Sami Ben Cheikh

Design of an Adaptive Frequency Hopping Algorithm Based On Probabilistic Channel Usage

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 28.11.2011

Thesis supervisor:

Prof. Riku Jäntti

Thesis instructor:

Prof. Horst Hellbrück

Aalto University
School of Electrical Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Sami Ben Cheikh

Title: Design of an Adaptive Frequency Hopping Algorithm Based On Probabilistic Channel Usage

Date: 28.11.2011 Language: English Number of pages:8+71

Department of Communications and Networking

Professorship: Radio communications Code: S-72

Supervisor: Prof. Riku Jäntti

Instructor: Prof. Horst Hellbrück

Dealing with interference in the 2.4 GHz ISM band is of paramount importance due to an increase in the number of operating devices. For instance systems based on Bluetooth low energy technology are gaining lots of momentum due to their small size, reasonable cost and very low power consumptions. Thus the 2.4 GHz ISM band is becoming very hostile.

Bluetooth specification enables the use of adaptive frequency hopping to improve performance in the presence of interference. This technique avoids the congested portions of the ISM band, however as the number of interferers increases for a given geographical environment, a greater number of bad channels are removed from the adapted hopping sequence. This results in longer channel occupancy, and consequently higher probability of collisions with coexisting devices, degrading their operation.

At CoSa Research Group a novel algorithm, based on probabilistic channel usage of all channels (good and bad), is developed. The scheme is named Smooth Adaptive Frequency Hopping (SAFH) and uses an exponential smoothing filter to predict the conditions of the radio spectrum. Based on the predicted values, different usage probabilities are assigned to the channels, such as good channels are used more often than bad ones. The discrete probability distribution generated is then mapped to a set of frequencies, used for hopping.

MATLAB/SIMULINK was used to investigate the performance of SAFH, in the presence of different types of interfering devices such as 802.11b , 802.15.4 and 802.15.1. Simulation study under different scenarios, show that our developed algorithm outperforms the conventional random frequency hopping as well as other adaptive hopping schemes. SAFH achieves lower average frame error rate and responds fast to changes in the channel conditions. Moreover it experiences smooth operation due to the exponential smoothing filter.

Keywords: Adaptive Frequency Hopping, Coexistence in the ISM Band, Probabilistic Channel Usage, Interference Mitigation, Exponential Smoothing Filter, WPAN, LR-WPAN, WLAN

Preface

First and foremost, I would like to thank the Almighty GOD for the reasons too numerous to mention. I couldn't stop praising Him for all the good things He has done and brought in my life. Without Him, I would have not completed this project.

I would like to express my love and gratitude to my parents, who encouraged me to pursue a better status in my professional life, and provided me with constant spiritual support and love.

I extend my sincere appreciation to my lovely wife Kirsi for her continuous understanding and support.

I would like to acknowledge and thank Prof. Horst Hellbrück and Tim Esemann for allowing me to conduct my research at CoSa research group, and providing useful feedback.

Finally, last but certainly not least, I would like to express my heartfelt gratitude to Prof Riku Jäntti for his valuable support and advice!

Otaniemi, 28.11.2011

Sami Ben Cheikh

Contents

Symbols and abbreviations

Acronyms and Abbreviations

ACL	asynchronous connection-less link
AFH	adaptive frequency hopping
AIS	adaptive interference suppression
AWGN	additive white Gaussian noise
AWMA	alternating wireless medium access
BER	bit error rate
CCA	clear channel assessment
CDF	cumulative distribution functions
CSMA/CA	carrier sense multiple access / collision avoidance
DIS	deterministic interference suppression
DLL	data link layer
DQPSK	differential quadrature phase shift keying
ED	energy detection
ETSI	European telecommunications standards institute
FCC	federal communications commission
FEC	forward error correction
FHSS	frequency-hopping spread spectrum
GFSK	Gaussian frequency shift keying
HEC	header error check
HiperLAN2	high-performance radio local-area networks
HV	high-quality voice
IEEE	institute of electrical and electronics engineers
ISM	industrial, scientific, and medical
ISO	international organization for standardization
LIFS	long inter frame spacing
LQI	link quality indication
LR-WPAN	low rate wireless personal area network
MAC	medium access control
NACK	negative acknowledgement
O-QPSK	offset quadrature phase-shift keying
PER	packet error rate
PHY	physical layer
PMF	probability mass function
PTA	packet traffic arbitration
RAFH	robust adaptive frequency hopping
RF	radio frequency
RFH	random frequency hopping
RSSI	received signal strength indication
RX	receive/receiver/receiving
SAFH	smooth adaptive frequency hopping
SCO	synchronous connection-oriented
SIFS	short inter frame spacing
TDD	time division duplex
TDMA	time division multiple access
TX	transmit/transmitter/transmission
UBAFH	utility based adaptive frequency hopping
U-NII	unlicensed national information structure
WLAN	wireless local area network
WPAN	wireless personal area network

Terminology and variables

Frame:	collection of bits
Frame error rate (FER):	percentage of erroneous frames
t	classification quantum
α	smoothing factor for the exponential filter
c, s	SAFH weighting factors
$d_i(t)$	difference between ξ and FER_i
$FER_i(t)$	FER estimated at channel i
$FER_i'(t+1)$	predicted FER for channel i
$\overline{FER}(t)$	average frame error rate
N	number of available channels
$N_i^e(t)$	number of erroneous frames over channel i
$N_i^{tr}(t)$	number of transmitted frames over channel i
ξ	threshold on the frame error rate
$P(C)$	Probability of Collision
$P(E)$	Packet error Rate
$P(EF)$	probability of error free packet
$p_i(t)$	probability that channel i is used
P_L	traffic load
$P(L)$	packet loss

Operators

$\sum_i^N$	Sum from i till N
$P(X=i)=p_i$	probability mass vector
max	maximize

1 Introduction

1.1 Motivation

Due to its unlicensed nature and large spectrum, the 2.4 GHz Industrial, Scientific, and Medical (ISM) band is growing in popularity. As a result, radio systems operating in this band exhibit adaptive usage of the spectrum in order to improve their performance, and cope with high level of interference from coexisting devices.

A typical mechanism used, is the standard adaptive frequency hopping (AFH) [1], which identifies and avoids using bad channels. This technique is efficient in the presence of static sources of interference i.e. coexisting devices that use the same portion of the ISM band continuously, such as WLAN.

However, if the source of interference is dynamic e.g. frequency hopping systems, then the standard AFH is not efficient. Schemes such as orthogonal hop-set partitioning (OHSP) [2] and dynamic adaptive frequency hopping (DAFH) [3], can handle both static and dynamic sources of interference simultaneously, at the cost of reducing the hop-set size. This results in longer channel occupancy and therefore higher probability of collisions with coexisting devices, degrading their operation.

A novel approach for mitigating interference is based on probabilistic channel occupancy [4–6], all channels (good and bad) are assigned usage probability based on the status of channels. This approach is appealing, since it exploits frequency diversity, however the schemes found in the literature have their limitations, therefore new adaptation techniques are needed.

This thesis discusses the design of a new algorithm, that rectifies the shortcomings of existing schemes.

1.2 Problem Formulation

The ISM Band

Use of radio frequency (RF) bands is regulated by authorities such as the Federal Communications Commission (FCC) in the United States (US), and the European Telecommunications Standards Institute (ETSI) in Europe. These regulators define part of the radio spectrum as licence exempt (unlicensed) for private users, i.e. anyone can transmit as long as they meet certain requirements.

There are three main unlicensed bands suitable for sophisticated data transmission: The industrial, scientific, and medical (ISM) bands; the unlicensed national information structure (U-NII) and the high-performance radio local-area networks (HiperLAN2). Specifications and allowable uses of these bands vary based on local regulations, so products must be certified to conform to the rules of the specified country, to be able to transmit.

The 2.4 GHz ISM band, is available globally, thus it offers a rare opportunity for manufacturers to develop products for world wide market. The Federal

Communications Commission (FCC), originally[1] required radios operating in the 2.4 GHz ISM band to apply spread spectrum techniques[2], if their transmitted power level exceeds 0 dBm. Systems using these techniques, deliberately spread the message signal in the frequency domain, resulting in a much wider bandwidth and consequently in lower power density. This is a desired feature, since it minimizes interference to other receivers nearby, and ensures robust performance in a noisy radio environment [7].

The Interference Problem

The most widespread networking systems in the 2.4 GHz ISM band are the IEEE 802.15.1 Bluetooth [8], IEEE 802.11 wireless local area networks (WLAN) [9] and IEEE 802.15.4 low rate wireless personal area networks (LR-WPAN) [10].

Bluetooth devices are based on frequency hopping spread spectrum (FHSS), since it better supports low-cost and low-power radio implementations; this technique divides the entire spectrum into several frequency channels; the signal is transmitted on a certain carrier frequency for a time T_{BT}, after which the carrier frequency shifts (hops) to another frequency and so on; the number of hops per second is referred to as the hop rate; In this text the term Bluetooth or simply IEEE 802.15.1 refers to a WPAN that utilizes the Bluetooth wireless technology.

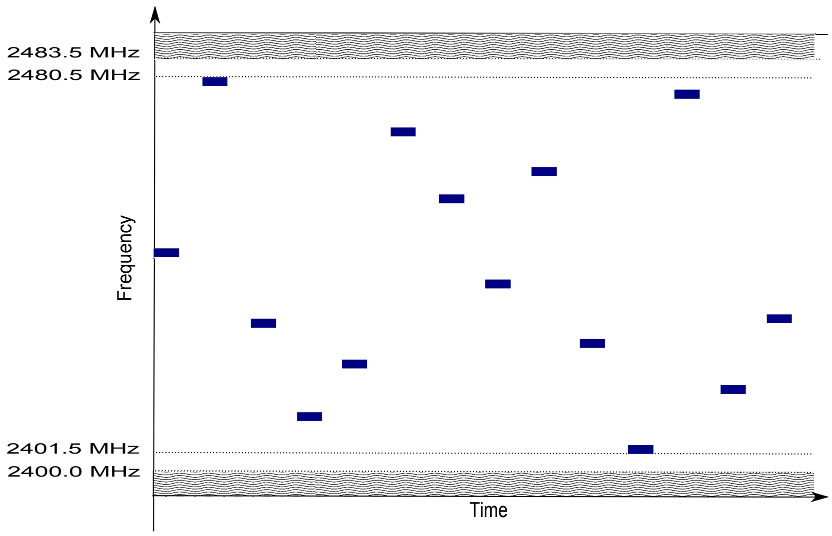

Figure 1: Frequency Occupancy of Bluetooth networks

Section 15.247(a) of the FCC regulations required FHSS devices to hop over at least 75 channels and limit the maximum bandwidth of each hopping channel to 1 MHz; as a result Bluetooth[3] devices hop over 79 frequencies numbered 0 to

[1] Since 1986

[2] Nowadays the rules are relaxed and digital modulation techniques, such as orthogonal frequency-division multiplexing (OFDM)are also allowed in the 2.4 GHz ISM band

[3] Bluetooth is designed to be compliant with international standards, including ETS 300 328

78 in a pseudo random manner.

The hopping pattern is represented graphically in Figure 1; each rectangle represents a Bluetooth transmission.

Figure 1 shows that at any specific instance only 1 MHz is occupied; however when viewed over time, the energy of the transmitted signal is effectively spread over a bandwidth of 79 MHz; this spreading allows Bluetooth to mitigate the effects of fading as well as interference.

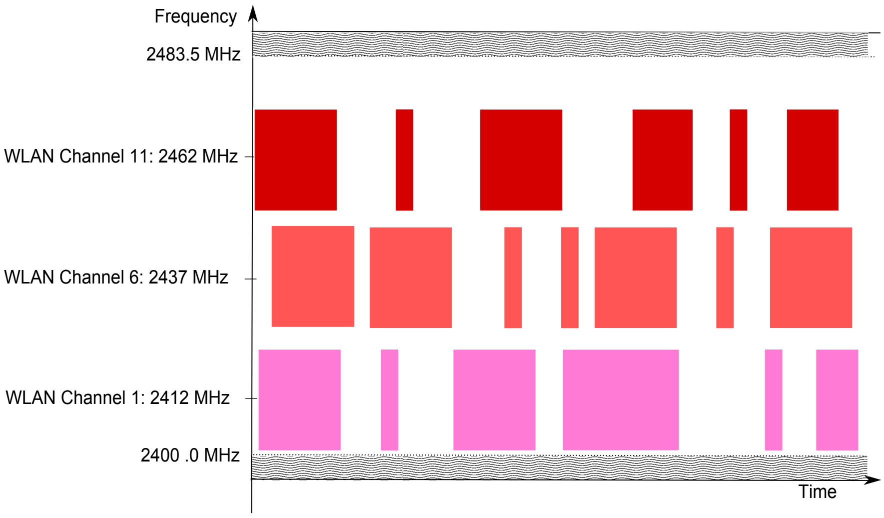

Figure 2: Frequency Occupancy of Three WLAN networks

The IEEE 802.11b[4] are based on direct sequence spread spectrum (DSSS) where information is spread out into a much larger bandwidth by using a pseudo-random chip sequence; in this text WLAN and IEEE 802.11b will be used interchangeably, unless otherwise stated.

The IEEE 802.11b standard defines 11 possible channels (22 MHz each), so only three of them can be used at the same time.

Figure 2 shows how IEEE 802.11 networks maintain the same frequency usage over time, thus they are referred to as *frequency static* devices [1], in contrast to Bluetooth which we will refer to as *frequency dynamic* devices.

IEEE 802.15.4[5] devices are also based on DSSS, however the spread signal has only a bandwidth of 2 MHz each.

Figure 3 shows a typical frequency occupancy for three LR-WPAN networks. Each network operates exclusively on one channel thus they are also considered as *frequency static* devices; the figure shows networks operating on channels 15, 20 and 25.

Because IEEE 802.15.1, IEEE 802.11b and IEEE 802.15.4 specify operations in the same 2.4 GHz unlicensed frequency band, there is potential for mutual

[4]802.11g is backwards compatible with 802.11b, however it achieves higher data rates by implementing an additional OFDM transmission scheme.

[5]LR-WPAN and 802.15.4 will be used interchangeably

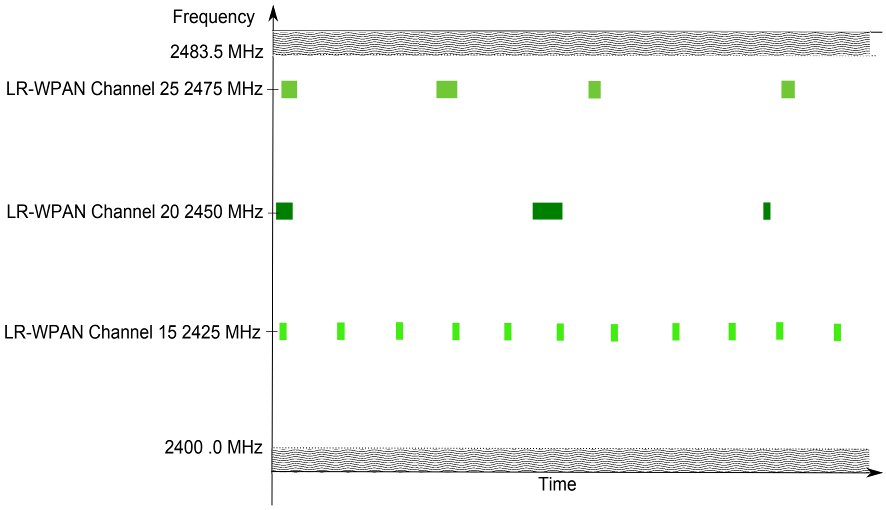

Figure 3: Frequency Occupancy of Three LR-WPAN networks

interference between the wireless systems. Figure 4 shows a typical dense deployment of two independent Bluetooth piconets (using different hopping sequences), two 802.11b and three 802.15.4 systems.

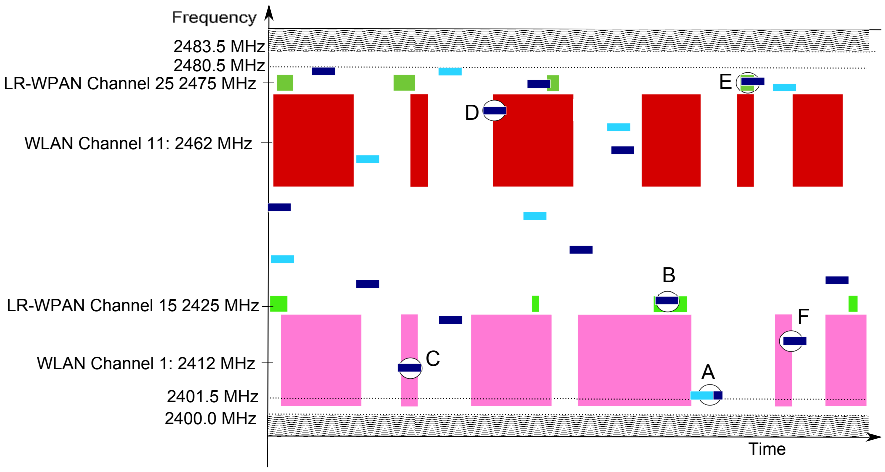

Figure 4: Collision in the ISM Band

The interference problem is characterized by a collision i.e. a time and frequency overlap between the wireless systems. This occurs when both Blutooth piconets use the same hop, or when IEEE 802.15.1 hops into IEEE 802.11 or IEEE 802.15.4 passband. This is depicted as circles in Figure 4.

When the radios are physically separated, spread spectrum techniques are effective in dealing with multiple users in the band; however when they operate in close proximity, neither FHSS nor DSSS is able to mitigate the interference [11] among devices belonging to different classes such, as a Bluetooth piconet interfering with an IEEE 802.11, or even among devices of the same type, such as Bluetooth on Bluetooth; as a result there will be significant performance degradation.

Need for Coexistence

Coexistence means that systems can be collocated without significantly impacting the performance of each other; it is defined as *"the ability of one system to perform a task in a given shared environment where other systems have an ability to perform their tasks and may or may not be using the same set of rules"* [1].

In view of this definition the pseudo random frequency hopping scheme used in Bluetooth Version 1.1 does not ensure Bluetooth coexistence, since the selection process happens without consideration for current occupants of the spectrum; therefore there is potential for collision and consequent possible degradation in performance for operating networks.

The Bluetooth Special Interest Group (SIG) [12] and the IEEE 802.15.2 Coexistence Task Group [1] collaborated on efforts to define mechanisms and recommended practices, to ensure the coexistence of Bluetooth devices. One of the practices proposed is Adaptive Frequency Hopping (AFH), a technique that addresses interference problem by actively modifying the hopping sequence to avoid congested channels.

1.3 Objective and Methodology

The objective of this thesis work are threefold:

- to investigate and classify different adaptive frequency hopping techniques, and study their limitations in the presence of different types of interference, i.e. frequency static devices such as IEEE 802.11b, as well frequency dynamic interfering devices, such other independent Bluetooth piconets;
- to propose more effective algorithm that can enhance the coexistence capability of IEEE 802.15.1 Networks;
- to examine the parameters and scenarios under which it is more practical to use one hopping mechanism over the others.

In order to quantify the effect of interference, two approaches[6] are used:

- A detailed analytical performance of the newly developed frequency hopping algorithm in order to obtain a first order approximation; The performance

[6] Unfortunately over the air experimental approach using "GNU Radio" [13] and USRP2 [14] framework is left out, due to timing constraints.

metrics in the theoretical part, are the frame error rate (FER) as well as the probability of collision between over the air frames;

- A PHY layer simulation, where different frequency hopping schemes are investigated and benchmarked with the new algorithm; this phase provides a more flexible framework and complements the results obtained from analytical studies. The performance metric is frame error rate FER, i.e. the percentage of frames in errors after performing forward error correction.

Note that the terms frame and packet are used interchangeably in the literature; however the IEEE 802.15.4 standard uses the term packet to refer to a collection of bits of be transmitted, but uses the term frame for a collection of bits that is processed at higher layers in the protocol stack. The IEEE 802.11b standard uses the term frame, while the IEEE 802.15.1 uses the term packet all the time. We try to adhere to these terms, when referring to a particular protocol. However, in general we will refer to a collection of bits as *frame*.

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

Section 2 provides the necessary background needed in this paper. It starts with an overview of the wireless technologies operating in the 2.4 GHz band; in particular this clause highlights the technical details needed to put the research problem into context; then it discusses different coexistence methods, and finally it treats in detail three interesting schemes, that will be compared to our developed algorithm; benchmarking is in term of performance and complexity.

In Section 3 the design of a novel adaptive frequency hopping scheme, named smooth adaptive frequency hopping (SAFH),is described; the main elements of the algorithm are discussed in detail; pseudo code and illustrative example are used to clarify the steps.

In Section 4 the coexistence problem is modelled mathematically, where the impact of SAFH on the performance of collocated networks (IEEE 802.15.1(BT), 802.11b (WLAN) and 802.15.4) is presented; in addition, the impact of other wireless devices on our algorithm is captured.

