioned earlier, the probing period might reduce the performance of FR-WP mechanism. In this section, we discuss the possibility of removing the probing state from FR-WP, which results in the second version of the FR mechanism.

In this version of FR mechanism, the piconet starts in rolling state over the full spectrum (i.e. 79 channels). The spectrum is interleaved using the same block interleaving scheme described in section 3.3.2. During the rolling period $T_{FR-FGR}$, the piconet collects the PER statistics on each individual frequency as well as on the whole rolling hopset. The first is referred to as the *individual PER*, while the latter is the *group PER*.

At the end of one rolling period, the piconet compares the collected group PER with a *jump threshold* $P_{j}^{FR-FGR}$. If the group PER is under the threshold, the piconet keeps on rolling, i.e. it shifts the current hopset to the right for $h_s$ step and continues the transmission for another rolling period. On the other hand, the piconet makes a random jump to another location should the PER exceed the threshold. Similar to FR mechanism, this is done to avoid possible self-interference.
In case a random jump occurs, the piconet assumes that either FSI or self-interference has happened. It should check the individual PERs to find out the nature of interference. As channels in the current hopset are interleaved, it is unlikely that all of them are affected by frequency-static interference. Therefore, when the interference is frequency static, only one or two channels in the hopset experience high level of PER. The piconet could just remove the channel(s) of which PER exceeds a removing threshold \( P_{r_{FR-FGR}} \). However, it is required to keep the minimum number of used channels to be equal to \( N_{min} \). The new list of used channels shall be sent to all slaves via unicast, together with the new starting position of the hopset. Again, this process costs up to 14 successful packets for each piconet.

The unused channels shall be reused after a period of \( T_{ru_{FR-FGR}} \), which is called the reused period. This mechanism enables these channels to be periodically re-evaluated. Each unused frequency is accompanied by a timeout counter, which is set to be equal to \( T_{ru_{FR-FGR}} \) at the moment that frequency is removed. The counter decreases with time, and when it reaches zero, that frequency is available for piconet operation. To reduce the overhead required for updating the list of used channels, the reused channels shall only be included at the next random jump event.

One problem with this mechanism is, due to slow rate of rolling, it takes a long time to assess the quality of all the frequencies. This problem can be solved by combining several frequencies into one group, and classifying the quality of the whole group. As the spectrum has been divided into \( N_G \) groups for interleaving operation, we could utilise such available groups in this algorithm: If one channel in the group is detected to have individual PER greater than the removing threshold \( P_{r_{FR-FGR}} \), the whole frequency group is removed instead of just one frequency. This is the reason for us to name this version Frequency Rolling with Frequency Group Removal (FR-FGR). The frequency-static interference, such as IEEE 802.11b, likely affects several adjacent channels. Therefore, the group removal strategy could speed up the channel assessment process, leaving more time for the piconet to operate under “good” channels.

In FR-FGR, the piconet is allowed to remove groups of channels until the minimum number of used channels, \( N_{min} \), is reached. It is also allowed to reuse the whole group after the timeout period, \( T_{ru_{FR-FGR}} \). This algorithm has not been proven to be conformed with FCC regulation. However, this is not our focus in this project and could be an interesting subject for future work.

### 3.4 Summaries

This chapter is dedicated to discuss and analyse the non-collaborative coexistence mechanisms for WPAN, namely AFH, DAFH and FR. While the AFH is effective against frequency-static interference, the DAFH and FR algorithms are designed to combat
self-interference. Based on our analysis, we have proposed several modifications of the original algorithms, which aim at mitigating both frequency-static and self-interference. In the next chapter, we are going to compare their performance and analyse the effects of their key parameters.
Simulation results

4.1 Simulation model

4.1.1 Bluetooth simulator

In order to analyse the performance of coexistence mechanisms, a C++ discrete event simulator is developed in this project. Its operation is briefly explained in this section. As mentioned in section 2.3, we assume that the distance between the devices in a piconet is small enough, so that we can represent that piconet as one entity with slotted transmission. Figure 4.1 shows possible states and events for such entity. Before the piconet entering and after leaving a hotspot, it is considered to be in inactive state. Otherwise, the piconet can be in either idle or transmitting/receiving state. In the latter, the piconet transmits and receives concurrently, i.e. the master is transmitting and one of the slaves is receiving or vice versa.

The state-machine diagram for our Bluetooth simulator is presented in Figure 4.2. For sake of clarity, we introduce several new states in this diagram. The starting point of
the piconet shall be in the inactive state. When the simulator issues a “piconet enters” event, the piconet shall initialise itself, schedule its first “clock” event and turn into the idle state. At this state, if a “clock” event is received, the piconet shall change to the transmission scheduling state. Here it shall schedule its next “clock” event. It shall also check whether the master or the slaves, depending on the current time slot is even or odd, has any data to send. If there is a packet to send, the piconet shall start the transmission by schedule two events, “Tx starts” and “Tx stops”, before coming back to the idle state. A “Tx starts” event shall bring the piconet from idle to transmitting/receiving state. A new frequency shall be selected for every packet transmission, except for the case of AFH where the “same channel mechanism” shall be applied. The piconet shall be at this state until it receives either “Tx stops” or “piconet leaves” event. If the “Tx stops” event is received, the piconet shall begin to check the received packet for error due to collision or channel impairment, which is referred to as packet processing state. If the “piconet leaves” event occurs, the piconet shall cancel all scheduled events and return to the inactive state.

4.1.2 Simulation scenario

In this project, the performance of proposed coexistence mechanisms is analysed using simulation scenario that is similar to one presented in [19] and [18]. We employ a quasi-static scenario, where piconets enter at a hotspot (i.e. lounge at airport, conference room, etc) and leave the hotspot after a random dwell time. The purpose of such simulation scenario is to measure the ability of coexistence mechanisms to adapt itself dynamically to the change of environment. We assume a Poisson process with generation
rate $\lambda$ for piconet arrivals, and an exponential distribution with the average of $T_d$ for the dwell time at the hotspot. To make the scenario quasi-static, we set a minimal dwell time, $T_{d_{\text{min}}}$, to be 20 seconds, which makes the exponential distribution shifted for 20 seconds. Using the Little’s formula, we can obtain the average number of piconets at the hotspot, $N_p$, as follows:

$$N_p = \lambda \times (T_d + T_{d_{\text{min}}})$$ (4.1)

In our simulation, we fix $T_d$ at 60 seconds, and change the average number of collocated piconets by setting different value for $\lambda$. The range for the average number of collocated piconets in this simulation is from 1 to 33. Note that scenarios in which more than 10 - 15 piconets are closely located are not likely to happen in practice, even if we consider crowded areas, such as the waiting lounges at airports or train stations. As a result, we consider 15 to be the realistic upper bound for the number of collocated piconets.

A new element is introduced in our simulation scenario, which is the presence of a IEEE 802.11b DSSS WLAN transmitter acting as the source of FSL. The transmitter occupies a portion of available spectrum, which can be either region I, II or III as shown in Figure 2.10. The interference spectrum is assumed to be unchanged over the entire simulation period.

To simplify the simulation scenario, we assume that only 1-slot packet is used by all piconets and piconets are fully loaded, i.e. they always have packets to transmit. Due to the fact that piconets enter the hotspot at random instances, their packets are not aligned (as illustrated in Figure 2.9). In particular, we choose the DH1 packet, which is not protected with FEC to increase payload size, for data transmission. This removes the need to implement the FEC in our simulation. In fact, the FEC for Bluetooth payload is proven to be ineffective against interference, as the errors caused by interference are often too many to correct [3]. The maximum packet size and payload size of the DH1 packet are 366 and 216 bits, respectively. The duration of transmitted packet $T_{on}$ is 366$\mu$s, and hence $T_{off} = T_s - T_{on} = 259\mu$s. The usage of DH1 packet results in the following maximum goodput for Bluetooth piconet:

$$S_{\text{max},DH1} = \frac{\text{Maximum DH1 payload [bit]}}{\text{Slot time [s]}} = \frac{216}{625 \times 10^{-6}} = 345.6 \text{ [kbit/s]}$$ (4.2)

We again use the assumptions stated in section 2.3 for this simulation scenario. First, only the Connection state is considered. We also restrict ourselves to implement only the ACL links. Secondly, we assume that the packet header is error-free, even in the presence of collisions. This assumption is not overly optimistic, since the packet header is quite short and heavily protected by 1/3 rate FEC [15]. As a result, slaves shall always respond in the next slave-to-master slots, since they can decode successfully their addresses in the packet headers.
A simplified wireless channel model is implemented in this scenario. All piconets in the hotspot experiences the same PER level, $p_n$, due to channel noise. Collision happens if packet overlap in both time and frequency with frequency-static interferer and/or packet(s) from collocated piconets. We consider the collision is not always destructive: The packet might not be corrupted due to collision, depending on two independent probabilities: $p_s$ and $p_d$. The first denotes the probability of having packet corrupted when FSI collision occurs, while the latter is the probability of having packet corrupted due to self-interference collision. In our simulation, these two probabilities are chosen to be 90 percent.