IEEE 802.15.1 uses two types of links that have different levels of sensitivity to the interference. We decided to study voice link because it may be more sensitive to interference than a data link used to transfer a data [1].

Section 5 introduces the methodology of simulating different adaptive frequency hopping algorithms, including the proposed algorithm (SAFH); different scenarios are considered with special attention to cases when a combination of dynamic and static sources of interference are operating near by.

Section 6 presents the outcome of the simulations (results); in particular it discusses how SAFH achieves lower average frame error rate (FER), faster adjustment to changes in the environment and smoother operation i.e. less fluctuations, compared to the other schemes.

In Section 7 we provide some concluding remarks and point out the future research directions.

2 Literature Review

This section introduces the specifications of Bluetooth, WLAN and ZigBee, followed by a discussion on the coexistence methods used to mitigate interference.

2.1 Wireless Technologies in the 2.4 GHz ISM Band

IEEE 802.11 and IEEE 802.15.4 standards [9, 10] define both the physical (PHY) and medium access control (MAC) layer protocols, for WLANs and LR-WPAN respectively. They use an architectural approach that emphasizes the logical divisions of the systems into two parts (PHY/MAC), and how they fit together. The IEEE 802.15.1 protocol stack, on the other hand, does not closely follow the traditional ISO layering except for the lower layers i.e. PHY/DLL, as shown in Figure 5. It is usually presented [8, 15] using the so called functional approach, which emphasizes the actual modules, their packaging, and their interconnections.

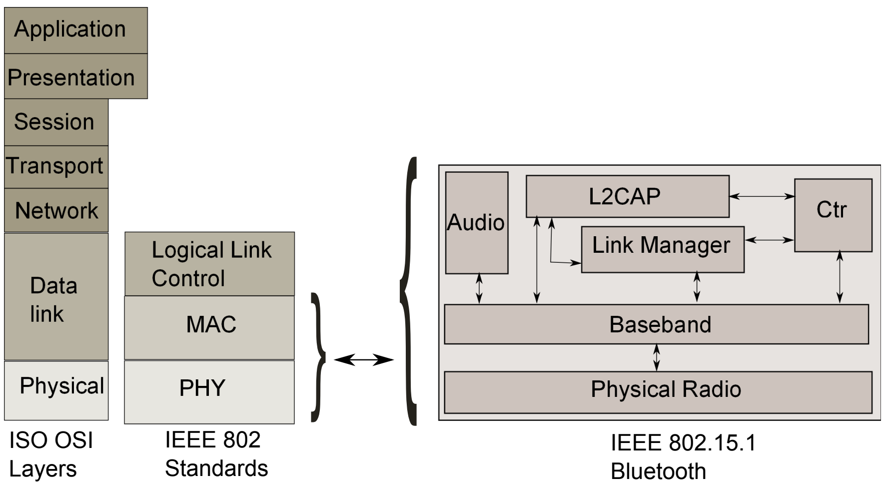

Figure 5: Mapping of ISO OSI to scope of IEEE 802.15.1 WPAN standard (after [1])

In what follows, an attempt is made to introduce these wireless systems using the traditional architectural approach. Only the subset of the communication protocols that are relevant to this report are discussed.

2.1.1 The IEEE 802.15.1 Specifications

Bluetooth technology and standards [8] provide the means for replacing a cable that connects one device to another with a universal short-range radio link. The technology encompasses a simple low-cost, low power, global radio system for integration into mobile devices.

Bluetooth transmitters fall into three basic classes, determined by their maximum power output. The class 1 transmitter has a maximum power of 100 mW (+20 dBm), while class 2 transmitters have a maximum power of 2.5 mW (+4 dBm). The class 3 transmitter has a maximum power of 1 mW (0 dBm) resulting in a range of up to 10 meters[7], which is sufficient for cable-replacement applications. In addition it is an attractive option due to its low power-consumption.

The Bluetooth network is called a piconet. In the simplest case, it means that two or more units are connected; one unit acts as a master, controlling traffic on the piconet, and the other units act as slaves (a maximum of seven slaves can be active at the same time). Bluetooth connections are typically ad hoc connections i.e. the network will be established just for the current task and then dismantled after the data transfer has been completed.

Channel definition Bluetooth operates in the ISM frequency band starting at 2.4015 GHz and ending at 2.4805 GHz. Since the $2.4GHz$ ISM band is unlicensed, Bluetooth radios use frequency hopping spread spectrum (FHSS) to cope with the unpredictable sources of interfering devices, as was discussed in Section 1.2. When, interference jams a hop channel, causing faulty reception, the erroneous bits are restored using error-correction schemes. There are 79 RF channels, $1MHz$ width each, with centre frequencies defined by the formula:

$$f = 2402 + k\ (MHz)\ k = 0 \dots 78 \tag{1}$$

With Gaussian shaped frequency shift keying (FSK) modulation, a symbol rate of $1Mbps$ can be achieved.

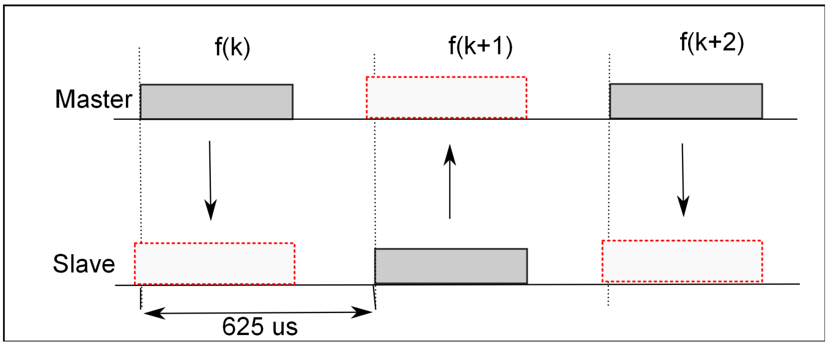

Figure 6: Frequency-hop/time-division-duplex channel.

The channel is divided into $625us$ intervals called slots where a different hop frequency is used for each slot. This gives a nominal hop rate of 1,600 hops per second. Thus Bluetooth channels are defined as frequency hop/time division duplex (FH/TDD) scheme. One packet can be transmitted per slot, and the additional time is used by the radio to change to the next frequency in the hop sequence and activate the appropriate transmitter or receiver. Subsequent slots are alternately used for transmitting and receiving, which results in a TDD scheme [8, 16], as shown in Figure 6.

[7]In an obstacle-free environment

The hopping sequence is determined by the hop-set generator which takes 27 bits of master's clock value and 28 bits of the master's device address as inputs, and then generates a hop frequency, as illustrated in Figure 7. The detailed mathematical operations can be found in [15], but generally speaking, the hop sequences generated have low correlation with each each other, and contain all the available channels with equal probability. In addition, the repetition interval of the sequence is 2^{27} i.e. more than 23 hours.

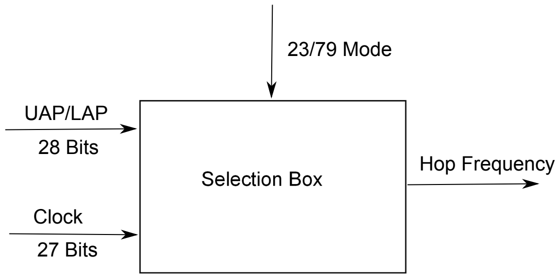

Figure 7: Block diagram of the hop-set generator.

Links and Packet Formats There are two types of link connections that can be established between a master and a slave: the Asynchronous Connection-Less (ACL) link, and the Synchronous Connection-Oriented (SCO).

- The ACL link, is an asymmetric point-to-point connection between a master and active slaves in the piconet. It is is used where data integrity is more important than latency. Several packet formats are defined for ACL and can occupy 1, 3, or 5 time slots. Each packet consists of three entities: the access code, the header, and the payload. The construction of the packet and the number of bits per entity are shown in Figure 8. The size of the access code and the header are fixed, while it varies for the payload (from 0 to 2745 bits per packet).

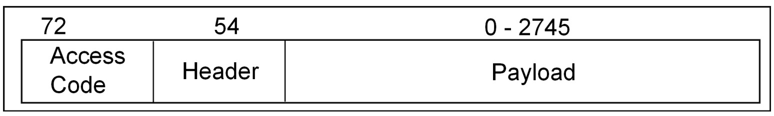

Figure 8: Standard packet format in Bluetooth

An Automatic Repeat Request (ARQ) procedure is applied to ACL data, where packets are retransmitted in case of loss, until a positive acknowledgement (ACK) is received at the source. To reduce the number of retransmissions, some ACL packets use Forward Error Correction (FEC).

- The SCO link is a symmetric point-to-point connection between a master and a slave, where packets are sent at regular intervals called SCO interval

T_{SCO} (counted in slots). The SCO link reserves slots and can therefore be considered as a circuit-switched connection, suited for time-bounded information like voice. There are three types of SCO packets: HV1[8], HV2, and HV3, shown in Table 1. All SCO packets occupy one time slot and are defined to carry 64 Kbits/s of voice traffic, that is not retransmitted in case of packet error or loss. T_{SCO} is set to either 2, 4 or 6 time slots for HV1, HV2 and HV3 respectively. In addition, SCO packets differ in the amount of digitized voice contained in each one due to FEC. HV1 uses (3,1) binary repetition code, where a 1 is encoded as 111 and a 0 is encoded as 000. At the receiver a majority vote is taken to determine the actual bit that was sent. HV2 uses (3,2) repetition code, while HV3 does not use FEC.

Table 1: Structure of SCO HV Packets

Type	Payload (number of bits)	FEC Rate
HV1	80	$\frac{1}{3}$
HV2	160	$\frac{2}{3}$
HV3	240	None

2.1.2 The IEEE 802.11 Specifications

The IEEE 802.11 standard [9] calls for different PHY specifications, such as frequency hopping spread spectrum (FHSS), direct sequence spread spectrum (DSSS), and infrared. This sequel will focus on the 802.11b specification DSSS spread spectrum which operates in the same frequency band as Bluetooth. The transmit power for DSSS devices is defined at a maximum of 1 W[9] and the receiver sensitivity is set to $-80dBm$ [9].

The IEEE 802.11b standard defines 11[10] possible channels spaced 5 MHz apart, as illustrated in Equation (2). The channels are numbered 1 to 11 and have a bandwidth of 22 MHz each, therefore to avoid overlap, only channels 1, 6 and 11 can be used at the same time [11], as illustrated previously in Figure 2.

$$f = 2407 + 5 * k \ (MHz) \ k = 1 \ldots 11 \qquad (2)$$

The IEEE 802.11b Physical layer delivers packets at $1, 2, 5.5$, and 11 Mbps rates in the 2.4 GHz ISM band. The basic data rate is $1Mbps$ encoded with differential binary phase shift keying (DBPSK). Similarly, a 2 Mbps rate is provided using differential quadrature phase shift keying (DQPSK) at the same chip rate. Higher rates of 5.5 and 11 Mbps are also available using techniques combining quadrature phase shift keying and complementary code keying (CCK) [11]; this is depicted in Figure 9.

[8] HV: high-quality voice

[9] In the US (FCC 15.247)

[10] Country specific bands have different number of frequencies, defined in IEEE 802.11 and IEEE 802.11.d)

[11] This applies to the US; in Europe the non overlapping channels are 1, 7 and 13

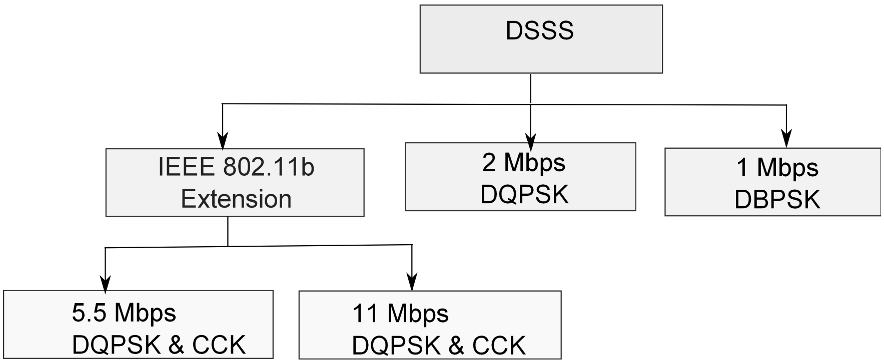

Figure 9: IEEE 802.11b: Different Bit Rates

PHY is also in charge of energy detection (ED) i.e. estimation of the received signal power within the bandwidth of an IEEE 802.11 channel. The ED threshold varies depending on the data rate and the transmit power (TX) e.g. ED level decreases as the TX power increases[12].

The PHY layer uses a clear channel assessment (CCA) algorithm to determine if the channel is busy or idle. The 802.11b specification defines several modes of CCA operation which can be incorporated into the node:

- Energy above threshold (low and high-rate data): the CCA reports a busy medium upon detection of any signal energy above the ED threshold.
- Carrier sense only (low-rate data): the CCA reports a busy medium only upon detection of DSSS signal.
- Carrier sense with energy above threshold (low-rate data): this is a combination of the aforementioned techniques. The CCA reports that the medium is busy only if it detects a DSSS signal and with energy above the ED threshold.
- Carrier sense with timer (high-rate data): CCA starts a timer upon detection of high-rate data signal. After the expiration of the timer CCA reports the status of the medium i.e. idle or busy.
- Carrier sense with energy above threshold (high-rate data): the CCA reports a busy medium upon detection of high-rate signal with energy above the ED threshold.

The IEEE 802.11 MAC layer specifications, common to all data rates, coordinate the communication between stations and control the behaviour of users who want to access the network. The Distributed Coordination Function (DCF), which describes the default MAC protocol operation, is based on a scheme known

[12] Since the node's higher transmit power has the potential to interfere with other networks over a great distance, it shall sense that the channel is busy when a weaker signal is present [17]

as carrier sense multiple access collision avoidance (CSMA/CA[13]), where Both the MAC and PHY layers cooperate in order to avoid collision [11].

The MAC layer also provides an optional mechanism called virtual carrier sense. It uses the request-to-send (RTS) and clear-to-send (CTS) message exchange, to make predictions of future traffic on the medium and updates the network allocation vector (NAV) available in stations [11]. Communication is established when one of the wireless nodes sends a short RTS packet, to request the use of the medium. If this succeeds, the receiver will quickly reply with a short Clear To Send (CTS), then the actual transmission takes place.

The MAC is required to implement basic access procedure as follows; when a frame is available for transmission, the sending node monitors the channel for a time equal to a DCF inter-frame space (DIFS). If the medium remains idle, the station goes into a back-off procedure before it sends its frame. Upon the successful reception of a frame, the destination station returns an ACK frame after a Short inter-frame space (SIFS), as shown in Figure 10.

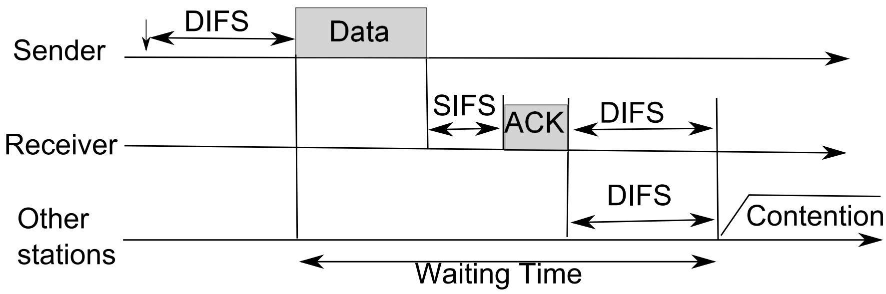

Figure 10: Basic access procedure, Regardless of whether the virtual carrier sense routine is used or not.

The back-off window is based on a random value uniformly distributed in the interval $[CWmin; CWmax]$; CWmin and CWmax represents the contention window parameters, and they are PHY dependent e.g. in 802.11b: $CWmin = 31, CWmax = 1023$ [18], as shown in Figure 11. If the medium is determined busy at any time during the back-off slot, the back-off procedure is suspended. It is resumed after the medium has been idle for the duration of the DIFS period. If an ACK is not received within an ACK time-out interval, the station assumes that either the data frame or the ACK was lost and needs to retransmit its data frame by repeating the basic access procedure.

2.1.3 The IEEE 802.15.4 Specifications

The IEEE 802.15.4 protocol [10], specifies the physical layer and MAC sub-layer for Low-Rate Wireless Personal Area Networks, shown in Figure 12. The intent of IEEE 802.15.4 is not to compete with WLANs and Bluetooth technologies, but rather to provide low data rate communications using nodes that are simple, low

[13]This is similar to p-persistent CSMA, in which p adjusts dynamically to channel loading

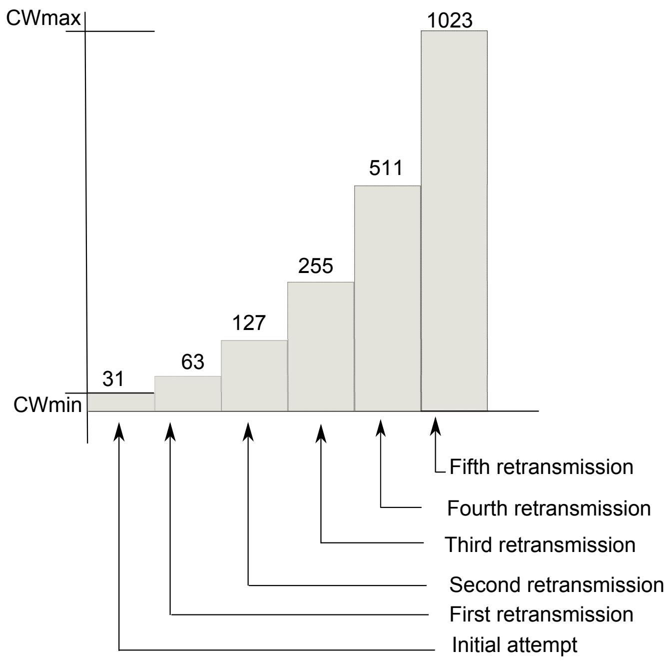

Figure 11: Contention Window adjustment

cost and consume little power. The operational duty cycle is also expected to be low (typically 1%) for applications, such as sensors and industrial control [17]. Transmitters shall be capable of a transmit power of at least −3 dBm, but should transmit at a lower power when possible to reduce interference. The receiver sensitivity is set to −85 dBm [10].

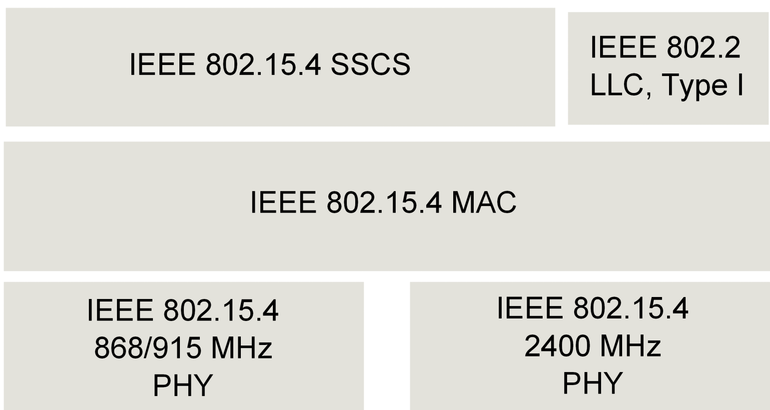

Figure 12: IEEE 802.15.4 Architecture

ZigBee [19] is the set of specifications built on the PHY and MAC layers laid out in the IEEE 802.15.4 specification; it adds network, security and application profiles as depicted in Figure 13. Since in this report we are concerned only with PHY and MAC layers, we will be using IEEE 802.15.4, LR-WPAN and ZigBee interchangeably.

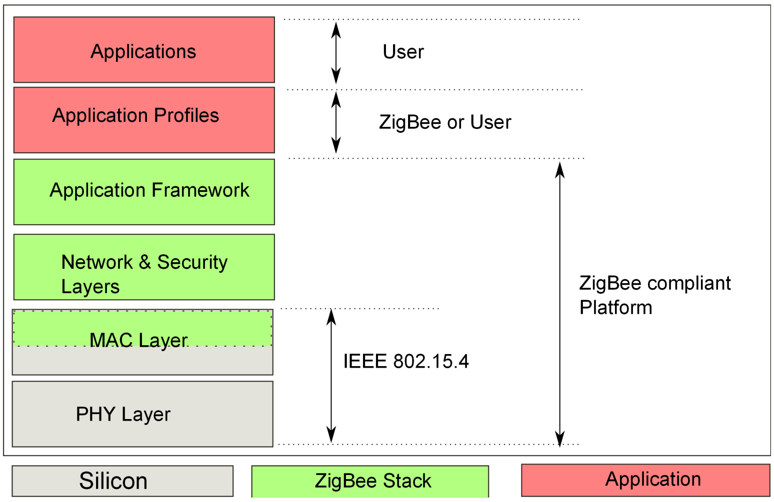

Figure 13: ZigBee Protocol Stack

The IEEE 802.15.4 device must operate in at least one of three bands: $868MHz$ in Europe, $915MHz$ in the United States and 2.4 GHz worldwide. The transmit scheme in all these frequency bands is based on the Direct Sequence Spread Spectrum (DSSS) technique. There is a single channel between 868 and 868.6 MHz, 10 channels between 902 and 928 MHz, and 16 channels numbered 11 through 26 between 2.4 and 2.4835 GHz. The centre frequencies are defined by the formula:

$$f = 2405 + 5 * (k - 11)\ (MHz)\ k = 11 \ldots 26 \tag{3}$$

Channel separation in the 2.4 GHz frequency band is 5 MHz to allow a faster chip rate of 2 Mchips/s. The data rate in the 2.4 GHz ISM band supports 250 Kbps, encoded with offset quadrature phase-shift keying (O-QPSK).

In a similar way to IEEE 802.11b, the physical layer of the IEEE 802.15.4 is in charge of energy detection (ED) and clear channel assessment (CCA), among other things. ED is an estimation of the received signal power within the bandwidth of an IEEE 802.15.4 channel.

The 802.15.4 specification defines three modes of CCA operation; at least one of which can be incorporated into the node:

- Energy above threshold: the CCA reports a busy medium upon detection of any signal energy above the ED threshold.
- Carrier sense only: the CCA reports a busy medium only upon detection of a signal with the modulation and the spreading characteristics of IEEE 802.15.4.
- Carrier sense with energy above threshold: the CCA reports a busy medium upon detection of a signal with the modulation and spreading characteristics of IEEE 802.15.4 and with energy above the ED threshold.

The MAC sub-layer of the IEEE 802.15.4 protocol has many common features with the MAC sub-layer of the IEEE 802.11 protocol, such as the use of CSMA/CA and the support of contention-free and contention-based periods. However, the specification of the IEEE 802.15.4 MAC sub-layer is adapted to the requirements of LR-WPAN, for instance, the Request to Send/Clear to Send RTS/CTS mechanism is eliminated [10, 20].

The timing associated with CSMA/CA algorithm is depicted in Figure 14. ZigBee measures inter frame spacing in terms of symbol periods. Long frames are followed by long inter frame spacing (LIFS), while short frames are followed by short inter frame spacing (SIFS). When the frame is acknowledged, LIFS and SIFS follow the associated ACK.

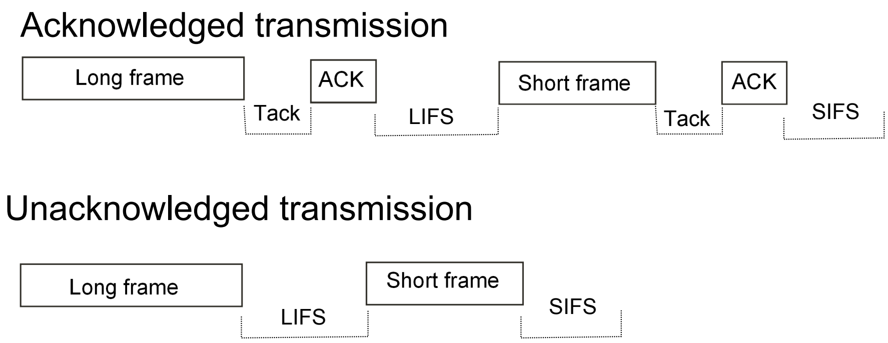

Figure 14: CSMA/CA Channel Access Timing

2.2 Coexistence Framework

Coexistence Task Force

There are few industry led activities and task forces tackling the issue of coexistence. The Dynamic Spectrum Access Networks (DySPAN) [21] standards committee, formerly known as the IEEE P1900 [22] Standards Committee, develops standards for radio and spectrum management. One of its recommended practices, the IEEE P1900.2, deals with interference and coexistence analysis. It provides technical guidelines for analysing the potential for coexistence or, in contrast, interference between radio systems operating in the same frequency band or between frequency bands [22].

Prior to the formation of the IEEE P1900 Standards Committee, the IEEE 802.15.2 Task Group on coexistence published a document [1] that considers solutions for mitigating the interference between Bluetooth and IEEE 802.11b devices; these solutions will be discussed shortly.

Types of Coexistence

Coexistence methods are classified as either collaborative or non collaborative.

- Collaborative coexistence mechanisms are intended for WLANs and WPANs that are able to negotiate access to the medium, therefore a communication link between the networks is required. A prime example that has profound effects on the market, is a personal computer equipped with both Bluetooth and WLAN.

  Collaboration can be based either on Medium access control (MAC) or physical layer (PHY) solution. The 802.15.2 recommended practice [1] lists three collaborative methods, to improve performance between WIFI and Bluetooth nodes. These are "Alternating Wireless Medium Access (AWMA)", "Packet Traffic Arbitration (PTA)" and "Deterministic Interference Suppression (DIS)".

  AWMA is a MAC time domain solution that utilizes a portion of the IEEE 802.11 beacon interval for IEEE 802.15.1 operations. Figure 15 illustrates how the beacon interval T_B, is subdivided into two subintervals: one for WLAN traffic and one for Bluetooth traffic (WPAN). From a timing perspective, the medium assignment alternates between IEEE 802.11 and IEEE 802.15.1, and each wireless network restricts their transmissions to the appropriate time segment. As a consequence interference between the two wireless networks is prevented. [1, 23].

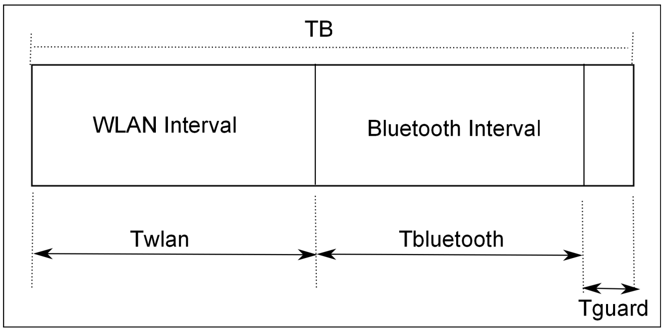

Figure 15: Timing of the WLAN and WPAN subintervals

  PTA is also a MAC time domain solution that provides per-packet authorization of all transmissions. Each attempt to transmit by either the IEEE 802.11b or the IEEE 802.15.1 is submitted to a control entity for approval, as shown in Figure 16; transmit requests that would result in collision are denied [1, 24].

  DIS is a PHY solution designed to mitigate the effect of IEEE 802.15.1 interference on IEEE 802.11b. The basic idea of this mechanism is to put a null in the WLAN's receiver at the frequency of the Bluetooth signal. However, because IEEE 802.15.1 is hopping to a new frequency for each packet transmission, the IEEE 802.11b receiver needs to know the hopping pattern, as well as the timing of the IEEE 802.15.1 transmitter [1, 25].

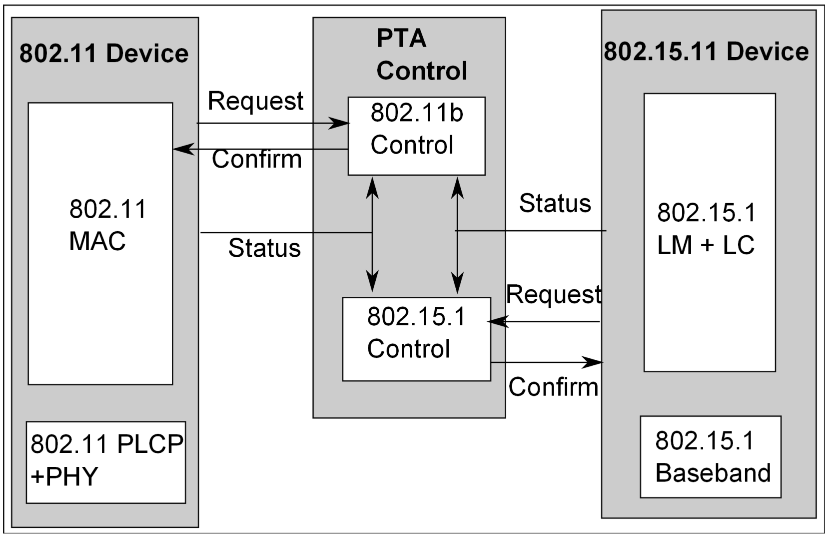

Figure 16: Structure of the PTA entity

- Non collaborative coexistence mechanisms do not require a communication link between WLAN and WPAN; they can be based either on MAC or PHY solution e.g. Adaptive Interference Suppression (AIS) is a non collaborative PHY solution, used by WLAN to estimate and cancel the Bluetooth signal without priori knowledge of the timing or frequency used by it. The block diagram of AIS system is shown in Figure 17. First of all, the received signal, $x(n)$, is delayed and passed through the adaptive filter, which exploits the uncorrelated nature of the wideband IEEE 802.11 signal to predict the unwanted narrowband IEEE 802.15.1 signal, $y(n)$. This estimate is subtracted from the received signal to generate the prediction error signal, $e(n)$, which is an approximation of the IEEE 802.11 signal [1].

  The next subsection goes into more details on Bluetooth Non collaborative schemes.

Coexistence Mechanisms in Bluetooth

In Bluetooth, the non collaborative coexistence schemes rely on adaptive control strategies such as frequency hopping, packet selection and MAC parameter scheduling. All the schemes start by assessing the ISM band, then take action based on the status of the channels.

The first control action known as adaptive frequency hopping (AFH)[14] modifies the frequency hopping pattern so that bad channels are avoided.

In adaptive packet selection technique, packets are selected according to the channel condition of the upcoming frequency hop, resulting in better network performance [1]. When the network performance is range limited[15], packets are

[14] This technique will be explained in detail shortly.

[15] The stations are separated by a distance, such that only small noise margin is maintained.

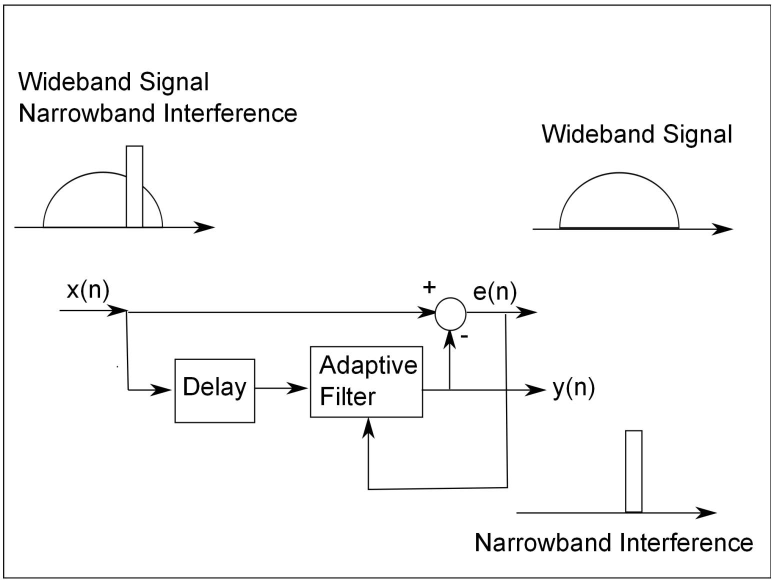

Figure 17: Adaptive notch filter used in AIS

mainly dropped due to random bit errors, therefore packets that use more error protection will increase the performance of the link e.g. HV1 packets are preferred over the HV2 packets in this case.

However in coexistence scenarios, the dominant reason for packet drop is due to the strong interference produced by the collocated networks. In this case, increasing FEC protection will cause more interference to the collocated networks,thus the total network throughput is severely degraded and the good neighbour policy is violated[16].

MAC scheduling is yet another action where packet transmission are carefully scheduled [1]. Since there is a slave transmission after each master transmission, the Bluetooth master checks both the slave's receiving frequency and its own, before choosing to transmit a packet in a given hop. The transmission is delayed until both the master's and slave's receiving frequencies have good status.

Adaptive frequency hopping as well as packet selection and scheduling policy are capable of reducing the impact of interference, that Bluetooth exhibits on other systems; however only AFH hopping technique can increase the throughput and thus it received a lot of attention recently. Due to this importance, a detailed discussion of AFH follows.

2.3 Adaptive Frequency Hopping Algorithms

This subsection starts with channel classification, a crucial step used in all Bluetooth coexistence mechanisms, including AFH; then it discusses different AFH

[16]Recall that HV1 packets are sent every second slot, thus they occupy the channel 3 times more often than HV3 packets (sent every sixth slot)

algorithms. Emphasis will be on three schemes that will be used to benchmark with our algorithm; these are AFH, RAFH and UBAFH.

2.3.1 Channel Classification

The purpose of channel classification is to determine the quality of the channels based on measurements conducted per frequency. A low-interference channel is classified as "good", while a high-interference channel is classified as "bad".

A number of criteria can be used to distinguish a good channel from a bad one, e.g. Received Signal Strength Indication (RSSI), Packet or Frame Error Rate (PER/FER), and Packet acknowledgement to name a few; these methods may be used separately or jointly.

A brief explanation of these classification methods follows:

- RSSI is an indication of the power level being received by the antenna; the higher the RSSI number (or less negative in some devices), the stronger the signal [17].
- FER is the rate of in-error frames to received frames; a channel is declared bad if its FER exceeds the system defined threshold, which is vendor specific [17].
- Packet acknowledgement (ACK), a built-in ACK mechanism that implicitly provides the status of the channel; if no ACK is received for a frame that requires it, the transmitter infers that the packet sent is lost [17].

The condition of the channels at the transmitter are not necessarily the same at the receiver, therefore there should be a mechanism by which the channel classification information is exchanged in a reliable manner between the transmitter and the receiver. In Bluetooth, the receiver uses the link management protocol (LMP) to send commands to the transmitter, to ensure that the information about its channels are updated [1];

2.3.2 Categories of AFH algorithms

Careful examination of AFH algorithms reveals that they belong to two classes; the first relies on reducing the cardinality of the hop-set, while the second approach relies on probabilistic channel visiting. In addition, these algorithms are optimal either in the presence of static sources of interference (SI) or in the presence of dynamic sources of interference (DI); some AFH algorithms go one step further and mitigate the effect of both SI and DI. Figure 18, illustrates the classification tree of AFH algorithms.

Algorithms Based on Reduced hop-sets avoid bad channels completely; as a result the hop-set consists only of a small number of channels from the available spectrum. Depending on the dynamics of frequency spectrum usage, we distinguish three algorithms:

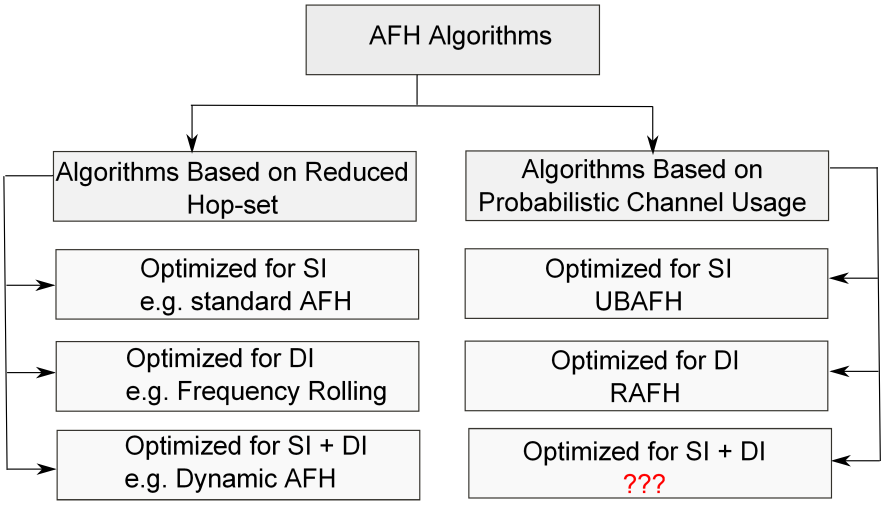

Figure 18: A taxonomy of AFH techniques

- Standard AFH [8] optimal in mitigating the effect of static sources of interference;
- Adaptive Frequency Rolling (AFR) designed to combat interference from dynamic source of interference [26];
- Dynamic Adaptive Frequency Hoping (DAFH) [3] able to mitigate interference from dynamic and static source of interference.

Devices deploying AFH algorithms based on reducing the hop-set size may cause a high level of interference on the available spectrum, which may completely disable the operation of neighbouring devices using these channels. As a result, a new technique based on probabilistic channel visiting is gaining momentum; this approach uses all the channels including bad ones, however channel marked as bad are assigned smaller usage probability that depends on the environmental conditions. This new paradigm reflects the channel condition more accurately and exploits frequency diversity, which is the main principle behind FHSS.

Assigning visiting probability for bad channels, was first introduced in [4]. Independently the authors in [5,27] developed an algorithm named Robust Adaptive Frequency Hopping (RAFH). The authors in [6,28] adopted the probabilistic channel assignment from RAFH and developed a scheme named Utility Based Adaptive Frequency Hopping (UBAFH); they will be discussed in greater details in Section 2.3.4 and Section 2.3.5 respectively.

2.3.3 Standard AFH

In Pseudo Random Frequency Hopping (Bluetooth Version 1.1), the hop sequence generation process happens without consideration for current occupants of the spectrum. Standard Adaptive Frequency Hopping (AFH) addresses these concerns by actively modifying the hopping sequence to use good channels and avoid interference. It is an effective measure in mitigating the interference resulting from frequency static devices such as IEEE 802.11b.

The standard AFH consists of three distinct components shown in Figure 19; the first component of the AFH mechanism is the selection box, which generates the hopping sequence defined in the IEEE Std 802.15.1-2002 [29] as discussed in Section 2.1.

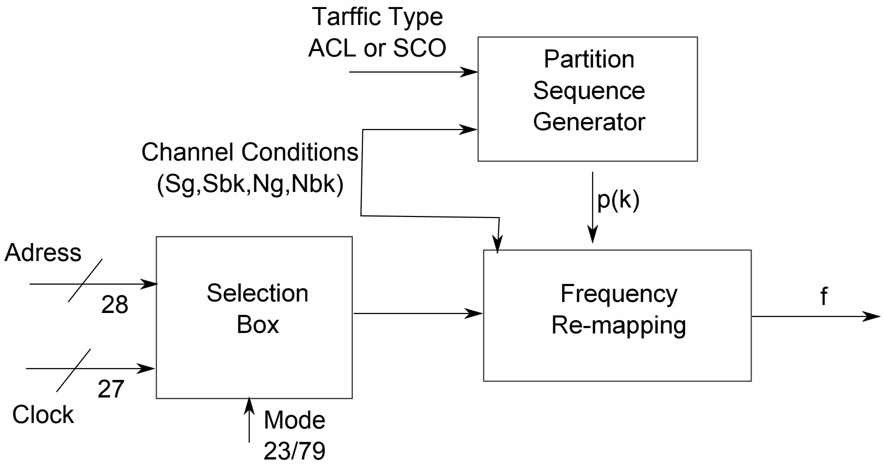

Figure 19: Block diagram of the AFH Mechanism

The second component is the partition sequence generator, which imposes a structure on the original hopping sequence. It divides the set of "bad" channels (S_B), into a set of "bad" channels that are to be kept in the hopping sequence (S_{BK}), and into a set of "bad" channels that are to be removed from the hopping sequence (S_{BR}).

The set S_{BK} is needed in case the size of the set of "good" channels (S_G) is less than the minimum number of hopping channels allowed (N_{min}). The size of each partition is given by the following two equation:

$$\begin{cases} N_{BK} = max(0, N_{min} - N_G) \\ N_{BR} = N_B - N_{BK} \end{cases} \quad (4)$$

The partition sequence generates a flag $p(k)$ at each time slot k to indicate if bad channels can be used or not; $p(k) = 1$ when $N_G < N_{min}$ otherwise $p(k) = 0$.

The third component of the AFH mechanism is the frequency remapping function; it compares the hop frequency generated by the pseudo-random hop selection scheme against the two set of good and bad channels. If the channel belongs to the "good" channel list, it will just be used normally without any special action. On the other hand, if the frequency assigned by the original scheme is included in the "bad" channel list, a remapping function is invoked to substitute the "bad" channel according to the flowchart in Figure 20.

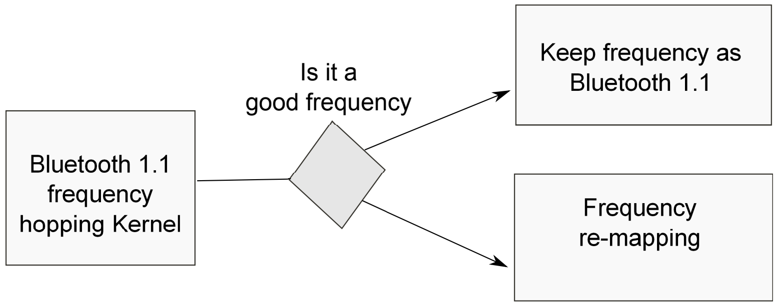

Figure 20: AFH Frequency Decision Flowchart

The remapping function uses $p(k)$ to check if the number of "good" channels is greater than N_{min}; if this is the case then all "bad" channels are remapped to "good" channels; otherwise some of the "bad" channels are still being used in order to conform to the regulation[17].