The simulation results in the following section are average from 10 different simulations, where each simulation lasts approximately 34 minutes, which corresponds to 3,200,000 time slots, having a slot of 625μs.

### 4.1.3 Simulation parameters for coexistence mechanisms

#### Choice of AFH parameters

Since the wireless medium is expected to be unchanged during simulation period, a long AFH updating interval is more advantageous. We select the updating interval $T_u^{AFH}$ to be 2 seconds. The AFH reused period $T_{ru}^{AFH}$ is set to be 2 times of the updating interval, or 4 seconds. We choose the AFH removing threshold $P_r^{AFH} = 70\%$, i.e. any channel experiencing PER greater than 70 percent during one updating interval shall be regarded as unusable.

#### Choice of DAFH parameters

In our simulation, we pick the number of packet errors to trigger PER calculation in DAFH, $MAX\_ERROR$, to 20. The triggering thresholds at level 0 for both DAFH-CT and DAFH-AT mechanisms are chosen to be equal to the PER due to channel noise, $p_n = 1\%$. For DAFH-AT scheme, the thresholds at the other levels are derived from the threshold at level 0 by Eq. (3.5), with number of hopsets at level $l$ is $M_l = 2^l$. Also, the bias selection parameter $\alpha$ is 4. The values of these parameters are the same with those adopted in [10].

As we do not expect the wireless medium to vary in this simulation, a long doubling period should be selected to reduce the amount of overhead information. In this case, $T_d^{DAFH}$ is equal to 6.25 seconds, or 10,000 time slots.
4.2 Result analysis

Choice of FR-WP parameters

We select the probing period $T_{p}^{FR-WP}$ to be 0.4 second, in order to minimise the probing state while still being able to collect relatively reliable PER statistics of the channels. During 0.4 second, each frequency is visited approximately 8 times on average. We also assign 0.4 second for the rolling period $T_{p}^{FR-WP}$, and 70 percent for the removing threshold is $P_{r}^{FR-WP}$ (which is essentially the same as the removing threshold for AFH mechanism). The jump threshold $P_{j}^{FRWP}$ is chosen to be 10 percent. It is higher than the PER due to the channel noise, so that the random jump will not be triggered by the channel noise.

The block interleaving scheme is performed with $N_{G} = 8$. The hopset size is 3, and on each roll it shall be shifted to the right by exactly one hop-size (i.e. $h_{s} = H = 3$). Furthermore, a piconet shall return to probing state only if it is triggered to jump for more than $J_{c} = 2$.

Choice of FR-FGR parameters

Similar to the FR-WP mechanism, we select $h_{s} = H = 3$ for FR-FGR simulation. The rolling period $T_{r}^{FR-FGR}$ is also 0.4 second, while the removing threshold $P_{r}^{FR-FGR}$ and jump threshold $P_{j}^{FR-FGR}$ are 70% and 10%, respectively.

The number of frequencies groups for interleaving and FR-FGR operation is $N_{G}^{FR-FGR} = 16$. We divide the spectrum into small groups of frequencies to make FR-FGR more spectrum efficient, i.e. it does not remove a large portion of the spectrum due to few frequencies with FSI. The reused period $T_{ru}^{FR-FGR}$ is selected to be 10 times of the rolling period, or 4 seconds.

All simulation parameters and their values are summarised in Table 4.1. Their values shall be used throughout the simulation, unless stated otherwise.

4.2 Result analysis

This section is devoted to analyse the results obtained from our simulation. The analyses are mainly based on the following performance measures:

- Normalised average goodput: The goodput, which is defined as the ratio between the total number of useful information bits (i.e. excluding overhead) received at the destination and the total time needed for the transmission, is the main parameter to be calculated and investigated in our simulation. The average goodput is obtained by averaging the goodput of all piconets which have been in the hotspot during the simulation. In order to fairly compare the performance of the systems
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Notation</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation time</td>
<td>$T_{sim}$</td>
<td>3,200,000 [slot]</td>
</tr>
<tr>
<td>Slot time</td>
<td>$T_s$</td>
<td>625 [$\mu$s]</td>
</tr>
<tr>
<td>Average number of piconets at the hotspot</td>
<td>$\bar{N}_p$</td>
<td>1 to 33</td>
</tr>
<tr>
<td>Average dwell time</td>
<td>$\bar{T}_d$</td>
<td>60 [s]</td>
</tr>
<tr>
<td>Minimum dwell time</td>
<td>$T_{dmin}$</td>
<td>20 [s]</td>
</tr>
<tr>
<td>Packet type</td>
<td></td>
<td>DH1</td>
</tr>
<tr>
<td>Maximum DH1 packet size</td>
<td></td>
<td>366 [bit]</td>
</tr>
<tr>
<td>Maximum DH1 payload size</td>
<td></td>
<td>216 [bit]</td>
</tr>
<tr>
<td>Maximum duration of packet transmission</td>
<td>$T_{on}$</td>
<td>366 [$\mu$s]</td>
</tr>
<tr>
<td>PER due to channel noise</td>
<td>$p_n$</td>
<td>1%</td>
</tr>
<tr>
<td>Probability of having packet corrupted due to FSI</td>
<td>$p_s$</td>
<td>90%</td>
</tr>
<tr>
<td>Probability of having packet corrupted due to self-interference</td>
<td>$p_d$</td>
<td>90%</td>
</tr>
<tr>
<td>IEEE 802.11b interference</td>
<td></td>
<td>Region I, II or III</td>
</tr>
<tr>
<td>AFH updating interval</td>
<td>$T_u^{AFH}$</td>
<td>2 [s]</td>
</tr>
<tr>
<td>AFH removing threshold</td>
<td>$p_r^{AFH}$</td>
<td>70%</td>
</tr>
<tr>
<td>AFH reused period</td>
<td>$T_r^{AFH}$</td>
<td>4 [s]</td>
</tr>
<tr>
<td>Number of packet errors to trigger PER calculation</td>
<td>$MAX_ERROR$</td>
<td>20 [packet]</td>
</tr>
<tr>
<td>DAFH initial triggering threshold</td>
<td>$P_t^{DAFH}(0)$</td>
<td>1%</td>
</tr>
<tr>
<td>DAFH-AT bias selection parameter</td>
<td>$\alpha$</td>
<td>4</td>
</tr>
<tr>
<td>DAFH doubling period</td>
<td>$T_d^{DAFH}$</td>
<td>6.25 [s]</td>
</tr>
<tr>
<td>FR-WP probing period</td>
<td>$T_p^{FR-WP}$</td>
<td>0.4 [s]</td>
</tr>
<tr>
<td>FR-WP rolling period</td>
<td>$T_r^{FR-WP}$</td>
<td>0.4 [s]</td>
</tr>
<tr>
<td>FR-WP removing threshold</td>
<td>$P_r^{FR-WP}$</td>
<td>70%</td>
</tr>
<tr>
<td>FR-WP jump threshold</td>
<td>$P_j^{FR-WP}$</td>
<td>10%</td>
</tr>
<tr>
<td>Number of frequency groups for interleaving in FR-WP scheme</td>
<td>$N_G^{FR-WP}$</td>
<td>8</td>
</tr>
<tr>
<td>FR-WP jump counter</td>
<td>$J_c$</td>
<td>2</td>
</tr>
<tr>
<td>Hopset size</td>
<td>$H$</td>
<td>3</td>
</tr>
<tr>
<td>Rolling step size</td>
<td>$h_s$</td>
<td>3</td>
</tr>
<tr>
<td>FR-FGR rolling period</td>
<td>$T_r^{FR-FGR}$</td>
<td>0.4 [s]</td>
</tr>
<tr>
<td>FR-FGR removing threshold</td>
<td>$P_r^{FR-FGR}$</td>
<td>70%</td>
</tr>
<tr>
<td>FR-FGR jump threshold</td>
<td>$P_j^{FR-FGR}$</td>
<td>10%</td>
</tr>
<tr>
<td>Number of frequency groups for interleaving in FR-FGR scheme</td>
<td>$N_G^{FR-FGR}$</td>
<td>16</td>
</tr>
<tr>
<td>FR-FGR reused period</td>
<td>$T_{ru}$</td>
<td>4 [s]</td>
</tr>
</tbody>
</table>
with and without coexistence mechanisms, we must account for the overhead packets that are used to convey the information about the change of hopsets among piconet members. As discussed in Chapter 3 we assume that master always employs unicast to inform its slave(s) and at most 14 successful packets are required per hopset updating operation. Hence, the goodput for coexistence mechanisms is calculated by subtracting those overhead packets from total number of successful packets. Furthermore, to ease analysis, the average goodput will be normalised with the maximum achievable goodput in Eq. (1.2).

- **Probability of collisions**: The probability of collision is estimated from simulation by the ratio between the number of collided packet and the total number of transmitted packets. It represents the level of collisions, either due to frequency-static or self-interference, in a particular scenario.

- **Frequency Occupancy**: Periodically, we calculate the probability that each frequency is used at the hotspot, by using hopset information of each piconet. The larger probability at a specific frequency means that the piconets, as a collective entity, can cause larger interference at that frequency. At each inspection interval, we pick the probability that is maximal among all the frequency. The frequency occupancy is calculated as an average of these maximal value over all simulation time. Thus, a coexistence mechanism with high frequency occupancy will have potential to cause larger interference at a particular frequency and has worse performance in terms of etiquette [19]. In our simulation, the inspection interval is every 2 time slots. The frequency occupancy is not valid for the variants of FR mechanism, as their hopsets does not occupy a fixed portion of the spectrum.

### 4.2.1 Performance of the AFH mechanism

Figure 4.3 and 4.4 represent the normalised average goodput and probability of collision for AFH mechanism, respectively. They are measured in both scenarios, without and with FSI. Performance of the standard Bluetooth, or Bluetooth piconets without any coexistence mechanism, are plotted as references. The theoretical collision probabilities for standard Bluetooth without and with FSI and for the “same channel mechanism”, which are given in Eq. (2.10), (2.15) and (3.1), are also included in Figure 4.4.

As can be seen from Figure 4.3, interference has strong impact on the performance of the standard Bluetooth piconet. The average goodput is severely degraded when the number of collocated piconets is increased and/or when a frequency-static interference is present. It is obvious that the FSI at region II produces worse goodput than that at region I, due to the fact that it occupies a larger portion of the available spectrum for Bluetooth devices.
Normalised average goodput for AFH mechanism (w/o and w/ FSI)

Figure 4.3: Normalised average goodput for AFH mechanism (without and with FSI)

Probability of collision for AFH mechanism (w/o and w/ FSI)

Figure 4.4: Collision probability for AFH mechanism (without and with FSI)
Without any FSI, AFH performs slightly better than the standard Bluetooth. This is due to the fact that the “same channel mechanism” is applied in AFH. The collision probability for the “same channel mechanism” is lower than that of the standard Bluetooth (see Figure 4.3). This figure also indicates that our simulator is working correctly. The simulation results are well-matched with the theoretical collision probabilities derived in previous sections.

In scenario where FSI exists, AFH shows better performance compared to the standard Bluetooth. The AFH average goodput is increased by 15-20 percent at small number of collocated piconets. If the number of collocated piconets is large and the self-interference is dominant, AFH mechanism becomes less effective.

It is also important to note that the performance of this variant of AFH mechanism is affected only by the size of interfering spectrum, but not its position. If the spectrums of FSI at region I and II were equal, the AFH would give the same performance in both cases. When AFH mechanism removes frequency by group as recommended in [3], different position of FSI would cause different level of goodput degradation.

![Frequency occupancy for AFH mechanism (w/o and w/ FSI)](image)

**Figure 4.5: Frequency occupancy for AFH mechanism (without and with FSI)**

Figure 4.5 shows the level of frequency occupancy for AFH mechanism in comparison with that of the standard Bluetooth. Since standard Bluetooth always operates on the full hopset of $M = 79$ channels, its theoretical frequency occupancy is given by:

$$F_{occ}(N_p) = 1 - \left(1 - \frac{1}{M}\right)^{N_p}$$  \hspace{1cm} (4.3)
where $F_{occ}(N_p)$ is the frequency occupancy of the standard Bluetooth, depending only on the number of collocated piconets $N_p$ and the size of the full hopset $M$.

In Figure 4.5, if FSI occurs at either region I or II, the frequency occupancy for AFH mechanism is higher than that of the standard Bluetooth. By reducing its hopset to avoid FSI, the AFH mechanism has put more load (or usage) onto the remaining frequencies.

On the other hand, when there is no FSI present, we observe that the frequency occupancy for AFH mechanism is very close to that of the standard Bluetooth. This is because the AFH removing threshold $P_{rAFH} = 70\%$ is high enough so that channel noise and self-interference cannot trigger AFH to reduce its operating hopset. If the channel noise and/or self-interference become significant and higher the threshold, we expect to see the frequency occupancy of AFH to increase, even if there is no collocated FSI. This problem indicates that the removing threshold $P_{rAFH}$ should be made adaptive to the operating environment, so as to avoid unnecessary hopset reduction due to channel noise or self-interference. Alternatively, another channel assessment method, such as RSSI, can be used in conjunction with PER to differentiate interference from channel noise.

### 4.2.2 Analyses of DAFH simulation results

**DAFH performance in absence of FSI**

Figure 4.6, 4.7, and 4.8 represent the simulation results for DAFH schemes under scenario where FSI is absent. Comparisons are made for DAFH-CT and DAFH-AT, and for the original and modified superset. In the figures, we denote the DAFH-AT with modified superset as MDAFH-AT. We also include the performance of standard Bluetooth for reference.

Figure 4.6 shows up to 10 percent of performance gain for Bluetooth piconets employed DAFH techniques. The gain is most obvious when the average number of collocated piconets is around 5 to 15. Due to the fact that the number of available hopsets is often greater than the number of collocated piconets in this region, it is possible for DAFH mechanism to achieve orthogonal assignment of the hopsets and minimise self-interference. Even if the number of collocated piconets exceeds number of available hopsets, DAFH mechanism continues to provide better goodput. This can be explained by the frequency division strategy discussed in section 2.3.2. By allocating collocated piconets into a number of frequency groups, the DAFH mechanism has effectively reduced the probability of collision induced by self-interference (see Figure 4.7).

However, when the number of collocated piconets becomes too large, the average goodput of DAFH will eventually be less than that of the standard Bluetooth at some point. This is the overload point for DAFH mechanism. If all frequency groups are fully
### 4.2 Result analysis

**Figure 4.6: Normalised average goodput for DAFH mechanism (without FSI)**

![Graph showing normalised average goodput](image)

**Figure 4.7: Collision probability for DAFH mechanism (without FSI)**

![Graph showing collision probability](image)
occupied, changing the hopset cannot help the DAFH-enabled piconets to reduce self-interference, but introduce unnecessary overhead packets. As we can see from Figure 4.6 the DAFH-CT goodput is reduced more quickly than DAFH-AT, while their collision probabilities are approximately equivalent. This is due to the fact that DAFH-AT uses higher threshold at higher level of the superset, therefore minimises unwanted hopset updating procedures. If DAFH is to implement, it is desirable for a piconet to detect the overload point and cease using DAFH. For instance, a piconet shall return to basic hopping sequence after several consecutive triggering.