The Standard Adaptive Frequency Hopping periodically maintains the hop-set to handle changing channel conditions. However it is difficult to decide on the appropriate period T after which bad channels are introduced again in the hop-set; this is because T is closely related to the dynamic characteristics of interference:

- if T is small compared to the change in interference, the performance of AFH will severely decrease because the channel may still remain in the bad state after reset;
- if on the other, T is large compared to interference change, AFH will slowly respond to a change in interference.

Standard AFH enables the Bluetooth link to operate at a high throughput and reliability, because it avoids the occupied spectrum. It has two main limitations thou; first it is specifically designed to coexist with static sources of interference such as 802.11b. The other shortcoming occurs when the number of interferers increases in a given coexistence environment; in this case a great number of channels are removed from the hopping sequence and bad channels would be used, resulting in decrease in throughput and reliability.

2.3.4 Robust Adaptive Frequency Hopping (RAFH)

In contrast to the standard AFH which avoids bad channels completely, Robust Adaptive Frequency Hopping RAFH [5, 27], uses all the available channels (N), but with different usage probabilities; The algorithm starts by estimating the frame error rate of all the channels in the hop-set, i.e. $\overrightarrow{FER}$: $\{FER_1(t), FER_2(t), ...FER_N(t)\}$; t is the time quanta.

[17]The current N_{min} allowed is 15 in USA and 20 Europe

Based on the obtained measurements, RAFH assigns usage probabilities $\overrightarrow{P}$: $\{p_i(t+1)\}$ to the channels; these probabilities are then mapped to a hop sequence to be used in the next time quanta. The steps used in RAFH are shown in Figure 21.

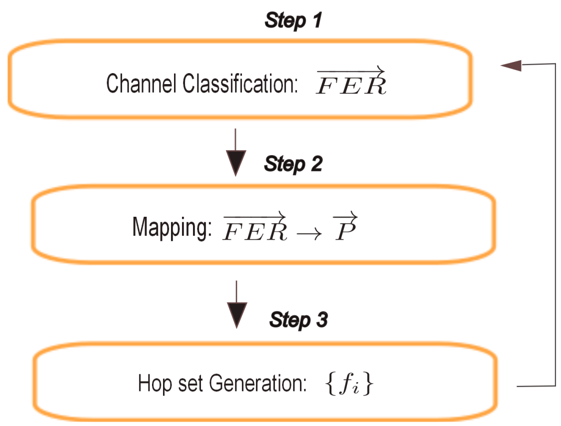

Figure 21: Block Diagram of RAFH

The mapping from $\overrightarrow{FER}$ to the hop probabilities $\overrightarrow{P}$ is subject to two conditions:

- the average frame error rate has to be below a given threshold, ξ i.e.

$$\sum p_i(t+1) * FER_i(t) < \xi \tag{5}$$

- the entropy, $H(\overrightarrow{P})$, of the obtained hop probabilities, is maximized i.e.

$$\max H(\overrightarrow{P}) \equiv -\sum_{i=1}^{N} p_i(t+1) * \log p_i(t+1) \tag{6}$$

The first condition i.e. Equation (5), ensures robustness of the link between the transmitter and receiver, since when the average FER exceeds the threshold, the receiver will not be able to recover the signal successfully, which results in system drop out in that interval.

Frame error rate, $FER_i(t)$, is based on the current measurement, therefore the constraint on the total FER does not provide robustness against randomness caused by a change in interference. To tackle this issue, RAFH uses the principle of maximum entropy [30], which states that among the probability distributions that satisfy Equation (5), the distribution with maximum entropy i.e. Equation (6), is the least surprising in terms of predictions.

The overall entropy maximisation problem for updating the hop probabilities $\overrightarrow{P}$ follows:

$$\max H(\overrightarrow{P}) \equiv -\sum_{i=1}^{N} p_i(t+1) * \log p_i(t+1) \tag{7}$$

subject to

$$\begin{cases} \sum_{i=1}^{N} p_i(t+1) * FER_i(t) < \xi \\ \sum_{i=1}^{N} p_i(t+1) = 1 \\ 0 \leq p_i(t+1) \leq 1 \end{cases}$$

The constrained convex optimisation in problem (7) can be efficiently solved using Lagrangian duality [31]. The pseudo code used in RAFH to solve Equation (6) is shown in Figure 22.

```
INPUT: A, ξ, ε, α // ε is a stopping error and α is a step size
OUTPUT: p
C ← ξ − A; // C = [c_1 ... c_M]^T := [ξ − a_1 ... ξ − a_M]^T, M by 1 column vector
λ ← 0; // Initial λ
if Σ_{i=1}^{M} c_i < 0 then
    while |Σ_{i=1}^{M} c_i e^{c_i λ}| > ε do
        λ ← λ − α Σ_{i=1}^{M} c_i e^{c_i λ}
    end while
end if
ν ← log Σ_{i=1}^{M} e^{−a_i λ} − 1
for i = 1 to M do
    p_i ← e^{−(a_i λ + ν + 1)}
end for
Return p
```

Figure 22: Pseudo-Code to solve Equation (7) (source [5])

The performance of this approach has been investigated by means of simulations, and the authors showed that RAFH outperforms both traditional frequency hopping as well as the standard AFH algorithm, with respect to frame error rates and stability.

Figure 23 illustrates the results obtained in the presence of dynamic sources of interference.

RAFH is an appealing mechanism since it exploits frequency diversity, combats dynamic interference and to a certain extent deals with static interference. However, the main shortcoming of RAFH is its computational complexity; this is due to the way the probably distribution is calculated, which lead to a convex optimization problem, that requires a lot of computation not suitable for low power devices.

2.3.5 Utility Based Adaptive Frequency Hopping (UBAFH)

The authors in [6,28] developed a probabilistic channel visiting algorithm named Utility Based Adaptive Frequency Hopping (UBAFH) that has lower computa-

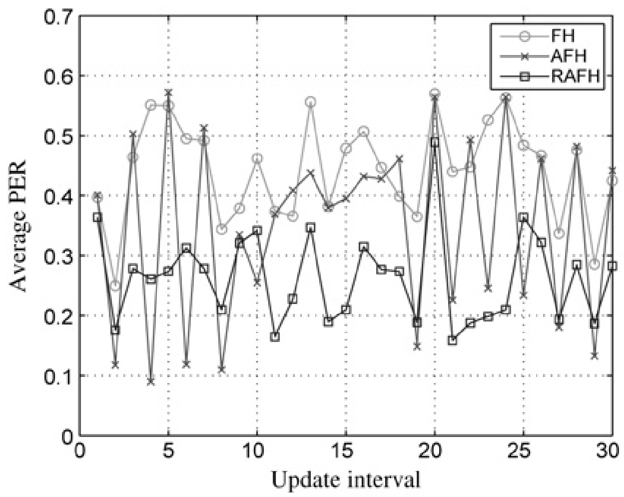

Figure 23: RAFH Performance: PER vs time (source [5])

tional complexity than RAFH. The trade off is that their scheme is suboptimal since does not consider the constraint on the average frame error rate.

UBAFH follows the same steps taken by RAFH, shown earlier in Figure 21. The fundamental difference between the two schemes lies in the procedure through which the estimated frame error rates are mapped into a probability mass function.

UBAFH assigns channel usage probabilities using a much simpler mapping function than RAFH; it does not take into account the constraint on the average FER, neither does it attempt to combat for unknown future interference.

Mapping the estimated frame error rates into a probability mass function is done according to the following procedure:

$$p_i = (1 - FER_i)^\kappa / (\sum_{i=1}^{N} (1 - FER_i)^\kappa) \tag{8}$$

The authors of UBAFH introduced a lower and upper bounds on the usage probabilities to pseudo-comply with spectrum regulations i.e. $P_{MIN} \leq p_i \leq P_{MAX}$.

P_{MAX} prevents the algorithm from converging to scenarios where only a few channels are used; P_{MIN} ensures a certain minimum degree of frequency diversity.

The parameter κ in Equation (8), is called a temperature by the authors and can be tuned to achieve different behaviours.

The performance of UBAFH was studied in the presence of frequency static interfering activities in [6]; the authors showed that UBAFH outperforms standard AFH. The performance metric is the total energy spent by the different communication systems for the successful delivery of specified amount of data. These results are therefore valid for channel scenarios where nodes experience only static sources of interference.

In [28], the behaviour of UBAFH over frequency selective fading channels is compared with AFH, as well as random frequency hopping. UBAFH outperforms both approaches with respect to the achieved packet error rate and throughput.

3 Algorithm Description

As mentioned in Section 2.3, probabilistic channel usage has many advantages including the exploitation of frequency diversity; however the two known algorithms in this category i.e. RAFH [5,27] and UBAFH [6,28] are optimal in the presence of **either** static or dynamic sources of interference, but not both; optimality is in terms of mitigating the effects of interfering devices.

Our developed algorithm is named smooth adaptive frequency hopping (SAFH) for reasons that become clear at the end of this section. SAFH is inspired by RAFH, and it developed to rectify the shortcomings of existing AFH, as a consequence we had three key design requirements in mind:

- ability to combat both static **and** dynamic sources of interference
- low computation complexity
- better performance than RAFH and UBAFH.

There are four main elements of the adaptive hopping procedure as illustrated by the block diagram in Figure 24: channel classification, channel prediction, probability mass function (PMF) determination and hop-set generation; these steps will be explained in greater detail in the following sub-sections.

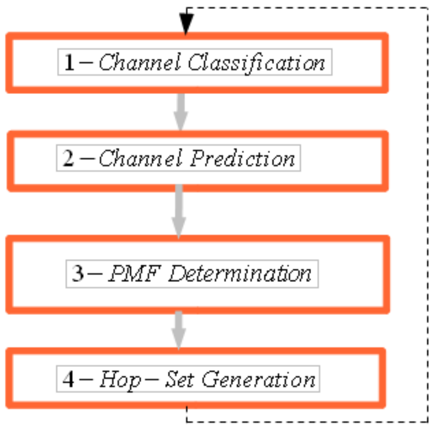

Figure 24: Block Diagram of the Algorithm

3.1 Channel Classification

The ability to detect the presence of other systems operating in the band is central to most interference mitigation techniques including SAFH, hence the algorithm starts by assessing the ISM spectrum to determine the presence of interference. The performance metric used is the frame error rate (FER), which can be calculated either instantaneously or by means of a time average.

Instantaneous measurements would result in overhead in the synchronization between the transmitter and the receiver and consequently in more power consumption; in addition there is a risk of incorrect classification due to instantaneous disturbances e.g. other frequency-hoppers [1]. To avoid these disadvantages, we adopted the use of time average, especially that this approach is recommended by the IEEE 802.15 wireless personal area networks Task Group (TG2) [1].

At each classification quantum t (equal to the time needed to use all the frequencies in the hop-set), a list of measurements is compiled for all N channels. The frame error rate of channel i is $FER_i(t)$; it is the ratio of erroneous frames $N_i^e(t)$, to the transmitted frames $N_i^{tr}(t)$; this is shown in line 10 of pseudo code (1) below. The way the average frame error rate $\overline{FER}(t)$ is calculated, is shown in line 12.

Pseudo Code 1 Channel Classification

1: **repeat**
2:   **if** Channel i is used **then**
3:     increment corresponding number of transmitted frames: $N_i^{tr}(t) \leftarrow N_i^{tr}(t) + 1$
4:     **if** Error in transmission **then**
5:       increments corresponding number of erroneous frames: $N_i^e(t) \leftarrow N_i^e(t) + 1$
6:     **end if**
7:   **end if**
8: **until** All channels in the hop-set are used
9: **for** $i = 1$ **to** N *(number of available channels)* **do**
10:   $FER_i(t) \leftarrow N_i^e(t)/N_i^{tr}(t)$
11: **end for**
12: `Average` FER : $\overline{FER}(t) \leftarrow \sum_{i=1}^{N} N_i^e(t)/\sum_{i=1}^{N} N_i^{tr}(t)$

Note that while evaluating SAFH, we assumed complete synchronization between the transmitter and the receiver i.e. both nodes have the same channel conditions; this action is justified since our focus is to compare the performance of different algorithm and therefore, implementation of a link layer adds to the development time and shifts our focus.

One more thing to highlight is that even thou channel classification based on FER method is simple and straightforward, it cannot directly distinguish whether the bad channel is due to interference or some other channel impairments [1]; different metrics can be used jointly to determine if errors are due to noise (i.e. poor SNR) or interference (i.e. poor SIR) [17]; here's how it works:

- if the packet is not decoded successfully while RSSI is low, then the error(s) nature is propagation effects;
- on the other hand if the packet is not decoded successfully but RSSI is high, then the error(s) nature is interference [1].

After classifying all the channels, the algorithm calculates the average FER at that interval (line 12 of the pseudo code), and checks if it exceeds the system threshold ξ. If this is the case, then the second step of the algorithm - discussed shortly- is invoked.

3.2 Channel Prediction

This step is crucial, because the measured FER does not account for random arrivals and departures of interfering sources. In order to compensate for the FER measurement error and combat for unknown future interference, SAFH predicts the conditions of all the channels, i.e. the respective frame error rates, for the following interval $FER_i'(t+1)$.

This is a mapping from $FER_i(t)$ to $FER_i'(t+1)$, and it is achieved by exponential smoothing [32], as shown in Equation (9); this popular forecasting tool, first used by Brown to track the location of submarines, computes a moving average, where all measurements contribute to the smoothed value and decrease exponentially.

$$\begin{cases} FER_i'(t+1) & = \alpha \cdot FER_i(t) + (1-\alpha) \cdot FER_i'(t) \\ FER_i'(1) & = FER_i(0) \longrightarrow \text{(initial condition)} \end{cases} \tag{9}$$

Equation (10) is a rearrangement of Equation (9); it illustrates how the predicted frame error rate at time $(t+1)$ is indeed the forecast at time t, plus an adjustment for the error that occurred in the last forecast.

$$FER_i'(t+1) = FER_i'(t) + \alpha \cdot (FER_i(t) - FER_i'(t)) \tag{10}$$

The balance between new and old data is controlled by the smoothing factor α, which ranges between 0 and 1. When α approaches 1, the filter gives more weight to recent data and has less of a smoothing effect; however when α approaches 0, the effect of the current observation is ignored and only the smoothed past is retained.

Substituting for $FER_i'(t)$ in the defining Equation (9), we obtain:

$$\begin{cases} FER_i'(t+1) = \\ \alpha \cdot FER_i(t) + (1-\alpha) \cdot FER_i'(t) = \\ \alpha \cdot FER_i(t) + (1-\alpha) \cdot (\alpha \cdot FER_i(t-1) + (1-\alpha) \cdot FER_i'(t-1)) = \\ \alpha \cdot FER_i(t) + \alpha \cdot (1-\alpha) \cdot FER_i(t-1) + (1-\alpha)^2 \cdot FER_i'(t-1)) \end{cases} \tag{11}$$

By substituting for $FER_i'(t-1)$, then for $FER_i'(t-2)$, and so forth, until we reach $FER_i'(1)$ (which is just $FER_i(0)$), the expanding equation can be written in a more compact form:

$$\begin{cases} FER_i'(t+1) & = \alpha \cdot \sum_{i=0}^{t-1} (1-\alpha)^i \cdot FER_i(t-i) + (1-\alpha)^t \cdot FER_i'(1) \end{cases} \tag{12}$$

Equation (12), shows that in exponential smoothing, all previous measurements contribute to the smoothed value, but their contribution is suppressed by increasing powers of the parameter α.

The predicted FER is used to assign usage probabilities to all the channel, for the next interval i.e. $(t+1)$. The details follows in the next subsection.

3.3 Probability Mass Function Determination

The third step of SAFH maps the predicted $FER'_i(t+1)$ to a discrete probability distribution, such as:

Condition 1. *The probability assigned to a channel is a decreasing function of its FER i.e.*
If $FER'_i(t+1) \geq FER'_j(t+1)$ then $p_i(t+1) \leq p_j(t+1)$, $i, j \in 1..N$

Condition 2. *The target average frame error rate $\overline{FER}(t+1)$ must not exceed a threshold (ξ) as shown in Equation (13); this parameter is specific to the type of application e.g. in Bluetooth voice communication $5\% \leq \xi \leq 10\%$ [33];*

$$\sum_{i=1}^{N}(p_i(t+1) \cdot FER'_i(t+1)) \leq \xi \tag{13}$$

The first condition results in channels with good condition, being used more often than bad channels; this technique exploits frequency diversity, which is the key advantage behind frequency hopping spread spectrum, and reflects channel conditions more accurately than the standard AFH which avoids bad channels completely.

The second condition ensures robustness of the link between the nodes since it results in less system drop out as explained earlier in Section 2.3.4.

A reasonable mapping function that fulfils the above conditions is shown in Equation (14):

$$\begin{cases} p_i(t+1) & = (\beta(t+1) + c \cdot d_i(t+1))/\delta(t+1) \text{ if } d_i(t+1) \geq 0 \\ p_i(t+1) & = (\beta(t+1) + s \cdot d_i(t+1))/\delta(t+1) \text{ if } d_i(t+1) \leq 0 \end{cases} \tag{14}$$

In what follows, we will explain the meaning of the following terms $d_i(t+1), c, s, \delta(t+1)$ and $\beta(t+1)$.

The term $d_i(t+1)$, is the difference between the predicted frame error rate of channel i and the threshold ξ, i.e.

$$d_i(t+1) = \xi - FER'_i(t+1) \tag{15}$$

A positive value of this metric indicates that we have a good channel e.g. channel i in Figure 25. Channel j is considered bad since $d_j(t+1) = \xi - FER'_j(t+1)$ is negative.

We also use two optional input parameters, c and s; they determine how good channels are rewarded and bad channels are punished; the naming of c and s, is

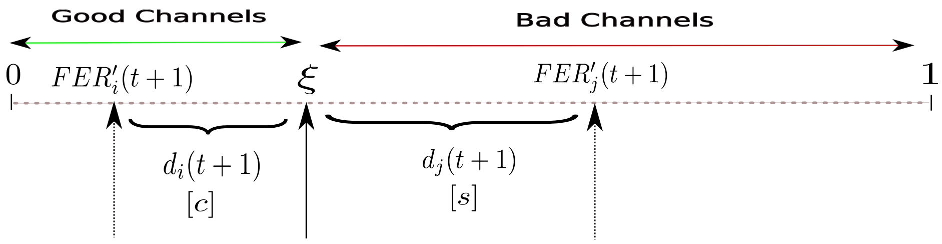

Figure 25: Mapping of FER into probability mass function

a reminder that we are using a carrot-stick approach. In case we do not specify the value of either parameters, the algorithm uses the default value of 1.

The parameters c and s are input to the algorithm at start up, and retain their values throughout the operation. In the future there will be further investigation to modify them at run time. In that case they become a function of time themselves i.e. $c(t)$ and $s(t)$ instead of c and s.

The term $\delta(t+1)$ is a normalizing factor as shown in Equation (16); it ensures that $\sum_{i=1}^{N} p_i(t+1) = 1$; moreover it guarantees that $p_i(t+1) \leq 1$.

$$\delta(t+1) = \sum_{i=1}^{N_g(t+1)} (\beta(t+1) + c \cdot d_i(t+1)) + \sum_{i=1}^{N_b(t+1)} (\beta + s \cdot d_i(t+1)) \qquad (16)$$

where $N_g(t+1)$ and $N_b(t+1)$, are the predicted numbers of good channels and bad ones respectively.

The term $\beta(t+1)$ is calculated by plugging the values of $d_i(t+1)$ and $\delta(t+1)$ from Equation (15) and Equation (16) respectively, into Equation (14); then solve the following:

$$\sum_{i=1}^{N} (p_i(t+1) \cdot FER'_i(t+1)) = \xi \qquad (17)$$

Illustrative Example

Before proceeding, we will illustrate how the probabilities are calculated. For simplicity, only four channels are considered. Let their frame error rates be $\{0.16, 0.2, 0.18, 0.14\}$, the threshold $\xi = 0.15$, $c = 10, s = 1$ and the smoothing parameter $\alpha = 1$, i.e. history is ignored entirely.

First $FER'_i(t+1)$ and $d_i(t+1)$ are calculated using Equation (9) and Equation (15) respectively; Table 2 shows that there is one good channel and three bad channels.