In this scenario where FSI does not exist, the performance of DAFH-AT with the modified superset is slightly lower than the case where the original superset is used. The MDAFH-AT has fewer hopsets at level 4 than DAFH-AT, meaning that it can accommodate fewer collocated piconets.

![Frequency occupancy for DAFH mechanism (w/o FSI)](image)

**Figure 4.8: Frequency occupancy for DAFH mechanism (without FSI)**

Figure 4.8 illustrates the frequency occupancy of DAFH mechanisms. DAFH causes much higher frequency occupancy than the standard Bluetooth, as a trade-off for better goodput performance. Due to the low triggering threshold, DAFH-CT is more sensitive to interference and tends to be triggered to use smaller hopsets. As a result, its frequency occupancy is always very high, even at only small number of collocated piconets.
4.2 Result analysis

DAFH performance in presence of FSI

Similar to the previous section, the normalised average goodput, probability of collision and frequency occupancy for DAFH mechanism in presence of FSI are displayed in Figure 4.9 to 4.11.

![Graph showing normalised average goodput for DAFH mechanism (w/ FSI)](image)

**Figure 4.9: Normalised average throughput for DAFH mechanism (with FSI)**

We can observe that the DAFH mechanism works very well under the existence of both frequency-static and self-interference. For small number of collocated piconets, which often occurs in practice, the DAFH goodput is still above 90 percent, compared to only 70 percent of the standard Bluetooth. Since a significant portion of the spectrum is occupied by FSI, the DAFH performance is lower than in the case there is no FSI: The overload point for DAFH-AT falls down from 33 piconets in Figure 4.6 to 25 piconets in Figure 4.9. Otherwise, the trends in these figures are pretty much the same as those in previous section.

We can see that the MDAFH-AT is only slightly better than the DAFH-AT with the original hopset in terms of frequency occupancy (see Figure 4.11). As a result, the advantage of using modified superset is negligible and only the DAFH-AT using the original superset will be analysed in the following sections.
Figure 4.10: Collision probability for DAFH mechanism (with FSI)

Figure 4.11: Frequency occupancy for DAFH mechanism (with FSI)
4.2 Result analysis

**Effect of bias selection parameter on DAFH-AT**

As mentioned in section 3.2.3, DAFH-AT employs the biased random selection mechanism to facilitate the selection of left or right half of the current hopset. In this section, we are going to evaluate the performance of DAFH with different values of the bias selection parameter $\alpha$.

![Graph showing the effect of $\alpha$ on DAFH-AT](image)

**Figure 4.12: Effect of parameter $\alpha$ on DAFH-AT (Normalised average throughput, with FSI)**

Figure 4.12 presents the normalised average goodputs for DAFH-AT at $\alpha$ equal to 0, 4 and 10. We can see that for different values of $\alpha$, the DAFH-AT goodput performance are not much different. Due to FSI and self-interference, the DAFH-enabled piconets are often triggered to use hopsets at the lowest level of the superset. Thus, they perform the random-pick hopset reduction much more often than the binary hopset reduction, which might make the biased random selection scheme ineffective.

**Effect of error model on DAFH-AT**

In [19], the evaluation of DAFH performance was done with assumption that collision is always destructive. In this project, we generalise the error model by introducing two independent parameters to represent the packet corruption probability due to frequency-static and self-interference, namely $p_s$ and $p_d$. This section is dedicated to analysing the effect of the error model on the DAFH-AT algorithm.
Figure 4.13: Effect of packet corruption probability (Normalised average throughput, with FSI)

Figure 4.14: Effect of packet corruption probability (Probability of collision, with FSI)
The normalised average goodput and the collision probability for DAFH-AT with $p_s = p_d = 50\%$ and $p_s = p_d = 90\%$ are plotted in Figure 4.13 and 4.14, respectively. While the probability of collision is approximately the same in both cases, their average goodput differs considerably, especially for large number of collocated piconets. This is because DAFH-AT algorithm enables collocated piconets to arrange themselves into available DAFH-AT hopsets and enjoy collision-free operation. If the number of collocated piconets is larger than the number of available hopsets, then the high value of packet corruption probability starts to bring in its effect.

**Choice of DAFH-AT triggering threshold**

As discussed in section 3.2, the triggering threshold for DAFH should be equal to or greater than the estimated maximal probability of packet error induced by channel noise, so that it does not perform unnecessary hopset reduction due to channel noise. In this section, we are going to discuss the selection of DAFH-AT triggering threshold with respect to the packet error rate due to channel noise in more details.

Figure 4.15 shows the normalised average goodputs for DAFH-AT under various conditions. The channel noise $p_n$, in this section, is set to be either 1 or 30 percent. We evaluate the conditions where the DAFH-AT triggering threshold is lower, equal and greater than the channel noise. The triggering threshold $P_t^{DAFH-AT}$ takes several different values from 1 to 30 percent.

First, we consider the case where triggering threshold is equal to or larger than the PER level due to channel noise. Figure 4.15 indicates large variations between low and high values of the triggering threshold. At the same level of channel noise (e.g. $p_n = 1\%$), a lower triggering threshold performs better for small number of collocated piconets, while higher triggering threshold offers excellent goodput when the number of piconets is large. The reason is that a lower triggering threshold is more sensitive to interference, making the orthogonalisation of hopsets for collocated piconets faster. However, when all available hopsets are utilised, sensitivity to interference causes unnecessary triggering and thus reduces the goodput. On the other hand, a higher triggering threshold takes longer time to achieve orthogonal hopsets between piconets, but it can avoid unwanted triggering and sustain better goodput for greater number of piconets.

Secondly, we analyse the case where triggering threshold is smaller than the PER of channel noise. Obviously, such threshold is not desirable, as it always triggers piconet to reduce the hopset and causes severe goodput degradation. As we can see from Figure 4.15, the DAFH-AT goodput for $P_t^{DAFH-AF} = 1\%$ at $p_n = 30\%$ is much worse than that of the standard Bluetooth. Thus, correct estimation of the PER level of the channel noise is critical to DAFH. To this end, another channel classification methods, such as RSSI, must be employed in conjunction with PER measurements.
4.2.3 Analyses of FR simulation results

Performance of FR mechanisms in absence of FSI

The performance of both FR-WP and FR-FGR coexistence mechanisms are illustrated in Figure 4.16 and 4.17. In this simulation scenario, there is no FSI present. The performance is measured in terms of the average goodput and the probability of collision. Since the removing thresholds, namely $P_{r}^{FR-WP}$ and $P_{r}^{FR-FGR}$, are relatively higher than the noise floor in our simulation (i.e. 70 percent), there is hardly any frequency being removed in this scenario. As such, the FR-FGR performs slightly better than FR-WP mechanism. By always staying at rolling state, it achieves lower probability of collision, and thus having higher goodput.

In comparison with the standard Bluetooth, both FR schemes are able to maintain high goodputs for a significant number of collocated piconets, which is approximately 12 in our simulation. When the number of piconets continues to increase, their average goodputs become closer to that of the standard Bluetooth. This is because it is more and more difficult to achieve completely orthogonal hopsets when the number of collocated piconets becomes larger. In this congested situation, the rolling and random jump mechanisms in FR mechanisms result in similar behaviour to the pseudo-random frequency hopping, which leads to the almost equivalent performance between the FR schemes and the standard Bluetooth.
4.2 Result analysis

**Figure 4.16:** Normalised average goodput for FR mechanisms (without FSI)

**Figure 4.17:** Collision probability for FR mechanisms (without FSI)
In Figure 4.17 note that the collision probabilities for both FR mechanisms are still lower than that of the standard Bluetooth at 33 collocated piconets. However, due to overhead to convey the jump and frequency removal information, their average goodputs are degraded and lower than the standard Bluetooth in Figure 4.16.

**Performance of FR mechanisms in presence of FSI**

The performance of FR-WP and FR-FGR are also compared in scenario with FSI. Figure 4.18 and 4.19 are respectively the normalised average goodputs and the probabilities of collision for FR mechanisms where FSI exists at region I or II.

![Figure 4.18: Normalised average goodput for FR mechanisms (with FSI)](image)

As we can see from Figure 4.18 both modified FR mechanisms are able to mitigate the effect of FSI. They could maintain much higher goodput than that of the standard Bluetooth for a large number of collocated piconets.

However, the FR-WP outperforms the FR-FGR in this scenario, due to the fact that the FR-WP can assess the quality of all channels in relatively short time. Due to slow rate of rolling, it takes the FR-FGR much longer time to remove all “bad” frequencies due to FSI, even if the frequency group removal strategy has been applied. To make FR-FGR classifying the channels faster, we could reduce the number of frequency group $N_G$ from 16 to 8 or even 4, which increases the size of the frequency group. Nevertheless,
increasing the size leads to inefficient usage of the available spectrum, as FR-FGR might remove more frequencies than necessary for combating the FSI. Due to its superiority, we focus on the FR-WP mechanism in the next sections, which discuss the effects of different parameters on this FR scheme.

**Impact of $J_c$ on FR-WP**

In Figure 4.20, the average goodputs for FR-WP with $J_c$ of 0, 2 and 10 are plotted against each other.

Due to the fact that the FSI source is constant in this simulation scenario, the higher the value of $J_c$, the better the goodput is. However, increasing $J_c$ is a process of diminishing return, that means the highest performance gain occurs when increasing $J_c$ from 0 to 2, while $J_c = 10$ offers only a negligible gain compared to $J_c = 2$. As such, a small and non-zero value of $J_c$ would be a good choice for FR-WP, because it makes the mechanism more robust against the variations of the wireless medium, while still offers high performance gain.

**Impact of jump threshold on FR-WP**

Another parameter for FR-WP is investigated in this section, which is the jump threshold $D_J^{FR-WP}$. The jump threshold is employed to trigger a piconet to return to probing
Figure 4.20: The effect of $J_c$ on FR-WP mechanism (Normalised average goodput, with FSI @ I)

Figure 4.20 shows that the average goodput for FR-WP is degraded if a relatively low, compared to channel noise, jump threshold is used. This is because a low jump threshold could cause a piconet to falsely turn into probing state, due to the channel noise plus self-interference from other piconets at probing state. This increases the level of overhead information on the piconet, and at the same time amplifies the self-interference level in the hotspot (as this piconet, in turn, will trigger more piconets to return to probing state). The jump threshold, therefore, should be high enough, so that the piconet cannot be triggered by self-interference from one or several piconets in their probing state.

Impact of hopset size on FR-WP

Figure 4.22 indicates the effect of the hopset size $H$ on FR-WP. Simulation results for $H$ equal to 3 and 6 are shown, with the FSI source at region I.

As displayed, the hopset size $H = 6$ offers lower goodput than that of $H = 3$. Obviously, the number of piconets that can be accommodated in a constant spectrum by FR-WP depends on the hopset size. With larger hopset size, the available spectrum becomes congested more quickly, resulting in degradation in performance.
4.2 Result analysis

The effect of $P_j$ on FR−WP mechanism (Normalised average goodput, w/ FSI @ I)

Standard BT
FR−WP, $P_j = 2\%$
FR−WP, $P_j = 10\%$

Figure 4.21: The effect of $P_j$ of FR−WP mechanism (Normalised average goodput, with FSI)

The effect of $H$ on FR−WP mechanism (Normalised average goodput, w/ FSI @ I)

Standard BT
FR−WP, $h_s = H = 6$
FR−WP, $h_s = H = 3$

Figure 4.22: The effect of $H$ of FR−WP mechanism (Normalised average goodput, with FSI)
On the other hand, reducing hopset size gives up the robustness of frequency hopping scheme, causing temporally high collision probability if more than two hopsets of collocated piconets are overlapped. This might lead to disruption in communication, or more severely, disconnection.

**Impact of probing period on FR-WP**

The effect of the probing period $T_p^{FR-WP}$ on FR-WP is shown in Figure 4.23. The long probing period is selected to be 2 seconds, while the short one is 0.4 second. The rolling period is the same as in Table 4.1, which is 0.4 second.

![The effect of $T_p$ on FR-WP mechanism (Normalised average goodput, w/ FSI @ I)](image)

**Figure 4.23:** The effect of $T_p^{FR-WP}$ of FR-WP mechanism (Normalised average goodput, with FSI)

The choice of probing period has significant influence on average goodput performance of FR-WP scheme. A long probing period reduces the performance gain mainly due to two reasons: First, FR-WP-enabled piconet tends to spend more time in probing than in rolling state, which minimises the advantage of rolling mechanism. Secondly, the self-interference level in the hotspot is increased as piconets stay longer time at probing state. This might cause other piconets in rolling state to trigger, if the jump threshold is reached.

As a result, a short probing period is necessary for FR-WP to obtain the optimal goodput. However, probing duration can not be arbitrary short, as the piconet needs to
4.2 Result analysis

collect accurate PER statistic for every channels to decide the FSI spectrum. If the probing duration is too short, the piconet might miss out some of the FSI channels or falsely remove channels with temporarily high PER due to self-interference.

4.2.4 Comparison of coexistence mechanisms

Figure 4.24 and 4.25 compare the average goodputs of all mentioned coexistence mechanisms. The evaluation scenarios are without and with FSI, respectively. Simulation parameters are shown in Table 4.1.

![Comparison of coexistence mechanisms (Normalised average goodput, w/o FSI)](image)

**Figure 4.24:** Normalised average goodput of coexistence mechanisms (without FSI)

In Figure 4.24, we observe that DAFH is the most effective coexistence mechanism in terms of average goodput. It offers significant performance gain for a large number of collocated piconets. The FR-WP ranks second, while AFH does not show meaningful improvement over standard Bluetooth in the case where the FSI does not exist.

If the FSI is present, AFH begins to have effect on the goodput performance. However, DAFH and FR-WP still provides higher goodput than that of AFH for up to 15 collocated piconets. As \( N_p = 15 \) is also the upper bound for the number of collocated piconets, this indicates that the DAFH- or FR-enabled piconets are likely to enjoy faster connection than AFH-enable piconets for most of the practical scenarios.
The goodput performance of DAFH mechanism is degraded more quickly with increasing number of collocated piconets than that of FR-WP. This is due to the fact that DAFH is triggered whenever the packet error counter reaches a value of MAX_ERROR, while FR-WP is triggered at constant interval of 0.4 second. While the using packet error counter to trigger makes DAFH achieve orthogonal hopsets faster for small number of collocated piconets, it also causes DAFH to suffer a lot of overhead information when the interference is severe.

4.3 Summaries

In this chapter, we have explained our implementation of the discrete event simulator for Bluetooth wireless technology. A quasi-static scenario with the presence of IEEE 802.11b DSSS WLAN has been selected for evaluating the coexistence mechanisms mentioned in Chapter 3. In addition, we have justified our choice of parameters for the simulation. Understanding these parameters is essential to interpret what actually happens in our simulation.

We have also analysed the performance of various variants of coexistence mechanisms, namely AFH, DAFH-CT, DAFH-AT, FR-WP and FR-FGR. Their performance are measured by the normalised average goodput, probability of collision and frequency occu-
panity. In general, the AFH mechanism is shown to be effective only if the FSI is present. On the other hand, DAFH and FR are capable of combating both frequency-static and self-interference. They outperform the AFH in terms of goodput for practically large number of collocated piconets, even when FSI exists.
Chapter 5

Conclusions and future works

5.1 Conclusions

The unlicensed spectrum, which is available at no cost for almost the entire world and does not require tedious and cumbersome process of obtaining licenses, is well-suited for WPAN and WLAN applications. However, the high level of interference at the unlicensed spectrum requires these technologies to employ certain coexistence mechanisms to enable sharing of the spectrum. By “sharing” we mean that they are more robust against interference and, at the same time, causing minimal interference to other system operating in the same band.

The coexistence mechanism for WPAN in the unlicensed spectrum is the focus of this project. In particular, Bluetooth wireless technology (or equivalently the IEEE 802.15.1 standard) is chosen as an example to demonstrate the concepts of coexistence mechanisms for WPAN, due to the fact that this is the most commonly-used WPAN technology today. We concentrate on non-collaborative coexistence schemes, which do not require interfering entities to exchange data to achieve mutual coordination.