Table 2: Mapping $FER'_i \rightarrow p_i$

$FER_i(t)$	$FER'_i(t+1)$	$d_i(t+1)$	Status
0.16	0.16	-0.01	Bad
0.2	0.2	-0.05	Bad
0.18	0.18	-0.03	Bad
0.14	0.14	0.01	Good

Substituting the obtained values (in Table 2) into Equation (16), we obtain:

$$\begin{cases} \delta(t+1) & = (\beta(t+1) + c \cdot d_4(t+1)) + \sum_{i=1}^{3}(\beta(t+1) + s \cdot d_i(t+1)) \\ & = 4 \cdot \beta(t+1) + c \cdot d_4(t+1)) + s \cdot d_1(t+1) + s \cdot d_2(t+1) \\ & +s \cdot d_3(t+1) \\ & = 4 \cdot \beta(t+1) + 10 \cdot 0.01 + 1 \cdot (-0.01) 1 \cdot (-0.05) 1 \cdot (-0.03) \\ & = 4 \cdot \beta(t+1) + 0.01 \end{cases} \tag{18}$$

Now the substitution of $\delta(t+1)$ into Equation (14) results in:

$$\begin{cases} p_1(t+1) & = (\beta(t+1) - 1 \cdot 0.01)/(4 \cdot \beta(t+1) + 0.01) \\ & = (\beta(t+1) - 0.01)/(4 \cdot \beta(t+1) + 0.01) \end{cases} \tag{19}$$

$$\begin{cases} p_2(t+1) & = (\beta(t+1) - 1 \cdot 0.05)/(4 \cdot \beta(t+1) + 0.01) \\ & = (\beta(t+1) - 0.05)/(4 \cdot \beta(t+1) + 0.01) \end{cases} \tag{20}$$

$$\begin{cases} p_3(t+1) & = (\beta(t+1) - 1 \cdot 0.03)/(4 \cdot \beta(t+1) + 0.01) \\ & = (\beta(t+1) - 0.03)/(4 \cdot \beta(t+1) + 0.01) \end{cases} \tag{21}$$

$$\begin{cases} p_4(t+1) & = (\beta(t+1) + 10 \cdot 0.01)/(4 \cdot \beta(t+1) + 0.01) \\ & = (\beta(t+1) + 0.1)/(4 \cdot \beta(t+1) + 0.01) \end{cases} \tag{22}$$

The value $\beta(t+1)$ is calculated by solving the following equation:

$$\begin{cases} \sum_{i=1}^{N}(p_i(t+1) \cdot FER'_i(t+1)) = \xi = 0.15 \ \Rightarrow \\ \beta(t+1) = 0.0562 \end{cases} \tag{23}$$

The obtained value of $\beta(t+1)$ is back substituted in Equation (19), Equation (20), Equation (21) and Equation (22), to obtain the following usage probabilities:

$$\begin{cases} p_1(t+1) = (0.0562 - 0.01)/(4 \cdot 0.0562 + 0.01) = 0.197 \\ p_2(t+1) = (0.0562 - 0.05)/(4 \cdot 0.0562 + 0.01) = 0.027 \\ p_3(t+1) = (0.0562 - 0.03)/(4 \cdot 0.0562 + 0.01) = 0.111 \\ p_4(t+1) = (0.0562 + 0.1)/(4 \cdot 0.0562 + 0.01) = 0.665 \end{cases} \tag{24}$$

Note that for the same frame error rates, RAFH discussed in Section 2.3.4, assigns the following probabilities $\{0.236, 0.031, 0.086, 0.647\}$. Figure 26 shows that when our algorithm does not use prediction (i.e. $\alpha = 1$), it assigns usage probabilities to the channels comparable to RAFH. Consequently SAFH will perform as good as RAFH in its worst case scenario.

Figure 26 also shows that in contrast to SAFH and RAFH, the standard adaptive frequency algorithm blacklisted the second channel, even thou its frame error rate ($FER_2 = 0.16$) is slightly above the the threshold $\xi = 0.15$.

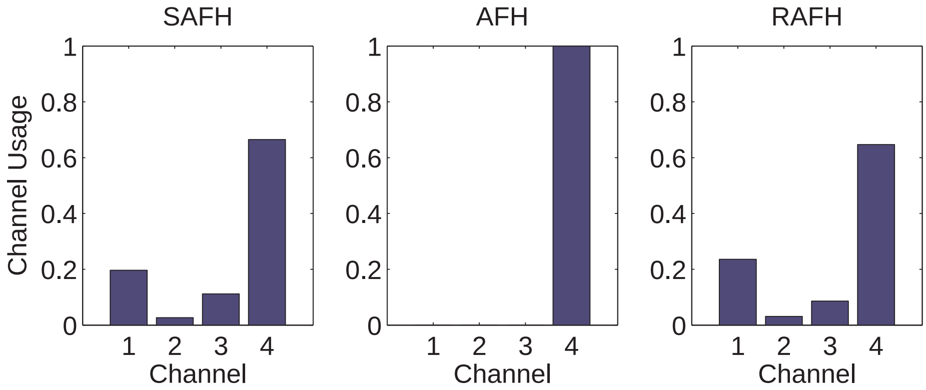

Figure 26: Channel Usage of SAFH, RAFH and the standard AFH

Another benefit[18] of SAFH (and RAFH) is that it reduces the self-interference i.e. between similar networks, such as two collocated Bluetooth piconets deploying the same AFH scheme. The collision probabilities of SAFH and RAFH are much smaller than the standard AFH, as shown below. This is an important issue in Bluetooth networks, since piconets are deployed without planning.

- The collision probability for SAFH is:

$$[0.197, 0.027, 0.111, 0.665] \cdot [0.197, 0.027, 0.111, 0.665]^T = 0.494$$

- The collision probability for RAFH is:

$$[0.236, 0.031, 0.086, 0.647] \cdot [0.236, 0.031, 0.086, 0.647]^T = 0.483$$

- The collision probability for standard AFH is:

$$[0, 0, 0, 1] \cdot [0, 0, 0, 1]^T = 1$$

Special Cases

The obtained value of beta is a function of the predicted frame error rate, therefore it is calculated every time the algorithm is invoked; SAFH provides option to fix β at run time.

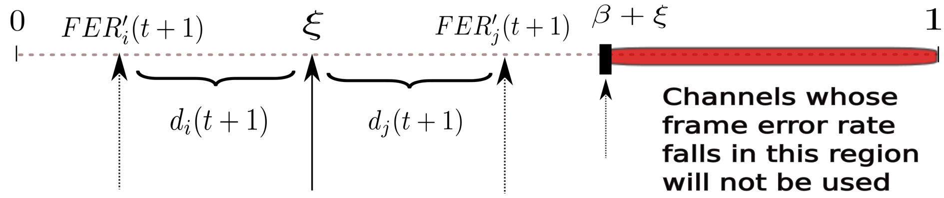

Figure 27: Mapping of FER when β is constant

When $c = s = 1$, the possible values of β range between 0 and $1-\xi$; the channels whose FER exceeds $\beta + th$ will not be used, since otherwise the probability would be negative.

We tested different values for β, the result will be shown in a later section; the special case when $c = s = 1$ and $\beta = 1 - \xi$ results in the probabilities $p_i = (1 - FER_i)/\delta$; this is the same probability mass function generated by the UBAFH algorithm, described in Section 2.3.5, with temperature $\kappa = 1$

Fixing the value of β results in less computation at the price of obtaining a sub optimum solution, since we can not guarantee any more that the average FER would be below the threshold. In addition the channels whose FER exceeds $(\beta + \xi)$ are blacklisted as shown in Figure 27, which implies that frequency diversity would not be exploited to its full extent.

Pseudo Code

The pseudo code for generating the probability mass function follows. First of all the algorithm requires that at least on channel has FER smaller than the threshold ξ. If that is not the case, SAFH would report the situation to a higher layer[19], since it would be impossible to find a probability distribution that satisfies the condition represented by Equation (13). Next, SAFH reads the values of c, s and β as indicated in pseudo code 2 (lines 1,2 and 10). Assigning -1 to these parameters (lines 3,7 and 11), instructs SAFH to use default actions. Lines $16-27$ calculates the probabilities for good and bad channels using Equation (14).

[18]adopted from [27]

[19]Upper layers can e.g. defer transmission or use only a subset of the channels that experience least interference, as suggested in [27]

Pseudo Code 2 Mapping $FER_i \rightarrow p_i$

Require: `At least one channel` i `/` $FER'_i(t+1) \leq \xi$

`read threshold` ξ
`read parameter` c
if $c = -1$ **then**
  `Use use default value i.e.` $c = 1$
end if
`read parameter` s
if $s = -1$ **then**
  `Use use default value i.e.` $s = 1$
end if
`read parameter` β
if $\beta = -1$ **then**
  `calculate` $\beta(t+1)$ `by solving Equation (17)`
else
  `Use input value of` β `to calculate` $p_i(t+1)$
end if
for $i = 1$ **to** N_g **do**
  $p_i(t+1) = \beta(t+1) + c \cdot d_i(t+1)$ {//probabilities not Normalized yet }
end for
for $i = 1$ **to** N_b **do**
  $p_i(t+1) = \beta(t+1) + s \cdot d_i(t+1)$ {//probabilities not Normalized yet }
  if $p_i(t+1) < 0$ **then**
    $p_i(t+1) \leftarrow 0$
  end if
end for
for $i = 1$ **to** N **do**
  $p_i(t+1) = p_i(t+1) / \sum_{i=1}^{N} p_i(t+1)$ {//Normalize the probabilities }
end for

The way the probabilities obtained are mapped to a hop-set is explained in the next sequel.

3.4 Hop-set Generation

Different techniques can be used for hop-set generation, e.g. direct methods [34] and acceptance-rejection methods [34, 35]. SAFH adopts the inversion methods [34,35] which is based on the observation that continuous cumulative distribution functions (CDF) range uniformly over the interval $(0, 1)$. If u is a uniform random number on $(0, 1)$, then using $X = F^{-1}(u)$ generates a random number X from a continuous distribution with specified CDF F.

Inversion methods also work for discrete distributions [34, 35] and two steps are needed to generate a random number X from a discrete distribution, with probability mass vector $P(X = i) = p_i$:

- Generate a uniform random number u on $(0, 1)$

- Obtain x by a monotone transformation of u as follows:

$$X = i \ \ if, \ \sum_{j=1}^{i-1} p_j < u < \sum_{j=1}^{i} p_j \tag{25}$$

It is clear from the inequality shown above, that $P(X = i) = \sum_{j=1}^{i} p_j - \sum_{j=1}^{i-1} p_j$. The solution of Equation (25) is uniquely defined with probability one, and can always be obtained in finite time [35]; i represents the channel number i.e. $1 \leq i \leq N$.

Example

Pseudo code 3 illustrates how to generate a hop-set from four channels. We will use the same values calculated in the first example i.e. $\{p_1 = 0.1970, p_2 = 0.027, p_3 = 0.111, p_4 = 0.665\}$. When the the value of the generated random number u falls between 0 and $p_1 = 0.1970$ (line 2), then channel 1 is added to the hop-set (line 3). Similarly lines 4,6 and 8 indicate how the other channels are added to the hop-set.

Pseudo Code 3 Inversion Methods for Simple distribution

1: `Generate a random number` u
2: **if** $0 \leq u < p_1 = 0.197$ **then**
3:   `set` $X = 1$ `(Add channel 1 to the hop-set)`
4: **else if** $p_1 = 0.197 \leq u < p_1 + p_2 = 0.224$ **then**
5:   `set` $X = 2$ `(Add channel 2 to the hop-set)`
6: **else if** $p_1 + p_2 = 0.224 \leq u < p_1 + p_2 + p_3 = 0.335$ **then**
7:   `set` $X = 3$ `(Add channel 3 to the hop-set)`
8: **else**
9:   `set` $X = 4$ `(Add channel 4 to the hop-set)`
10: **end if**

The amount of time to generate random variate is proportional to the number of intervals one must search. Thus a more efficient way that will result in a reduced search time, is to sort the probabilities in decreasing order first, as shown in the pseudo code below.

Pseudo Code 4 A more Efficient Way to Generate Hop Frequency

```
Generate a random number u
if 0 ≤ u < p4 = 0.665 then
   set X = 4 (Add channel 4 to the hop-set)
else if p4 = 0.665 ≤ u < p4 + p1 = 0.862 then
   set X = 1 (Add channel 1 to the hop-set)
else if p4 + p1 = 0.862 ≤ u < p4 + p1 + p3 = 0.973 then
   set X = 3 (Add channel 3 to the hop-set)
else
   set X = 2 (Add channel 2 to the hop-set)
end if
```

In our implementation the amount of time to generate random variate is $O(n)$, because we used linear search; if we used binary search instead, the search time would have been reduced to $O(\log(n))$.

3.5 SAFH in a Nutshell

SAFH has four major steps; at start up it reads the values of the input parameters α, ξ, c, s and β, shown in Figure 28. The algorithm then assigns equal probability to all the channels (line 2 in pseudo code 5 below), and generates an initial hop-set based on that. At every interval t (equal to the time needed to uses all the frequencies in the hop-set), SAFH measures the FER of all the channels $FER_i(t)$, and calculates the average $\overline{FER}$ (lines 5-9).

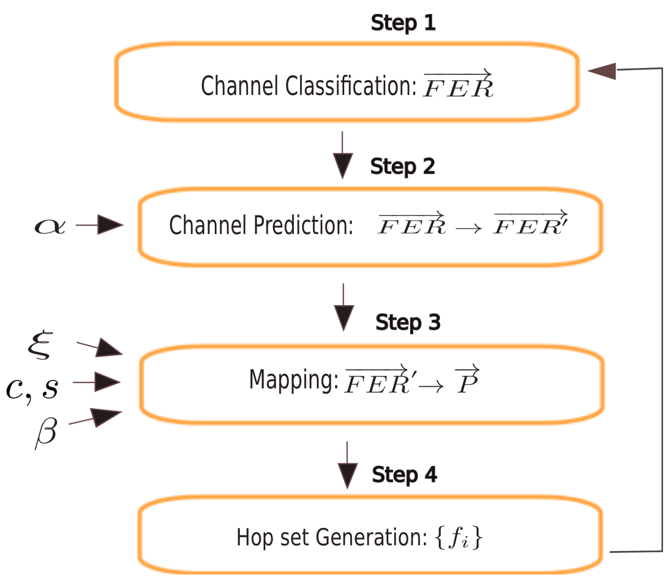

Figure 28: Block Diagram of the SAFH Algorithm

The other steps of SAFH are invoked when the average frame error rate $\overline{FER}$ exceeds the predefined threshold (ξ), *and* at least one of channels has frame error rate below the threshold; in that case the algorithm uses exponential smoothing to predict $FER'_i(t+1)$, updates the hopping probability and generates a new hop-set (lines 12-14).

The fact that the input parameters α, ξ, c, s, β can be changed on the fly, can be exploited by a higher layer. An interesting use case, is a cognitive radio that adapts these parameters to the dynamic environment, to maximize the utilization of the radio resources.

Pseudo Code 5 Summary of the key steps used by SAFH

Ensure: (Average FER) $\leq \xi$
1: read parameters: c, s, α, ξ, β
2: $p_i \leftarrow \frac{1}{N}$:All channels start with equal p_i
3: generate initial hop-set
4: **loop**
5:   **for** $i = 1$ to (hop-set-length) **do**
6:     use channel indexed by i
7:     update FER of the channel used
8:     update average FER
9:   **end for**
10:   **if** (Average FER) $> \xi$ **then**
11:     **if** At least one channel i / $FER'_i(t+1) \leq \xi$ **then**
12:       predict $FER'_i(t+1)$, for all the channels
13:       recalculate $p'_i(t+1)$, for all the channels
14:       generate the new hop-set
15:     **else**
16:       Inform upper layers
17:     **end if**
18:   **end if**
19: **end loop**

4 Performance Analysis

In this Section, the impact of our algorithm on the performance of Bluetooth, adopting standard AFH (BT), 802.11b (WLAN) and 802.15.4 (ZigBee) is studied; The analysis provides details on the following metrics: probability of collision and packet loss, and shows that devices adopting our scheme will be good neighbours in the 2.4 GHZ ISM band; moreover we will capture the impact of other wireless devices on our algorithm.

The results obtained however, provide only a low order approximation [36] on the impact of interference, due to the many assumptions that are made during the analysis[20].

4.1 System Model

We consider the following node topology:

- Two BT devices (a master and a slave) that adopt our algorithm, we will call them SAFH nodes; theycommunicate using one time slot voice packets (HV1, HV2 and HV3) and each time slot consists of data portion followed by an idle period.
- Two wireless neighbours (e.g. WLAN, BT, ZigBee) where one is the transmitter and the other is the receiver as shown in Figure 29 below.

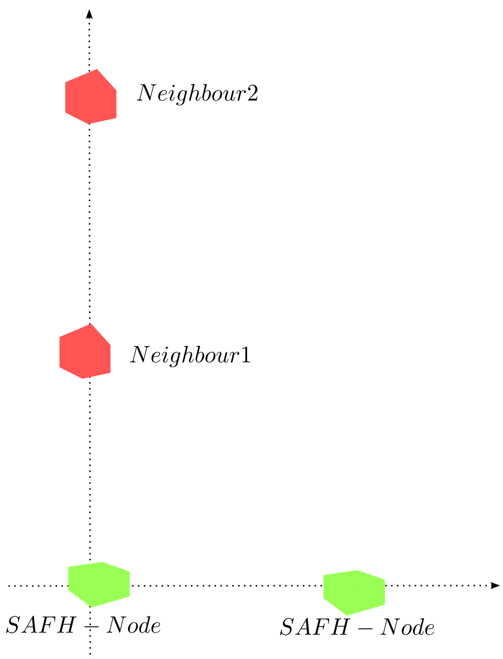

Figure 29: Topology: Two WLAN and Two SAFH devices.

[20]e.g. Interference from adjacent-channels is not considered

The following scenarios are considered:

- Neighbour 1 is the source and Neighbour 2 is the sink; in this scenario, both SAFH nodes will suffer from interference.
- Neighbour 2 generates traffic to Neighbour 1; SAFH will interfere the transmission due to its close vicinity from Neighbour 1; therefore this scenario deals with the effect of SAFH nodes on their wireless Neighbours.

4.2 Probability of Collision P(C)

A collision occurs when the desired signal overlaps in time *and* in frequency with an interfering signal, therefore:

$$P(C) = P(overlap\,in\,time) * P(overlap\,in\,frequency)$$

4.2.1 SAFH in the presence of WLAN

To derive $P(overlap\,in\,frequency)$, we recall that WLAN (802.11b) consists of three non overlapping channels with a bandwidth of $22MHZ$ each; in this topology only one channel is occupied by the WLAN as a result,

$$P(overlap\,in\,frequency) = \sum_{i=c}^{i=c+21} P_i \tag{26}$$

Where c is the first channel occupied by the WLAN, and P_i is the probability that SAFH is using channel i.

In the next subsections we will derive P(overlap in time) for both radio systems; This analysis is similar to the discussion in [37], except that we assign non uniform probabilities to the channels.

SAFH interfered by WLAN We start the analysis for a 100% traffic load[21] and extend the result to accommodate different duty cycles; Let T_{SAFH}, T_W and T_{Boff} be the time of SAFH packet, the time of WLAN packet and the back-off time of the WLAN, respectively.

From Figure 30, Figure 31 we can make two observations:

- When $T_{SAFH} \geq T_{Boff}$, there would be always an overlap in time, and therefore $P(C) = \sum_{i=c}^{i=c+21} P_i$.
- When SAFH transmission time is smaller than the WLAN backoff time i.e. $T_{SAFH} < T_{Boff}$; there would not be overlap in time, if the SAFH packet hops during the back-off period.

  Let $\tilde{M}$ be the number of times packets overlap in time.

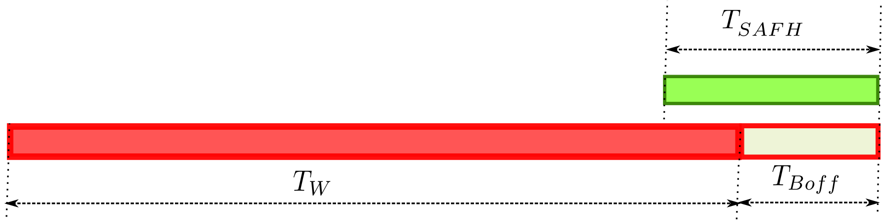

Figure 30: SAFH interfered by WLAN.

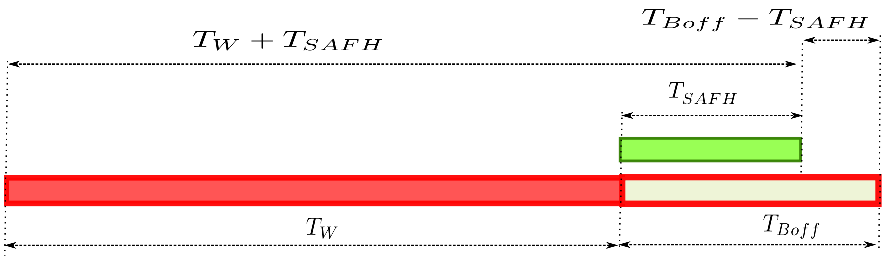

Figure 31: SAFH interfered by WLAN.