The project considers two sources of interference for Bluetooth wireless technology, namely frequency-static interference and self-interference. The first often occupies a group of frequencies for a period that is much longer than the Bluetooth packet duration. Such interference is caused by non-Bluetooth transmitter, such as a WLAN device. On the other hand, the self-interference occurs among the collocated Bluetooth piconets. Both types of interference can adversely affect the goodput performance of a Bluetooth piconet. The probability of packet collision due to either frequency-static or self-interference, which has been derived in this project, is governed by the number of collocated piconets, the packet duration, the duration of time slot, the number of available hopping channels, as well as the size of spectrum interfered by FSI.

Currently, the AFH mechanism, which has been incorporated in the Bluetooth specification version 1.2, is viewed as the best non-collaborative coexistence method by the Bluetooth SIG. This method is designed to mitigate the effect of FSI, but not self-interference. On the other hand, two novel approaches, termed DAFH and FR, are intended to deal with the problem of self-interference and have been proven to be effective. Our aims in this project are to propose necessary modifications for DAFH and FR.
mechanisms, so that they could cope with both frequency-static and self-interference. We also investigate the performance of the proposals by means of simulation. The FSI model for investigation is the IEEE 802.11b DSSS WLAN, but our discussions in this project can be extended to any kinds of frequency-static interferer and/or channel impairments, such as channel fading.

Based on our conceptual analysis and simulation results, we obtain several interesting conclusions, which will be presented in this section.

5.1.1 The AFH mechanism

The AFH mechanism possesses several advantages. First, it is relatively simple to implement and compatible with the previous versions of Bluetooth specification. As AFH only re-maps the unused RF channel to one of those used channels, the adapted channel hopping sequence is identical to the basic channel hopping sequence on all locations where the used RF channels are present. As a result, non-AFH slaves are able to keep synchronisation with the master while the other slaves in AFH mode. In addition, the master can broadcast information to both AFH-enable and non-AFH slaves, using one of the used RF frequencies. Secondly, the AFH mechanism strictly adheres to the FCC regulations for the 2.4GHz unlicensed spectrum, due to the fact that it is hopping pseudo-randomly over at least 20 frequencies. Thirdly, the AFH is effective in mitigating the negative effect of FSI. The simulation has shown up to 15-20 percent increase of the average goodput, compared to the standard Bluetooth, in the with-FSI scenario.

The main drawback of the AFH mechanism is that it cannot overcome the self-interference problem. In the non-FSI scenario, the performance of AFH is similar to that of the standard Bluetooth. Furthermore, when an AFH-enabled piconet reduces its hopset to avoid FSI, it actually increases the chance of having collision due to self-interference.

5.1.2 The DAFH mechanism

Working as a non-collaborative method, the DAFH mechanism successfully achieves mutual coordination among interfering piconets without requiring explicit information exchange between them. If the number of collocated piconets is less than the number of available hopsets, it is possible for DAFH mechanism to achieve orthogonal assignment of the hopsets and minimise collision due to both frequency-static and self-interference. In both without- and with-FSI scenarios, DAFH is shown to outperform both of the FR and AFH mechanism in terms of goodput for practically large number of collocated piconets.
5.1 Conclusions

Among two versions of DAFH, the DAFH-AT has better performance. This is due to the fact that DAFH-AT uses higher threshold at higher level of the superset, thus minimises unnecessary hopset reductions. The biased random selection scheme in DAFH-AT has little impact on the performance of the mechanism, and could be replaced by uniformly random selection scheme, which simplifies the implementation.

We have customised a superset for DAFH to work with the 802.11b DSSS interference, which aims at increasing the spectrum efficiency. Although it shows some improvement of frequency occupancy compared to the DAFH with original superset, the advantage of using the modified superset is negligible.

The DAFH-AT triggering threshold must be adapted to the wireless environment so that it does not perform unnecessary hopset reductions due to channel noise. To this end, other channel assessment mechanisms, such as RSSI, are required in conjunction with PER measurement.

There are several restrictions for DAFH mechanism. First, it is not in strict conformance to the FCC regulations for the unlicensed spectrum. However, this could be justified if we consider the group of $N_p$ collocated DAFH-enabled piconets as a collective entity. In this case, the interference caused by such entity to other devices in the spectrum would be similar to that caused by $N_p$ collocated non-DAFH piconets. Secondly, the hopset reduction mechanism in DAFH reduces the robustness of a frequency hopping system. A small hopset could fall completely under an FSI spectrum or channel fading, resulting in poor or lost connection between the master and the slaves.

5.1.3 The FR mechanism

The FR has several advantages, which make it a promising coexistence mechanism for Bluetooth. First, it is proven to be effective for both frequency-static and self-interference. Its performance, especially the FR-WP variant, is higher than that of the AFH scheme for practical number of collocated piconets (i.e. up to 15). Second, by careful design of the parameters, FR mechanism can be made strictly compliant to the current FCC regulations. For example, we can make several rolling cycles in one rolling period, i.e. to have faster rolling speed. In this way, we can always select the duration for a rolling cycle so that, in one rolling period, the FR-enabled piconet passes over at least 20 channels. Third, by utilising block interleaving for the channels, the FR can avoid “deadlock” situation which could occur to DAFH. As its hopset is a combination of frequencies from different parts of the spectrum, it is unlikely that all of them are blocked by FSI or channel fading.

There are two variants of the FR mechanism discussed in the project, namely FR-WP and FR-FGR. The FR-WP exhibits better goodput performance than that of the FR-FGR when FSI is present. Due to slow rate of rolling, it takes the FR-FGR much longer
time to assess the quality of all channels, and thus reducing its performance. A remedy for FR-FGR is to increase the size of frequency group so as to make the assessment process faster. However, this results in inefficient usage of the available spectrum, as FR-FGR might remove more frequencies than necessary for combating the FSI.

5.2 Future works

The first step in the future work is to devise the rules for FR-WP to make it strictly compliant to the FCC regulations. Although we did point out the general method to make FR-WP conformance, the real work has not been done in the project.

Secondly, the project can continue with more extensive studies of the FR-WP mechanism. In this project, we assume the piconet is fully loaded (i.e. the master and the slaves always have packet to send) and uses only 1-slot packet for communications. It is desirable to study the coexistence mechanism under different level of traffic load and with mixture of packet sizes. In addition, it is also important to investigate the performance of FR-WP under more practical channel model. When piconets in the hotspot experience different level of channel noise and interference, they might choose a different set of frequencies for rolling. This makes it more difficult for FR-WP to achieve orthogonalisation of hopsets of the collocated piconets, which might result in performance degradation.

The concepts of coexistence mechanisms, such as DAFH and FR, could be extended to apply for other wireless systems in the unlicensed spectrum. For instance, the subcarriers of OFDM-based transmitters can be turned on and off using DAFH or FR algorithm, to achieve mutual coordination among themselves and with other non-OFDM systems.
Bibliography