$\tilde{M}$ assumes two values:

$$\begin{cases} \tilde{M} & = 1 \quad \text{with probability} \quad \frac{T_W + T_{SAFH}}{T_{WI}} \\ \tilde{M} & = 0 \quad \text{with probability} \quad \frac{T_{Boff} - T_{SAFH}}{T_{WI}} \end{cases} \tag{27}$$

Where $T_{WI} = T_W + T_{Boff}$ is the inter arrival time of WLAN packets. Using the theorem of total probability $P(C) = \sum_m \left[P(\tilde{M} = m)(1 - (1 - p)^m) \right] \Longrightarrow$

$$\begin{cases} P(C) & = \frac{T_W + T_{SAFH}}{T_{WI}} \cdot (1 - (1 - \sum_{i=c}^{i=c+21} P_i)^1) + \frac{T_{Boff} - T_{SAFH}}{T_{WI}} \cdot (1 - 1) \\ & = \frac{T_W + T_{SAFH}}{T_{WI}} \cdot (\sum_{i=c}^{i=c+21} P_i) \end{cases} \tag{28}$$

For arbitrary traffic load $P_L \leq 100\%$, we multiply the results obtained in (28) by P_L. Here's the final result for the probability of collision of SAFH interfered by WLAN:

$$\begin{cases} P(C) & = P_L \cdot (\sum_{i=c}^{i=c+21} P_i) \quad \text{when} \quad T_{SAFH} \geq T_{Boff} \\ P(C) & = P_L \cdot \left(\frac{T_W + T_{SAFH}}{T_{WI}} \right) \cdot (\sum_{i=c}^{i=c+21} P_i) \quad \text{when} \quad T_{SAFH} < T_{Boff} \end{cases} \tag{29}$$

WLAN interfered by SAFH We again assume 100% duty cycle, and accommodate for different traffic load later. The WLAN packet time T_W is assumed to be larger than the inter arrival time of SAFH packets, T_{SAFH}^{II}, as illustrated in Figure 32. Let $T_I = T_{SAFH}^{II} - T_{SAFH}$, be the idle time and $R = T_W - \lfloor \frac{T_W}{T_{SAFH}^{II}} \rfloor \cdot T_{SAFH}^{II}$, the residual; where $\lfloor x \rfloor$ is the largest integer $\leq x$.

[21] Also referred to as the duty cycle; it measures the amount traffic sent as a percentage of the total capacity of the channel.

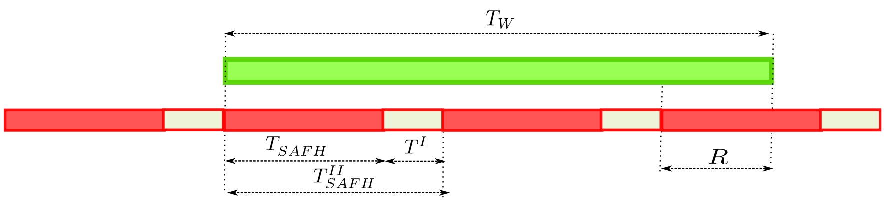

Figure 32: WLAN Interfered by SAFH.

Again let $\tilde{M}$ be the number of times packets overlap in time; there are two cases to consider:

- When $R \leq T_I$

$$\begin{cases} P(\tilde{M} = N) & = \frac{T_I - R}{T^{II}_{SAFH}} \text{ as shown in Figure (33)} \\ P(\tilde{M} = N+1) & = 1 - \frac{T_I - R}{T^{II}_{SAFH}} = \frac{T_{SAFH}+R}{T^{II}_{SAFH}} \quad \text{in the rest of the cases} \end{cases} \tag{30}$$

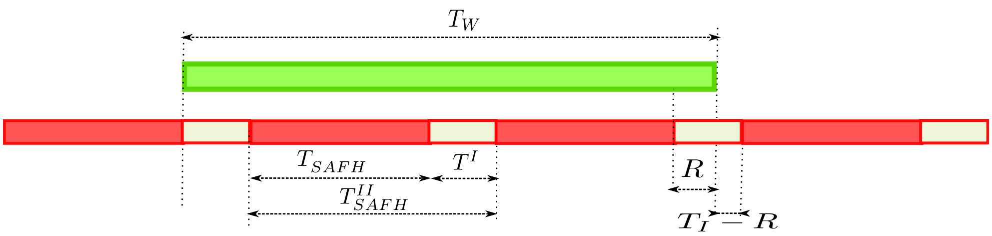

Figure 33: WLAN: $R < T_I$

- When $R > T_I$

$$\begin{cases} P(\tilde{M} = N+2) & = \frac{R - T_I}{T^{II}_{SAFH}} \text{ as shown in Figure 34} \\ P(\tilde{M} = N+1) & = 1 - \frac{R - T_I}{T^{II}_{SAFH}} = \frac{T^{II}_{SAFH} - R + T_I}{T^{II}_{SAFH}} \quad \text{in the rest of the cases} \end{cases} \tag{31}$$

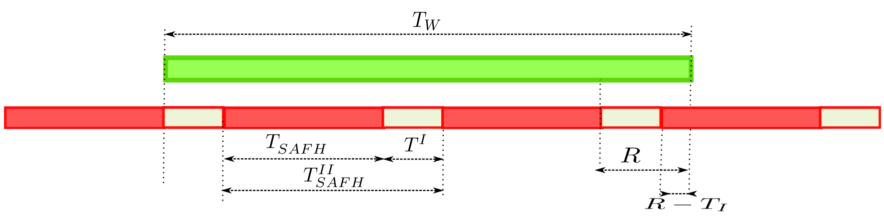

Figure 34: WLAN: $R > T_I$

Let $P_s = (1 - \sum_{i=c}^{i=c+21} P_i)$; using the theorem of total probability ($P(C) = \sum_m \left[P(\tilde{M} = m)(1 - (1-p)^m) \right]$), When $R < T_I$

$$P(C) = \left[(\frac{T_I - T_R}{T^{II}_{SAFH}}) \cdot (1 - P_s^N) \right] + \left[(\frac{T_{SAFH} + R}{T^{II}_{SAFH}}) \cdot (1 - P_s^{N+1}) \right] \quad (32)$$

When $R < T_I$

$$P(C) = \left[(\frac{R - T_I}{T^{II}_{SAFH}}) \cdot (1 - P_s^{N+2}) \right] + \left[(\frac{T^{II}_{SAFH} - R + T_I}{T^{II}_{SAFH}}) \cdot (1 - P_s^{N+1}) \right] \quad (33)$$

Now we include the traffic load and combine Equation (32) and Equation (33) into a more compact form:

$$\begin{cases} P(C) & = P_L \cdot [(\frac{|T_I - R|}{T^{II}_{SAFH}}) \cdot (1 - (P_s)^{N+1-(\frac{|T_I-R|}{(T_I-R)})}) + (\frac{T^{II}_{SAFH} - |T_I - R|}{T^{II}_{SAFH}}) \cdot (1 - (P_s)^{N+1})] \\ & = P_L \cdot [(\frac{|T_I - R|}{T^{II}_{SAFH}}) \cdot (1 - (1 - \sum_{i=c}^{i=c+21} P_i)^{N+1-(\frac{|T_I-R|}{(T_I-R)})}) \\ & +(\frac{T^{II}_{SAFH} - |T_I - R|}{T^{II}_{SAFH}}) \cdot (1 - (1 - \sum_{i=c}^{i=c+21} P_i)^{N+1})] \end{cases} \quad (34)$$

4.2.2 SAFH in the presence of ZigBee

To derive $P(overlap\, in\, frequency)$, we recall that one ZigBee (802.15.4) channel occupies a bandwidth of $2MHZ$, as a result:

$$P(overlap\, in\, frequency) = \sum_{i=c}^{i=c+1} P_i = P_i + P_{i+1} \quad (35)$$

where i and i+1, denote the channels used by ZigBee

802.15.4 employs CSMA/CA therefore, we can use the results obtained for WLAN to derive P(overlap in time) for SAFH and ZigBee.

SAFH interfered by ZigBee From Equation (29), we can deduce the probability of collision:

$$\begin{cases} P(C) & = P_L \cdot (\sum_{i=c}^{i=c+1} P_i) \quad \text{when} \quad T_{SAFH} \geq T_{Boff} \\ P(C) & = P_L \cdot \left(\frac{T_Z + T_{SAFH}}{T_{ZI}} \right) \cdot (\sum_{i=c}^{i=c+1} P_i) \quad \text{when} \quad T_{SAFH} < T_{Boff} \end{cases} \quad (36)$$

ZigBee interfered by SAFH Based on Equation (34), we obtain the probability of collision:

$$\begin{cases} P(C) & = P_L \cdot [(\frac{|T_I - R|}{T^{II}_{SAFH}}) \cdot (1 - (1 - \sum_{i=c}^{i=c+1} P_i)^{N+1-(\frac{|T_I-R|}{(T_I-R)})}) \\ & +(\frac{T^{II}_{SAFH} - |T_I - R|}{T^{II}_{SAFH}}) \cdot (1 - (1 - \sum_{i=c}^{i=c+1} P_i)^{N+1})] \end{cases} \quad (37)$$

4.2.3 SAFH in the presence of Blutooth (BT)

Let $P_i = P(overlap\, in\, frequency)$, of SAFH packet, with the bluetooth packet; i.e. channel i is used by both systems. We will follow the same approach used by [38] to derive $P(overlap\, in\, time)$ for different bluetooth packet types. We will deduce the probability of collision for 100% traffic load, then extend the results to accommodate other duty cycles.

SAFH interfered by Bluetooth A Bluetooth packet occupies either 1, 3 or 5 time slots.

1-Slot BT packet During the transmission of a SAFH packet, one interfering bluetooth packet may overlap a maximum of two $(\lceil \frac{T_{SAFH}}{T_{BT}^{II}} \rceil + 1)$ times; the time offset is uniformly distributed between 0 and T_{BT}^{II} as shown in Figure 35, where all possible over-shifts of SAFH packet are considered. T_{BT}^{II} is the inter-arrival time between BT packets, T_{BT} is the data portion of the BT packet and T_I is the idle time which is same for both systems, (i.e. SAFH and BT).

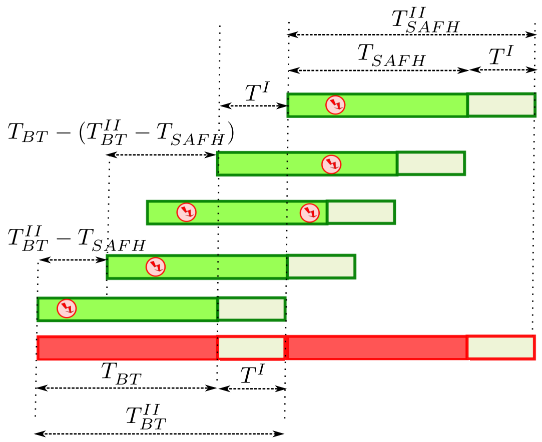

Figure 35: SAFH interfered by BT: 1 slot packet

We can see that SAFH packet can experience double collision with probability $\frac{T_{BT}-(T_{BT}^{II}-T_{SAFH})}{T_{BT}^{II}}$; it can experience single collision with probability $(1 - \frac{T_{BT}-(T_{BT}^{II}-T_{SAFH})}{T_{BT}^{II}}) = \frac{2*T_{BT}^{II}-T_{SAFH}-T_{BT}}{T_{BT}^{II}}$. By using the theorem of total probability $P(C) = \sum_m \left[P(\tilde{M} = m)(1-(1-p)^m) \right] \Longrightarrow$.

$$P(C) = \frac{2*T_{BT}^{II} - T_{SAFH} - T_{BT}}{T_{BT}^{II}} \cdot (1-(1-p_i)^1) + \frac{T_{BT} - (T_{BT}^{II} - T_{SAFH})}{T_{BT}^{II}} \cdot (1-(1-p_i)^2) \tag{38}$$

For arbitrary traffic load $P_L \leq 100\%$, we multiply the result in (38) by P_L, to obtain:

$$\begin{cases} P(C) & = P_L * [\frac{2*T_{BT}^{II}-T_{SAFH}-T_{BT}}{T_{BT}^{II}} \cdot (1-(1-p_i)^1) \\ & + \frac{T_{BT}-(T_{BT}^{II}-T_{SAFH})}{T_{BT}^{II}} \cdot (1-(1-p_i)^2)] \end{cases} \tag{39}$$

3-Slot and 5-Slot BT packet During the transmission of 5-slot BT packets and 3-slot BT packets (shown in Figure 36), we obtain the same results as in (39). However T_{BT}, T_{BT}^{II} take different values dependent on BT packet type.

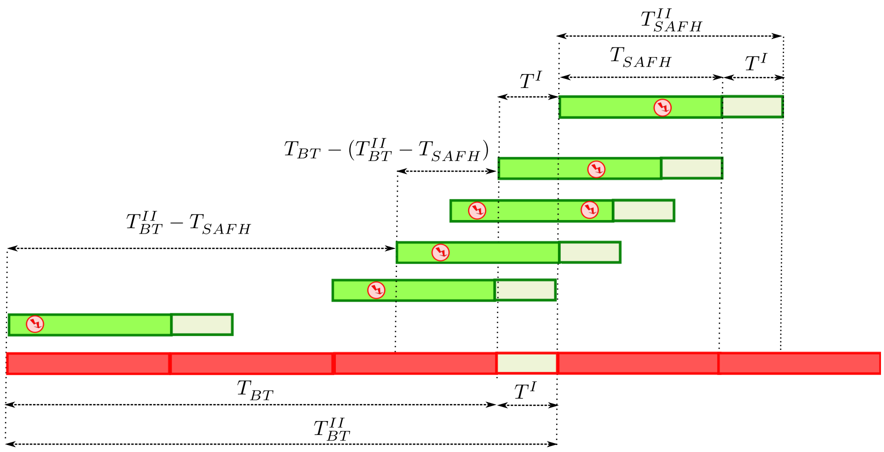

Figure 36: SAFH interfered by BT: 3 slot packet.

Coexistence with N Piconets The probability of successful transmission in the presence of one piconet is:

$$\begin{cases} P(S_1) & = 1 - P(C) \quad \Longrightarrow \\ P(S_1) & = \frac{2*T_{BT}^{II}-T_{SAFH}-T_{BT}}{T_{BT}^{II}} \cdot (1-p_i)^1 + \frac{T_{BT}-(T_{BT}^{II}-T_{SAFH})}{T_{BT}^{II}} \cdot (1-p_i)^2 \end{cases} \tag{40}$$

In the presence of N piconets the probability of collision is

$$P(C) = 1 - P(S_1)^N \tag{41}$$

Bluetooth interfered by SAFH we have three different situations depending on the BT packet type.

1-Slot BT packet In this case we have the same result as in (39).

3-Slot BT packet A 3-Slot BT packet occupies three successive slots in the same frequency channel; SAFH packet may interfere with the BT packet a maximum of 4 times, i.e. $\lceil\frac{T_{BT}}{T_{SAFH}^{II}}\rceil + 1)$ with probability $\frac{T_{SAFH}-(3*T_{SAFH}^{II}-T_{BT}}{T_{SAFH}^{II}})$, and a minimum of 3 times,with probability $1-(\frac{T_{SAFH}-(3*T_{SAFH}^{II}-T_{BT}}{T_{SAFH}^{II}}) = \frac{4*T_{SAFH}^{II}-T_{SAFH}-T_{BT}}{T_{SAFH}^{II}}$ as shown in Figure 37.

Accommodating for different traffic loads and applying the theorem of total probability $P(C) = \sum_m \left[P(\tilde{M}=m)(1-(1-p)^m)\right]$, we obtain:

$$\begin{cases} P(C) & = P_L * [\frac{4*T_{SAFH}^{II}-T_{SAFH}-T_{BT}}{T_{SAFH}^{II}} \cdot (1-(1-p_i)^3) \\ & + \frac{T_{SAFH}-(3*T_{SAFH}^{II}-T_{BT}}{T_{SAFH}^{II}} \cdot (1-(1-p_i)^4)] \end{cases} \tag{42}$$

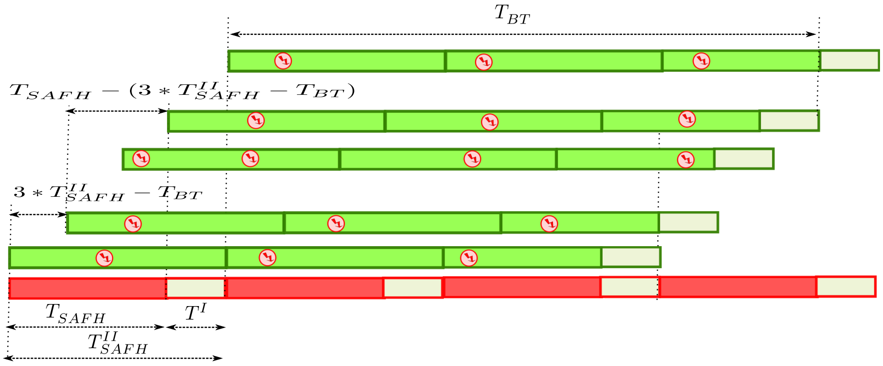

Figure 37: SAFH interfered by BT: 3 slot packet

5-Slot BT packet A 5-Slot BT packet occupies five successive slots in the same frequency channel; SAFH packet may interfere with the BT packet a maximum of 6 times, i.e. $(\lceil \frac{T_{BT}}{T^{II}_{SAFH}} \rceil + 1)$ with probability $\frac{T_{SAFH}-(5*T^{II}_{SAFH}-T_{BT})}{T^{II}_{SAFH}}$, and a minimum of 5 times with probability $\frac{6*T^{II}_{SAFH}-T_{SAFH}-T_{BT}}{T^{II}_{SAFH}}$; therefore the probability of collision is :

$$\begin{cases} P(C) & = P_L * [\frac{6*T^{II}_{SAFH}-T_{SAFH}-T_{BT}}{T^{II}_{SAFH}} \cdot (1-(1-p_i)^5) \\ & +\frac{T_{SAFH}-(5*T^{II}_{SAFH}-T_{BT})}{T^{II}_{SAFH}} \cdot (1-(1-p_i)^6)] \end{cases} \quad (43)$$

4.3 Packet Error Rate & Packet Loss

Packet error Rate $P(E)$ is the percentage of packets containing at least one error, prior to applying error correction; packet loss $P(L)$ on the other hand, is the fraction of packets discarded due to uncorrected errors.

Both metrics i.e. P(E) and P(L), are closely related as will be shown; we start by capturing packet error, then use the result to discuss the packet loss.

4.3.1 Packet Error

THe Packet error Rate $P(E)$, is related to the collision probability; this is explained as follows: When a collision occurs between the desired and the interfering packet, it is detected at the wireless receiver as signal to interference ratio (SIR), which is then mapped to bit error rate (BER) according to the modulation used [39].

Let $P(EF)$ be the probability of error free packet. Using the theorem of total probability, it can be expressed as follows :

$$P(EF) = P(EF|C) * P(C) + P(EF|NC) * [1 - P(C)] \quad (44)$$

where $P(EF|C)$, is the probability of "error free packet", conditioned on the occurrence of a collision; $P(EF|NC)$ is the conditional probability of "error free packet" given no collision.

Let BER_c and BER_{nc} denote the bit error probability when there is a collision and no collision respectively; $\tilde{L}$ is the number of bits involved in the impact. This is clearly a random variable that has a uniform distribution i.e. $U(0, M)$, where M is $min(N_D, N_I)$; N_D and N_I are the number of bits in desired and interfered packet respectively.

If we condition on $\tilde{L} = l$, we will get the following expressions:

$$\begin{cases} P(EF|NC) & = (1 - BER_{nc})^{N_D} \\ P(EF|C) & = (1 - BER_c)^l (1 - BER_{nc})^{N_D - l} \end{cases} \tag{45}$$

Now we can remove the condition on $\tilde{L} = l$, by averaging over all possible values:

$$\begin{cases} P(EF|NC) & = (1 - BER_{nc})^{N_D} \\ P(EF|C) & = \frac{1}{N_D} \cdot \sum_{l=1}^{N_D} \left((1 - BER_c)^l (1 - BER_{nc})^{N_D - l}\right) \end{cases} \tag{46}$$

By substituting (46) into (44), the packet error $P(E) = 1 - P(EF)$ is obtained; the values for BER_c and BER_{nc} are calculated using formula (47), which provides the BER [39] for GFSK modulation at the bluetooth receiver.

$$BER = Q_1(a, b) - \frac{1}{2} \cdot \exp\left(\frac{a^2 + b^2}{2}\right) \cdot I_0(a, b) \tag{47}$$

where $Q_1(.)$ is the first order Q function and I_0 is the 0-order modified Bessel function.

$$\begin{cases} a & = \sqrt{\frac{\Gamma}{2} \cdot (1 - \sqrt{1 - (\frac{\sin(2*\pi*h)}{2*\pi*h})^2})} \\ b & = \sqrt{\frac{\Gamma}{2} \cdot (1 + \sqrt{1 - (\frac{\sin(2*\pi*h)}{2*\pi*h})^2})} \end{cases}$$

where $\Gamma = \frac{E_b}{N_0}$, E_b is the signal power for BT and N_0 is the noise spectral density. Note that $\frac{E_b}{N_0}$ is replaced by SNR for BER_{nc} and by SINR in the case BER_c. The modulation index $h = 0.32$.

4.3.2 Packet Loss

Packets consist of three portion: the access code AC, the header(HE) and the payload (P); the AC and HE use error correction and if the operation fails, the packet is discarded;

$$P(L) = 1 - (1 - P_{AC}(L))(1 - P_{HE}(L))(1 - P_P(L)) \tag{48}$$

In [40], the authors observed that the probability of loss in the header $P_{HE}(L)$ and in the access code $P_{AC}(L)$ is negligible due to sophisticated error correction, therefore the probability of packet loss is approximately the probability of payload loss $P_P(L)$, i.e. $P(L) \approx 1 - (1 - P_P(L)) = P_P(L)$

In HV3 packets the payload is not protected; in this case $P(E) = P(L)$.

The packets where the payload is protected by FEC e.g. single slot HV1 and HV2 packets, the analysis requires the knowledge of the code word error probability $P(CWE)$ for each FEC code adapted; $P(CWE)$ is a function of the number of correctable errors and the code word length [40].

In order to validate the accuracy of the proposed analysis, both theoretical and simulation results are presented in Appendix A; a numerical example is also provided.

5 Simulation

Simulation of wireless system is a crucial step in performance evaluation and benchmarking [41]; moreover it helps the discovery of design flaws early in the development process.

5.1 Tools

Software packages for building simulation models fall into two categories [41]:

- Tools that use general purpose programming languages; they offer flexibility and extensibility.

- Graphic model builders; they provide a natural representation of communication systems.

In order to take full advantage of both approaches, we chose SIMULINK® [42], an extension of MATLAB; as a result we are able to use features offered in both environments during the analysis process.

Another compelling reason for choosing SIMULINK is the ability to describe blocks using computer languages such as C and C++. This is a powerful mechanism that allows us to implement our own algorithm. The resulting code, called system-functions or S-functions, can be added to the model and then customized to have user interface by means of masking, as shown in Figure 38.

In addition to the features mentioned above, SIMULINK includes an extensive block library of toolboxes for communication and signal processing analyses. The communications blockset in particular, contains Bluetooth Demo [43] that contains the following elements: master transmitter, IEEE 802.11b interferer and slave receiver.

The transmitter subsystem performs speech coding, header error control, forward error correction, Gaussian frequency shift keying (GFSK) modulation, and frequency hopping. The slave receiver recovers speech from the transmitted signal, performing all the complementary operations that the transmitter does, but in reverse order [43].

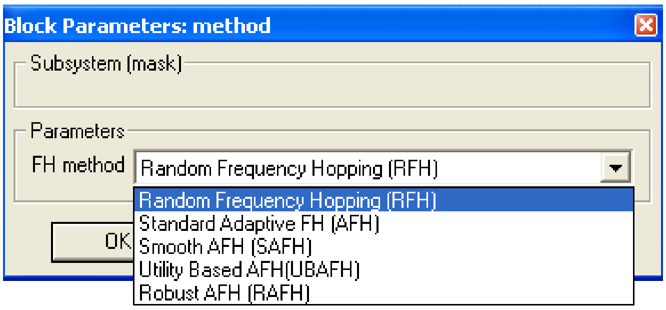

Figure 38: Masking of S function

In order not to reinvent the wheel and start from scratch, we built our model based on the bluetooth demo; fundamental modifications were needed thou. First of all the generator of the hop frequencies was removed, because it can only generate random frequency hopping; we implemented a block capable of generating four more schemes i.e. our algorithm SAFH, the standard AFH [8], RAFH [5] and UBAFH [6]. The block was realised by means of an S-function, written in C++, and masked as shown in Figure 38. The second major modification to the Bluetooth model, was the design of three IEEE 802.15.4, and two more IEEE 802.11b interferers.

5.2 System Model

The model consists of six blocks as shown in Figure 39; opening up these subsystems reveals further levels of details.

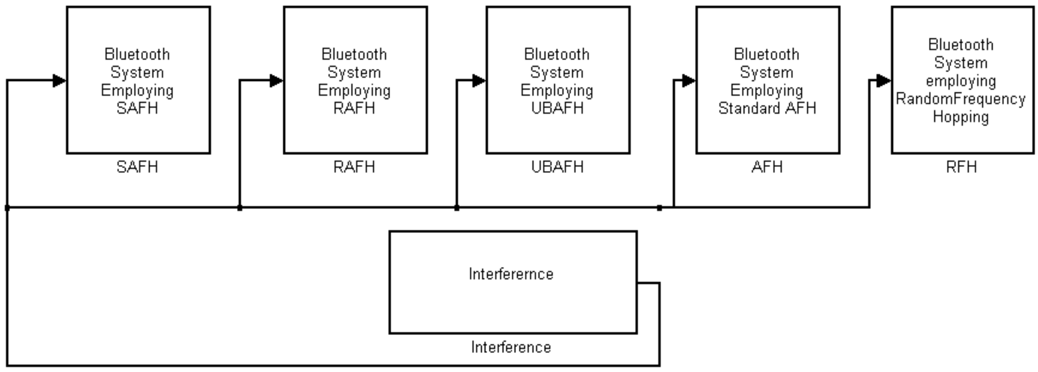

Figure 39: Top level View of the Model

The block named *interference*, consists of three WLANs, three 802.15.4 based Personal Area Networks, and twelve Bluetooth piconets as shown in Figure 40.

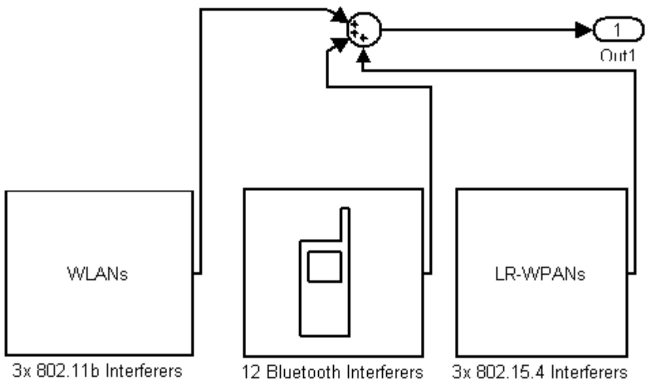

Figure 40: Interference Block

The WLAN (and ZigBee) transmitters were configured to use different parameters such as mean packet rate, packet length, power, and frequency location in the ISM band. Dynamic interference is added by including twelve Bluettoth transmitters, that use the random frequency hopping algorithm,and configured with different seeds.

The other blocks in Figure 39, SAFH, RAFH, UBAFH, AFH and RFH, have the same structure as shown Figure 41; Additive white Gaussian noise (AWGN) is used to model the noise at the slave receiver. Each of these blocks uses the corresponding scheme that generates the hop-set, e.g SAFH block uses SAFH algorithm etc.

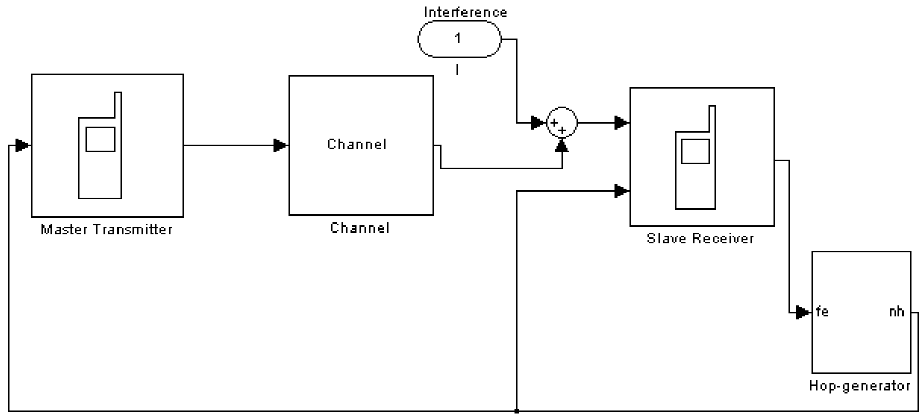

Figure 41: Structure of the Bluetooth Piconets Used in the Model

The fading is flat because the transmitted bandwidth of Bluetooth on any one hop (1 MHz), is smaller than the channel coherence bandwidth (4 MHz for indoor radio propagation); it is calculated using the rms delay spread, which is around 25 nsec in indoor environment [44].

The propagation model consists of two parts [1]:

- line-of-sight propagation (free-space) for the first 8 meters and
- a propagation exponent of 3.3 for distances over 8 meters.

For efficiency, all signals and system elements are represented by their baseband equivalent representation. Analysis of the system using complex low pass equivalent is justified, since no information is lost when using complex baseband signal instead of passband signals. In addition, simulating the baseband equivalent of a passband signal substantially reduces the amount of storage required, as well as the computational time. For example in our case, we are using the 2.4 GHz ISM band which ranges from 2400 MHZ to 2500 MHZ; therefore, according to the Nyquist sampling theorem, sampling the passband signal requires a sampling rate of at least $2 * fmax = 2 * 2500$ MHZ = 5 GHZ.

5.3 Coexistence Environment

The network topology consists of one Bluetooth piconet employing the SAFH algorithm. Operating nearby, there are three 802.15.4 and three 802.11b stations acting as static sources of interference, and up to 12 Bluetooth interferers.

The WLANs use channel 1, channel 6 and channel 11, centred at 2412 MHz, 2437 MHz and 2462 MHz, respectively; these sub-bands do not overlap as discussed in Section 2.1. For the IEEE 802.15.4 nodes, we considered the channels centred at 2425 MHz, 2450 MHz and 2475 MHz, since they do not overlap with the bands used by 802.11b networks.

The Bluetooth piconet under test i.e employing SAFH, uses single slot voice packets (HV1), while the interfering Bluetooth devices, use different voice packets i.e. HV1, HV2 and HV3. This is easily configured for each transmitter, as shown in Figure 42. In HV1 the radio hops into a new frequency each second slot, while HV2 and HV3 hop every fourth and sixth slot respectively, resulting in different traffic load.

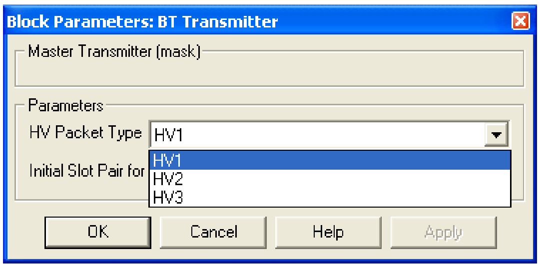

Figure 42: Configuration of Bluetooth Interferers

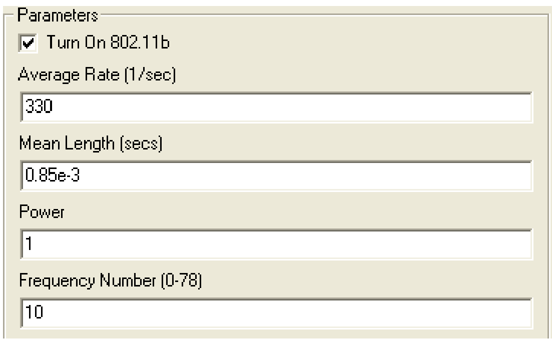

Figure 43: Parameter Settings for the IEEE 802.11b

The traffic load for the collocated IEEE 802.11b and ZigBee nodes is shown in Table 3; it is set by specifying for each node, the mean packet rate and mean packet length in the mask dialogue, shown in Figure 43.

Table 3: Traffic Load of the Interfering Devices

802.11b	**802.15.4**	**802.15.1 (HV1)**	**802.15.1 (HV2)**	**802.15.1 (HV3)**
Ch 1: 28.05 % Ch 6: 6.3 % Ch 11: 30 %	Ch 15: 0.8 % Ch 20: 0.3 % Ch 25: 0.8 %	50 %	25 %	16.67 %

5.4 Scenarios

The default simulation run lasts $30T$, where T is the time needed to go through all the frequencies in the hop-set. The hop-set length of Bluetooth devices consists of 1000 hops, and the interferes use different hopping pattern.

We considered different scenarios:

- One scenario to evaluate the performance of SAFH, under both static and dynamic sources of interference; to highlight this issue, all WLAN, ZigBee and Bluetooth interferers are switched on.
- One scenario to study the performance of SAFH, under static sources of interference; only the 802.11b and 802.15.4 devices are switched on, whereas all Bluetooth interferers are switched off.
- One scenario to determine the responsiveness of SAFH to changes in dynamic environment; the number of Bluetooth interferes, was gradually changed from 0 to 12.
- One scenario to investigate the responsiveness of SAFH to sudden change in interference; one 802.11b device was switched off at time 10T, and then back on at time 20T.

6 Results and Discussions

In this section we present and discuss the results obtained. We quantified the performance of the different communication schemes by measuring the frame error rate. There is a disagreement in the literature on the tolerable FER when analysing audio, because it is more subjective to the listener and is most affected by how many bits are randomized within a burst error event [17]. Some claim that the maximum limit for the loss is 2%, whereas others claim that FER can be as high as to 20% [17]. In this work we required FER to be between 5% to 10% as suggested in [33].

Figure 44, shows the performance of SAFH, RAFH, UBAFH and AFH under the default topology. We can clearly see how SAFH outperforms the other algorithms with respect to frame error rate (FER). SAFH was able to achieve an average FER of 10%, in contrast to AFH (15%), RAFH (18%)and UBAFH (19%). The Figure also shows that the standard AFH exhibit fluctuations; the reason behind this effect, is that it resets bad channels and includes them in the hop-set, every other interval.

RAFH also fluctuates; the stringent requirement on the FER (10%) instructs RAFH to updates the probabilities of the channels, whenever it is unable to meet the threshold (every interval), resulting in the fluctuations. Moreover RAFH does not take advantage of the history of the channels, since it uses only the FER of the most recent interval. We managed to efficiently overcome this shortcoming in our algorithm, by using the Exponential Smoothing filter, as a result SAFH exhibit stable and smooth operation.

The second reason that affects RAFH's performance, is the logic used to assign probabilities for the channels; they attempt to solve $\sum p_i(t+1) * FER_i(t) < \xi$, instead of $\sum p_i(t+1) * FER_i(t+1) < \xi$.

We tackled this issue in a simple and efficient way, by first predicting the FER of the upcoming interval; Based on the forecast FER, we assigned probabilities to the channel.

The result shown in Figure 44, clearly indicate the gain in performance. Moreover we did not have to solve a convex optimization problem, thus reducing the computational complexity.

Clearly, the worst performance is the one termed UBAFH in the Figure 44; this is a special case of our algorithm, when $c = s = 1$ and $\beta = 1 - th$ as mentioned in Section 3; in this case, the probability distribution is a function of the FER only, therefore it does not attempt to meet any constraints.

In Figure 45, we illustrate the performance of SAFH, for different values of the parameter α. The first observation is that the algorithm converges faster for large values of α; in fact the number of time periods needed for convergence is proportional $\frac{1}{\alpha}$, which is a property of the exponential smoothing filter.

After convergence is achieved, we notice that smaller values of α, result in lower average FER; this is due to the fact that larger values α give less weight to the channel history.

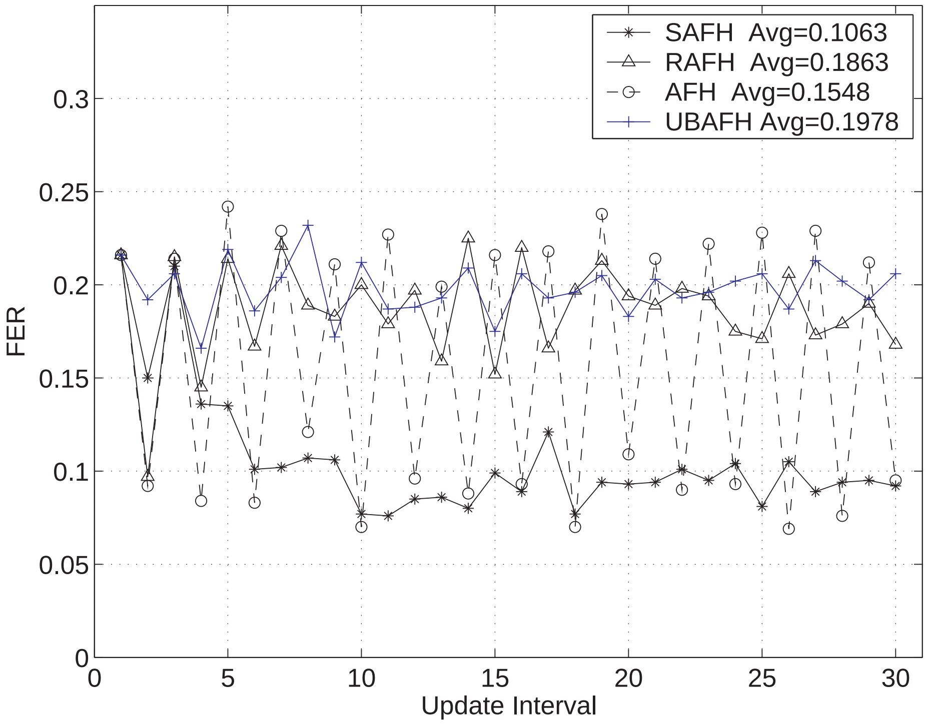

Figure 44: Performance of SAFH versus RAFH, UBAFH and AFH

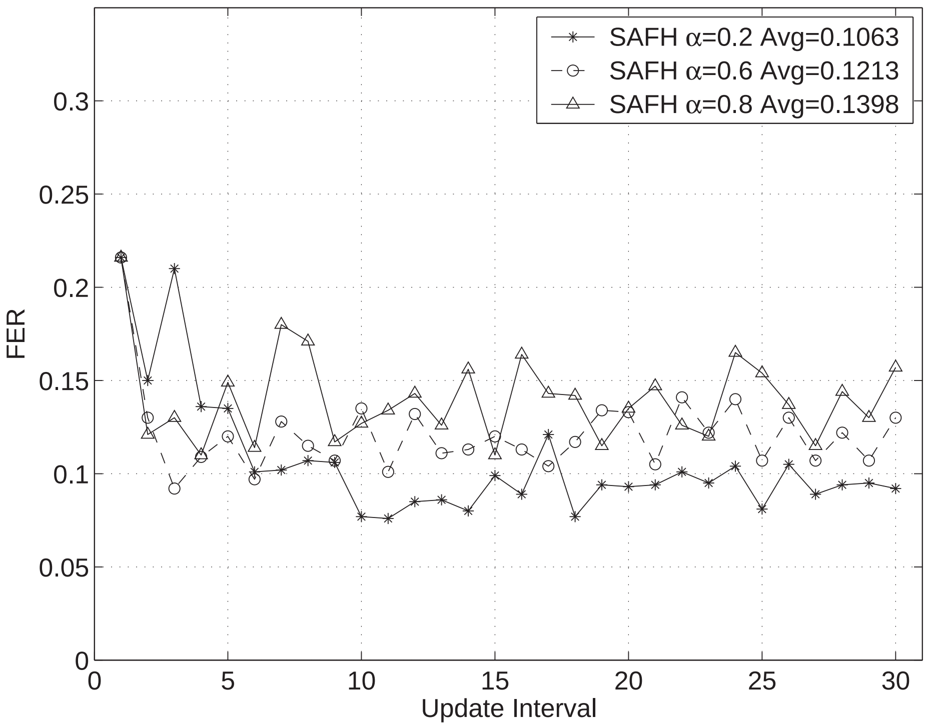

Figure 45: Performance of SAFH for different values of α

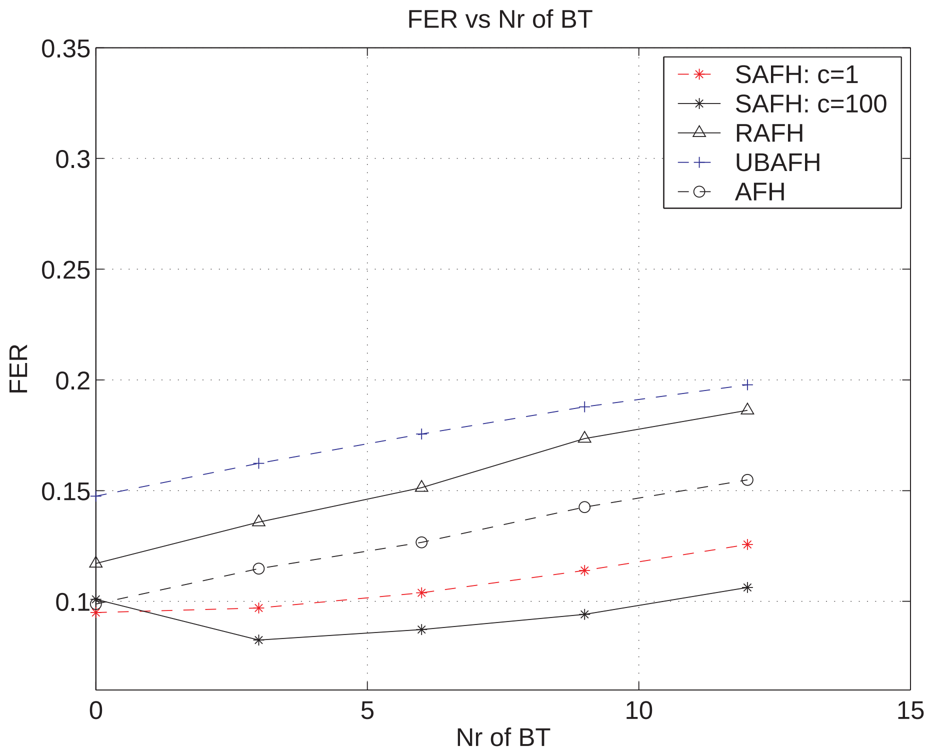

Figure 46: FER versus number of Bluetooth interferers

Figure 46 shows the average FER with respect to the number of Bluetooth interferers. In these scenarios, all WLANs and LR-WPANs are switched on, while the number of Bluetooth interferes is increased gradually. As expected, the FER increases with the number of Bluetooth for all the algorithms. Figure 46 also illustrates, that SAFH outperforms RAFH, UBAFH as well as AFH under different dynamic interference scenarios.