List of acronyms

ASB  Active Slave Broadcast
ACK  Acknowledgement
ACL  Asynchronous Connection-Less
AFH  Adaptive Frequency Hopping
ARQ  Automatic Repeat Request
BER  Bit Error Rate
CCK  Complementary Code Keying
CRC  Cyclic Redundancy Checksum
DAFH  Dynamic Adaptive Frequency Hopping
DAFH-AT  DAFH with Adaptive Threshold
DAFH-CT  DAFH with Constant Threshold
DSSS  Direct Sequence Spread Spectrum
eSCO  Extended Synchronous Connection-Oriented
FEC  Forward Error Correction
FCC  Federal Communications Commission
FCS  Frame Check Sequence
FHS  Frequency Hop Synchronisation
FHSS  Frequency Hopping Spread Spectrum
FR  Frequency Rolling
FR-WP  Frequency Rolling with Probing
FFR-WP  Fast Frequency Rolling with Probing
FR-FGR  Frequency Rolling with Frequency Group Removal
FSI    Frequency-static interference
GFSK   Gaussian Frequency Shift Keying
HR-WPAN High Rate - Wireless Personal Network
IEEE   Institute of Electrical and Electronic Engineers
ID     Identification
ISM    Industrial, Scientific and Medical
I/O    Input/Output
HEC    Header Error Check
HIPERLAN Hi gh PE rformance Radio LAN
L2CAP  Logical Link Control and Adaptation Protocol
LAN    Local Area Network
LAP    Lower Address Part
LFSR   Linear Feedback Shift Register
LLC    Logical Link Control
LMP    Link Manager Protocol
LR-WPAN Low Rate - Wireless Personal Network
MAC    Medium Access Control
NAP    Non-significant Address Part
NACK   Negative Acknowledgment
OFDM   Orthogonal Frequency Division Multiplex
OSI    Open System Interconnection
PAN    Personal Area Network
PDA    Personal Digital Assistant
PSB  Parked Slave Broadcast
PER  Packet Error Rate
PHY  Physical Layer
POS  Personal Operating Space
QoS  Quality of Service
RF   Radio Frequency
RSSI Received Signal Strength Indicator
SCO  Synchronous Connection-Oriented
SDP  Service Discovery Protocol
SIG  Special Interest Group
SNR  Signal to Noise Ratio
TCS  Telephony Control Protocol Specification
TDD  Time Division Duplex
TDMA Time Division Multiple Access
WAN  Wide Area Network
WLAN Wireless Local Area Network
WPAN Wireless Personal Area Network
UAP  Upper Address Part
UHF  Ultra High Frequency
U-NII Unlicensed - National Information Infrastructure
### List of symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q$</td>
<td>The channel number for Bluetooth wireless technology</td>
</tr>
<tr>
<td>$f_q$</td>
<td>The center frequency of the $q^{th}$ RF channel for Bluetooth wireless technology</td>
</tr>
<tr>
<td>$BT$</td>
<td>The bandwidth-bit period product for the Gaussian filter</td>
</tr>
<tr>
<td>$T$</td>
<td>The Bluetooth symbol duration</td>
</tr>
<tr>
<td>$f_d$</td>
<td>The peak frequency deviation</td>
</tr>
<tr>
<td>$P_{\text{max}}$</td>
<td>The Bluetooth maximum output power</td>
</tr>
<tr>
<td>$f_n$</td>
<td>The transmission frequency for the $i^{th}$ collocated piconet at time slot $n$</td>
</tr>
<tr>
<td>$N_G$</td>
<td>Number of frequency groups</td>
</tr>
<tr>
<td>$N_{BC}$</td>
<td>Number of repeats for a broadcast message</td>
</tr>
<tr>
<td>$p_n$</td>
<td>The probability of packet error due to channel noise</td>
</tr>
<tr>
<td>$\hat{p}_n$</td>
<td>The estimated maximal probability of packet error due to channel noise</td>
</tr>
<tr>
<td>$p_d$</td>
<td>The probability of having packet corrupte due to FSI</td>
</tr>
<tr>
<td>$p_s$</td>
<td>The probability of having packet corrupte due to self-interference</td>
</tr>
<tr>
<td>$\pi_i$</td>
<td>The $i^{th}$ piconet in a collocated scenario</td>
</tr>
<tr>
<td>$N_p$</td>
<td>The total number of piconets in a collocated scenario</td>
</tr>
<tr>
<td>$\bar{N}_p$</td>
<td>The average number of piconets in a collocated scenario</td>
</tr>
<tr>
<td>$\bar{T}_d$</td>
<td>The average dwell time of a piconet at the hotspot</td>
</tr>
<tr>
<td>$T_{d\text{min}}$</td>
<td>The minimum dwell time of a piconet at the hotspot</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>The generation rate of Poisson process representing the rate of piconet arrivals at the hotspot</td>
</tr>
<tr>
<td>$T_{\text{sim}}$</td>
<td>The simulation time</td>
</tr>
<tr>
<td>$S_{\text{max},DH1}$</td>
<td>The maximum goodput for DH1 packet</td>
</tr>
<tr>
<td>$M$</td>
<td>The total number of channels for Bluetooth wireless technology</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>The frequency-static interference spectrum</td>
</tr>
<tr>
<td>$M_{\Gamma}$</td>
<td>The size of frequency-static interference spectrum</td>
</tr>
<tr>
<td>$P_{\text{coll}}$</td>
<td>The general probability of collision due to both frequency-static and self-interference</td>
</tr>
</tbody>
</table>


\[ P_{\text{col},b} \quad \text{: The probability of collision due to frequency-static interference} \]

\[ P_{\text{col},\bar{b}} \quad \text{: The probability of collision due to self-interference} \]

\[ T_s \quad \text{: The duration of a time slot in Bluetooth wireless technology} \]

\[ T_{\text{on}} \quad \text{: The duration of transmitted packet} \]

\[ T_{\text{off}} \quad \text{: The remaining time slot after packet transmission, which is equal to } T_s - T_{\text{on}} \]

\[ T_{\text{SCO}} \quad \text{: The SCO interval, counted in time slots} \]

\[ T_u^{\text{AFH}} \quad \text{: The AFH updating interval} \]

\[ T_r^{\text{AFH}} \quad \text{: The AFH reused period} \]

\[ P_F^{\text{AFH}} \quad \text{: The AFH removing threshold} \]

\[ N_{\text{min}}^{\text{AFH}} \quad \text{: The minimum number of used channels for AFH mechanism} \]

\[ L \quad \text{: The total number of level in a DAFH superset} \]

\[ M_l \quad \text{: The number of hopsets at the } l^{\text{th}} \text{ level of DAFH superset} \]

\[ u(l,k) \quad \text{: The } k^{\text{th}} \text{ admissible utilisation vector at the } l^{\text{th}} \text{ level in DAFH superset} \]

\[ N_{\text{min}}^{\text{DAFH}} \quad \text{: The minimum allowable size of hopset for DAFH superset} \]

\[ MAX\_ERROR \quad \text{: The pre-defined value of packet errors to start PER evaluation for DAFH} \]

\[ P_c \quad \text{: The estimated average PER for DAFH mechanism} \]

\[ Error\_Count \quad \text{: The DAFH packet error counter} \]

\[ Total\_Count \quad \text{: The DAFH packet received counter} \]

\[ T_d^{\text{DAFH}} \quad \text{: The DAFH doubling period} \]

\[ P_t^{\text{DAFH}}(l) \quad \text{: The DAFH triggering threshold at the } l^{\text{th}} \text{ level} \]

\[ T_d^{\text{DAFH-CT}} \quad \text{: The DAFH-CT doubling period} \]

\[ P_t^{\text{DAFH-CT}}(l) \quad \text{: The DAFH-CT triggering threshold at the } l^{\text{th}} \text{ level} \]

\[ T_d^{\text{DAFH-AT}} \quad \text{: The DAFH-AT doubling period} \]

\[ P_t^{\text{DAFH-AT}}(l) \quad \text{: The DAFH-AT triggering threshold at the } l^{\text{th}} \text{ level} \]

\[ \alpha \quad \text{: The DAFH-AT bias selection parameter} \]

\[ L_e \quad \text{: The number of packet errors at the left half of the current DAFH-AT hopset} \]

\[ R_e \quad \text{: The number of packet errors at the right half of the current DAFH-AT hopset} \]

\[ p_L \quad \text{: The probability to select the left half of the current DAFH-AT hopset} \]

\[ f_s(t_0) \quad \text{: The starting frequency of a FR hopset at time slot } t_0 \]

\[ H \quad \text{: The hopset size for FR mechanism} \]

\[ h_s \quad \text{: The rolling step size for FR mechanism} \]

\[ T_{\text{ri}}^{\text{FR}} \quad \text{: The FR rolling interval} \]

\[ p_j^{\text{FR}} \quad \text{: The FR jump threshold} \]
\(J_k\) : The random jump step size for FR mechanism  
\(T_{p}^{FR-WP}\) : The FR-WP probing period  
\(T_{r}^{FR-WP}\) : The FR-WP rolling period  
\(p_{p}^{FR-WP}\) : The FR-WP removing threshold  
\(p_{j}^{FR-WP}\) : The FR-WP jump period  
\(J_c\) : The maximum value for the jump counter in FR-WP mechanism  
\(M_u\) : The total number of used channels in FR mechanism  
\(T_{r}^{FR-FGR}\) : The FR-FGR rolling period  
\(T_{ru}^{FR-FGR}\) : The FR-FGR reused period  
\(p_{r}^{FR-FGR}\) : The FR-FGR removing threshold  
\(p_{j}^{FR-FGR}\) : The FR-FGR jump period
Design of FCC-compliant FR-WP algorithm

The FCC regulations for the 2.4GHz unlicensed spectrum, as mentioned in section 32.231 requires that a frequency-hopping system operate over at least 15 non-overlapping frequencies, and each frequency be occupied less than 0.4 second within any period of 6 seconds. Due to its slow rolling rate, the FR-WP algorithm does not strictly conform to the FCC rules. In this section, we are going to propose a variant of the FR-WP algorithm, termed Fast Frequency Rolling with Probing (FFR-WP), to cope with the incompliant problem. We will prove that the new variant is in strict compliance with the FCC regulations for the 2.4GHz unlicensed spectrum. In addition, its performance will be analysed under the presence of FSI and self-interference.

C.1 Fast Frequency Rolling with Probing (FFR-WP) algorithm

In this coexistence mechanism, the initial state of a piconet is the probing state. The piconet shall probe the medium for possible FSI for a probing period $T_p^{FFR-WP}$. During this period, the piconet shall use the basic channel hopping sequence, as defined in the Bluetooth specifications, for its transmission. The piconet shall keep records of number of in-error packet and received packet for each hopping channel to estimate the PER statistic. At the end of the probing period, the channels are sorted in order of their average PER. All channels having PER greater than a removing threshold $P_r^{FFR-WP}$ are marked as “bad”. The “bad” channels are then removed, until there is none of them left or the minimum number of used channel, $N_{min} = 20$, is reached. The remained channels shall be used in the next rolling state.

The main difference of the FFR-WP algorithm, compared to the FR-WP, is a faster speed of rolling. In other words, to be in compliant with FCC regulation, the FFR-WP shall use at least 15 frequencies during one rolling period, rather than just $H$ frequencies in case of FR-WP mechanism. This can be done as illustrated in Figure C.1. The rolling period $T_r^{FFR-WP}$ is divided into several equal intervals, which are referred to as rolling sub-intervals $T_{si}^{FFR-WP}$. The number of rolling sub-intervals in one rolling period is $N_{si}$, which can be calculated by:

$$N_{si} = \frac{T_r^{FFR-WP}}{T_{si}^{FFR-WP}}$$  \hspace{1cm} (C.1)
During a rolling sub-interval, a piconet selects channels pseudo-randomly from a subset of used channels, which is defined by a starting frequency $f_s(t)$ and its size $H$. At the end of the rolling sub-interval, the piconet changes the current hopset by shifting it $h_s = H$ step to the right. Note that the shifting is performed in modulo $M_u$, where $M_u$ is the number of used channels. The number of rolling sub-intervals $N_{si}$ is designed in a way that the piconet operates over exactly $N_{min} = 20$ frequencies in one rolling period, or:

$$N_{si} = \frac{N_{min}}{H} \quad (C.2)$$

For example, we choose $H = 4$ to make $N_{si}$ integer. This results in 5 rolling sub-intervals during one rolling period. If the rolling period is 0.4 second, the rolling sub-interval will be 0.08 second.

Similar to the FR-WP algorithm, the PER statistic collected during a rolling period is compared to the jump threshold $P^F_{FR-WP}$ at the end of the period. If the PER is less than the threshold, the piconet continues its rolling operation. Otherwise, the piconet decides that self-interference or FSI has occurred in the last rolling period, and it shall turn into probing state for one probing period before coming back to rolling.

![Diagram](image-url)

**Figure C.1: fast frequency rolling mechanism**

Recall that the piconet always hops randomly over 79 channels during probing period, while it might use less than 79 channels in rolling state. Thus, the rolling period of the FR family is often considered as the worse scenario in terms of frequency occupancy. To
show that the FFR-WP is in strict compliance with FCC regulations, we assume that the piconet is always in rolling state with minimum number of used channels, $N_{\text{min}}$. For FFR-WP algorithm, the piconet operates over $N_{\text{min}}$ frequencies during one rolling period, and thus the average dwell time on a frequency in one rolling period is given as follows:

$$T_D = \frac{T^{\text{FFR-WP}}}{N_{\text{min}}}$$  \hspace{1cm} (C.3)

If we observe the piconet rolling for 6 second, the maximum number of observable rolling periods is $N_v$:

$$N_v = \frac{6}{T^{\text{FFR-WP}}} + 1$$  \hspace{1cm} (C.4)

As a result, during any 6 second period, the maximum dwell time of a frequency ($T_{D\text{max}}$) can be expressed by:

$$T_{D\text{max}} = N_v \times T_D = \frac{T^{\text{FFR-WP}}}{N_{\text{min}}} \times \left( \frac{6}{T^{\text{FFR-WP}}} + 1 \right)$$  \hspace{1cm} (C.5)

As we can see from Eq. (C.5), the dwell time on a frequency depends only on the rolling period and the minimum number of used channels. In our project, the rolling period is 0.4 second and the minimum number of used channels is 20. Thus, the piconet, in the worse scenario, shall use a frequency for maximum 0.32 second in any 6 second observation period, making FFR-WP compliant to the FCC regulations.

Besides compliance to FCC requirements, the FFR-WP has several additional advantages. First, it makes the piconet more robust against channel fading and FSI. In parallel with the usage of channel interleaving scheme, the FFR-WP can further avoid potential “deadlock” situation by hopping over larger number of frequencies during one rolling period. Secondly, the fact that piconet in the rolling state is moving faster is beneficial for piconet in the probing state. If piconet in rolling state is moving slowly, it is possible that the piconet in probing state considers it as a FSI and falsely removes those frequencies from the list of used channels.

### C.2 Performance of FFR-WP mechanism

The performance of FFR-WP mechanism is illustrated in Figure C.2 and C.3. The average goodput and probability of collision for FFR-WP are measured in two simulation scenarios: without and with FSI. Both of the probing period $T_p^{\text{FFR-WP}}$ and rolling period $T_r^{\text{FFR-WP}}$ are 0.4 second. The removing threshold is chosen to be 10 percent, while the jump threshold is 70 percent. In the figures, we compare the performance of FFR-WP
mechanism with that of the FR-WP, both using hopset size $H = 4$. In addition, the number of frequency groups for channel interleaving $N_G$ is 8 and the jump counter $J_c$ is 2 for both schemes. Other simulation parameters are the same as those presented in Table 4.1.

![Comparison of FR-WP and FFR-WP mechanisms (Normalised average goodput, w/o and w/ FSI)](image)

**Figure C.2:** FFR-WP average goodput, with and without FSI

Figure C.2 and C.3 indicate a very similar performance between FFR-WP and FR-WP in without-FSI simulation scenario. Since all piconets in rolling state roll at the same rate, their performance are not affected by fast or slow rate speed of rolling. On the other hand, the FFR-WP exhibits slightly better goodput than that of the FR-WP scheme in scenario where the FSI exists. Such gain in performance might be due to the fact that FFR-WP could further reduce the impact of FSI by hopping over large number of frequencies during one rolling period. In addition, thanks to the fast rolling speed, the piconet in probing state now can differentiate between FSI and self-interference and avoid unnecessarily reduction of the hopset due to self-interference. Keeping the rolling hopset as large as possible is beneficial, especially when the number of collocated piconets is high.
C.3 Conclusions

The design of a FCC-compliant variant of the FR-WP algorithm is presented in this section. The variant is referred to as the Fast Frequency Rolling with Probing (FFR-WP), as it employs special mechanism to increase the rolling speed to make itself conformed to the FCC regulations. We have designed its parameters and proved that the algorithm, with those parameters, is strictly compliant to the regulations.

The performance of FFR-WP is analysed in both simulation scenarios, without and with FSI. The simulation results have indicated that the FFR-WP performs equivalently or better (in the presence of FSI) than the original FR-WP. Besides being conformed to the FCC regulations, the FFR-WP is more robust against channel fading and FSI, while reduces the chance of removing “bad” frequencies due to self-interference.