In Figure 46, we observe also, that when SAFH has its rewarding factor $c = 100$, it performs better than with $c = 1$; this matches the intuition, since for larger the value of c, good channels are used more often as indicated in Equation (14).

A strange behaviour is observed, in the absence of dynamic source of interference, i.e. the only coexistence nodes are the IEEE 802.11b and IEEE 802.15.4. In this case, SAFH ($c = 1$), has lower average rate than SAFH ($c = 100$); the average FER of all the algorithms, are presented in Table 4.

No explanation is found yet, for this seemingly counter intuitive result. We expect future investigation will shed the light on this issue, as well as to what combination of the parameters (c, s), leads to the best performance, and under which scenarios.

In Figure 47, we illustrate the performance of SAFH under the default topology, by fixing values of β. We notice better performance for smaller values β than for larger ones. This however comes at the price of less frequency diversity, since smaller β results in less channels being used as illustrated previously in Section 3.

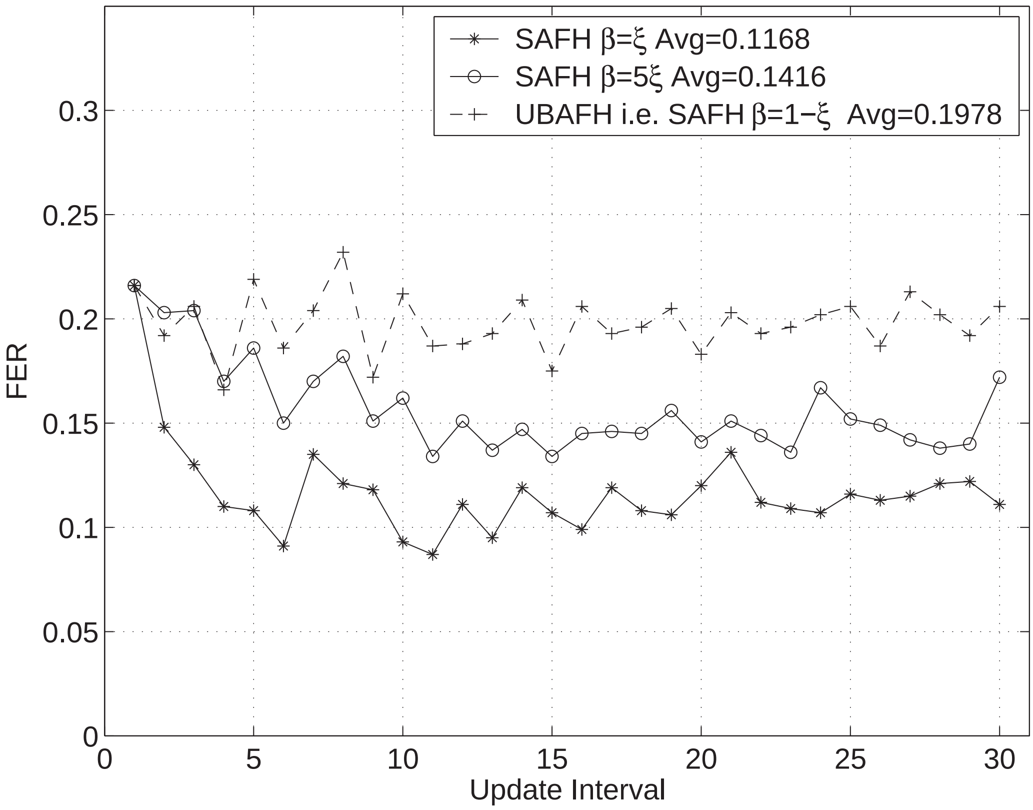

Figure 47: Performance of SAFH for fixed values of β

Table 4: Average FER under Static sources of Interferences

SAFH c=100 $\alpha = 0.2$	**SAFH** c=1 $\alpha = 0.2$	**SAFH** c=100 $\alpha = 0.6$	**AFH**	**RAFH**	**UBAFH**
0.1009	0.0949	0.0732	0.0988	0.1171	0.1475

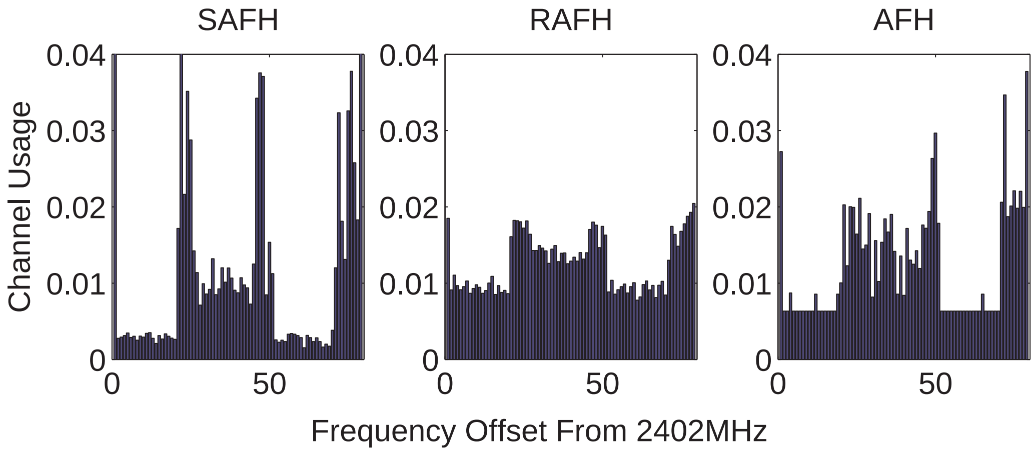

Figure 48: Channel Usage of SAFH, RAF and AFH

Figure 48 illustrate the channel usage of SAFH, RAFH and AFH under the default topology. The x -axis in the figure ranges from 0 to 80, where 0 corresponds to channel 2402MHz in the ISM band.

The 802.11b networks centered at channel 1,6 and 11 are identified successfully by *all* the algorithms; the main difference lies in how often one algorithm uses bad channels than the others.

Looking at the facts, we can say that SAFH uses bad channels less than the other schemes. This was in fact a key requirement that we set while designing our algorithm, and Figure 48 clearly indicates, that we achieved this goal.

The last evaluation criterion is the responsiveness of SAFH to sudden change of the interference. To highlight this issue, we switched one the 802.11b devices off at time $10T$, and then back on at time $20T$. Recall that 802.11b is the main source of interference.

Figure 49 shows how turning 802.11b off, lowers the FER slightly. When we switch the device back on at time $20T$, an FER peak ($T = 20$, $FER = 14\%$) occurs, that drops down within less than $3T$ as the algorithm converges. The quick adaptation to changing conditions, is an important asset of AFH algorithms. Some applications such as voice communication would benefit from this effect, while for others, such as file transfer, the achieved throughput is a more important factor.

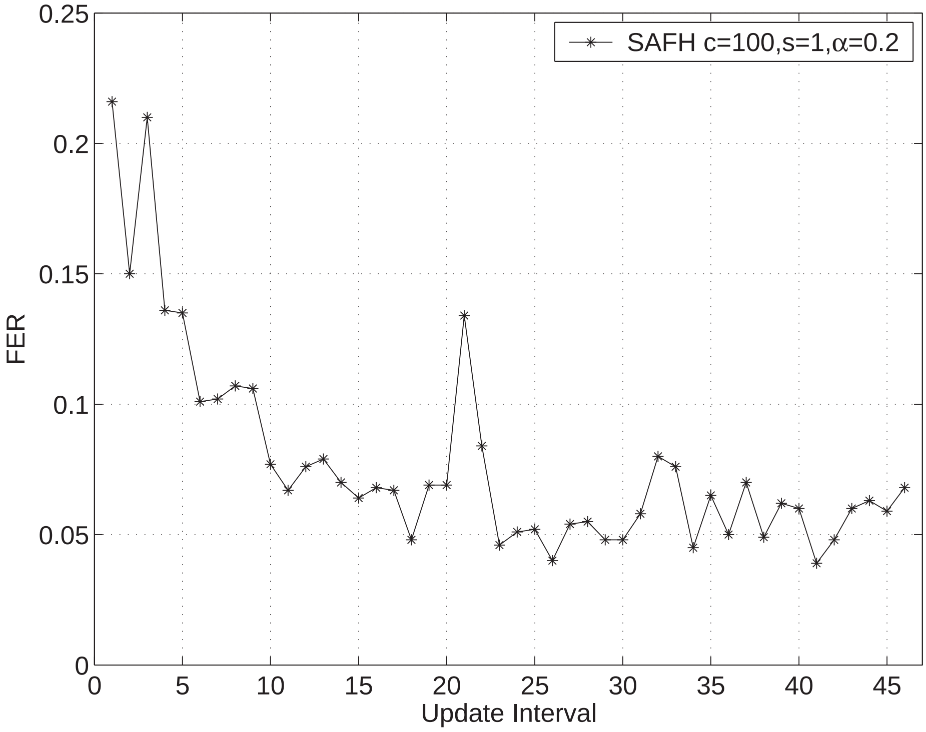

Figure 49: Response of SAFH to Change in the Environment; Switch off 1 WLAN at $T = 10$ and again on at $T = 20$

7 Conclusion and Future Work

In this report we motivated the need for a new adaptive frequency hopping algorithm, and proposed a novel scheme based on probabilistic channel usage.

The developed algorithm, named smooth adaptive frequency hopping (SAFH), consists of four steps. First, it performs channel classification, then uses exponential smoothing filter to make channel prediction; based on the forecast status of the channels, SAFH determines the probability mass function, which is later mapped to a hop-set for frequency hopping.

In order to quantify the performance of SAFH, both analytical performance as well as simulation studies were carried out. Different scenarios were investigated, with emphasis on dynamic channel environment.

Our achievements were compared to the results of other adaptive hopping algorithms; it shows that SAFH outperforms the other schemes with respect to frame error rate, under static and dynamic sources of interferences. In addition it exhibits fast adjustment to changes in the environment and very stable operation i.e. less fluctuations.

The fact that SAFH outperforms robust adaptive frequency hopping (RAFH), has another implication in terms of energy consumption; this is because the determination of probability distribution in SAFH, is much simpler than the convex optimization used in RAFH, which translates to lower energy consumption.

In this thesis, we evaluated the interference at the SAFH receiver, while ignoring the mutual interactions with the interferers (WLAN, LR-PAN, Bluetooth). Ignoring this aspect provides pessimistic results, since it does not consider the changes to the traffic patterns.

For the future we suggest an SDR implementation of SAFH and compare the results to the ones obtained in the simulation; in addition further investigation of the effects of mutual interference as well as SAFH parameter space, is worth the effort.

References

[1] "IEEE 802.15.2: Coexistence of wireless personal area networks with other wireless devices operating in unlicensed frequency bands," [http://standards.ieee.org/getieee802/download/802.15.2-2003.pdf], online; accessed 22-May-2011.

[2] Z. Jiang, V. C. Leung, and V. W. Wong, "Reducing collisions between bluetooth piconets by orthogonal hop set partitioning," in *Proc. IEEE Radio and Wireless Conference RAWCON*, 2003.

[3] P. Popovski, H. Yomo, R. Prasad, and S. Member, "Dynamic adaptive frequency hopping for mutually interfering wireless personal area networks," *ACM Mobihoc*, vol. 4, pp. 991–1003, 2004.

[4] Q. Pang and V. C. Leung, "Improved channel classification and scheduling for non-collaborative bluetooth, WLAN coexistence," *IEEE 63rd Vehicular Technology Conference*, 2006.

[5] K.-J. Park, T. R. Park, C. D. Schmitz, and L. Sha, "Entropy-maximization based adaptive frequency hopping for wireless medical telemetry systems," in *Proc. 1st ACM International Workshop on Medical-grade Wireless Networks WiMD*, 2009.

[6] S. Luca, "Toward reliable wireless sensor networks : Energy-aware distributed interference management for unlicensed bands," Ph.D. dissertation, KTH, Communication Systems, CoS, 2010.

[7] *Communication Systems*. Wiley Publishing, 2009.

[8] *Part 15.1: Wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs)*, IEEE Standard 802.15.1-2005 Std., 2005.

[9] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher-Speed Physical Layer Extension in the 2.4GHz Band,*, IEEE Std. 802.11b Std., 1999.

[10] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Standard 802.15.4-2006, revision of IEEE Standard 802.15.4-2003. Std., 2006.

[11] N. Golmie, "Interference in the 2.4 ghz ism band: Challenges and solutions," [http://www.antd.nist.gov/pubs/golmie.pdf], online; accessed 10-February-2011.

[12] "The bluetooth special interest group SIG," [http://en.wikipedia.org/wiki/Bluetooth-Special-Interest-Group], online; accessed 18-November-2011.

[13] "GNU radio," [http://gnuradio.org/redmine/projects/gnuradio/wiki], online; accessed 18-November-2011.

[14] "USRP2 board," [http://www.ettus.com], online; accessed 18-November-2011.

[15] *Bluetooth Specifications*, [http://grouper.ieee.org/groups/802/15/Bluetooth], Std., online; accessed 10-September-2011.

[16] "Bluetooth white paper 1.1, AU-system," [http://www.cse.iitd.ernet.in/ csd97403/btwp.pdf], online; accessed 20-October-2011.

[17] *Wireless Network Coexistence.* McGraw-hill, 2004.

[18] R. Jäntti, "Lecture notes: S-72.3235 network access," [http://noppa.aalto.fi/noppa/kurssi/s-72.3235/etusivu].

[19] *ZigBee Specification Version 1.0, June 2005*, ZigBee Alliance Std.

[20] A. Koubaa, M. Alves, and E. Tovar, "Technical report: IEEE 802.15.4: a federating communication protocol for time-sensitive wireless sensor networks," 2006.

[21] "IEEE DySPAN standards committee (DySPAN-SC)," [http://www.dyspan-sc.org], online; accessed 24-November-2011.

[22] "IEEE 1900.2 working group on recommended practice for the analysis of in-band and adjacent band interference and coexistence between radio systems," [http://grouper.ieee.org/groups/dyspan/2/index.htm], online; accessed 24-November-2011.

[23] Y. C. Maa and A. Batra, "Alternating wireless medium access (AWMA)," *IEEE 802.15-02/138r0*, 2001.

[24] J. Lansford and R. Nevo, "MEHTA: A method for coexistence between co-located 802.11b and bluetooth systems," *IEEE 802.15-00/036r0*, 2000.

[25] R. V. Dyck and A. Soltanian, "Deterministic frequency nulling to mitigate bluetooth interference," *IEEE 802.15 TG2 Coexistence mechanisms presentations*, 2001.

[26] H. Yomo, P. Popovski, H. C. Nguyen, and R. Prasad, "Adaptive frequency rolling for coexistence in the unlicensed band," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 598 – 608, 2007.

[27] K.-J. Park, T. R. Park, C. D. Schmitz, and L. Sha, "Design of robust adaptive frequency hopping for wireless medical telemetry systems," *IET Commun.*, vol. 4, no. 2, pp. 178–1191, 2010.

[28] L. Stabellini, L. Shi, A. A. Rifai, J. Espino, and V. Magoula, "A new probabilistic approach for adaptive frequency hopping," in *20th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications PIMRC*. IEEE, 2009, pp. 2147–2151.

[29] "WPANs specifications," [http://standards.ieee.org/about/get/802/802.15.html], online; accessed 18-November-2011.

[30] "Principle of maximum entropy," [http://en.wikipedia.org/wiki/Principle-of-maximum-entropy], online; accessed 1-May-2011.

[31] *Convex Optimization*. Cambridge University Press, 2003.

[32] *Smoothing, Forecasting and Prediction of Discrete Time Series*. Courier Dover Publications, 2004.

[33] T. Bourk, "Techniques mitigate interference between 802.11 and bluetooth," [http://www.design-reuse.com/articles/4288/techniques-mitigate-interference-between-802-11-and-bluetooth.html], online; accessed 12-Mars-2011.

[34] Mathworks, "Common generation methods," [http://www.mathworks.com/help/toolbox/stats/br5k9hi-1.html], online; accessed 22-Mars-2011.

[35] *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.

[36] "Orders of approximation," [http://en.wikipedia.org/wiki/Ordersofapproximation], online; accessed 1-May-2011.

[37] S. Shellhammer, "Probability of bluetooth/WLAN packet collision," *IEEE P802.19 Coexistance tag, 2006*, 2006.

[38] A. Nallanathan, W. Feng, and H. K. Garg, "Coexistence of wireless LANs and bluetooth networks in mutual interference environment an integrated analysis," *Journal Computer Communications*, 2006.

[39] S. M., S. S., and G. D., "Coexistence of IEEE 802.11b and bluetooth: an integrated performance analysis," *Springer Science*, 2008.

[40] A. Conti, D. Davide, G. Pasolini, and O. Andrisano, "Bluetooth and IEEE 802.11b coexistence: analytical performance evaluation in fading channels," *IEEE journal on selected areas in communications*, 2003.

[41] *Principles of Communinication Systems Simulation with Wireless applications*. prentice Hall, 2004.

[42] Mathworks, "Simulink - simulation and model-based design," [http://www.mathworks.com/products/simulink/], online; accessed 5-June-2011.

[43] "Communications blockset demos," [http://www.physnet.uni-hamburg.de/physnet/matlab/help/toolbox/commblks/usersguide/moredem3.html], online; accessed 16-November-2011.

[44] H. Hashemi and D. Tholl, "Analysis of the RMS delay spread of indoor radio propagation channels," [http://www.trlabs.ca/library/pubs/cal-C0015/C0015.pdf], online; accessed 10-October-2011.

Appendix A

Illustrative Example

We consider the following topology: two SAFH nodes collocated with one WLAN transmitter , using channel 6 (i.e. 2427-2448 MHz). SAFH packets are assumed to be HV3 (no FEC and length: 366 μs), therefore $P(E) = P(L)$; in addition we assume that the mean packet length of WLAN is 1.5 ms, the mean back-off period is 166 μs and the mean rate is 600, thus the traffic load of the WLAN is 90%.

In this scenario, $T_{SAFH} \geq T_{Boff}$, therefore to calculate the probability of collision, we use the following equation:

$$P(C) = P_L \cdot (\sum_{i=c}^{i=c+21} P_i) = 0.9 \cdot (\sum_{i=c}^{i=c+21} P_i) \tag{A1}$$

The probability mass function calculated by SAFH (third step of the algorithm), is used to find $\sum_{i=c}^{i=c+21} P_i$.

To calculate $P(EF|C)$ and $P(EF|NC)$, we need the values of BER_c and BER_{nc}, which can be calculated using Equation (47), or using the approximation, based on standard non-coherent FSK detection [10]

$$BER = 0.5 \cdot \exp\left(-0.5 \cdot SINR\right) \tag{A2}$$

With SNR equal to 14dB,

$$BER_{nc} = 0.5 \cdot \exp\left(-0.5 \cdot 10^{(14/10)}\right) = 1.7558 \cdot 10^{-6}$$

To find BER_c, we need to find SINR $= SNR * SIR/(SNR + SIR)$, and plug its value in Equation (A2); where SIR is the signal to interference ratio;

In this model SAFH transmit power is 1 mw (0dBm); the transmitted power of the 802.11b station is $P_{WLAN} = 1W$ (30 dBm) and it is located 1 m from the SAFH receiver, resulting in path loss $L = 40dB$. The interference power at the SAFH receiver is $P_{WLAN}/(L*22)$, or in dB $P_{WLAN} - L - \log(22) = -11.34$.the digit 22 is the ratio of the 802.11b spread bandwidth (22 MHz) to SAFH information bandwidth (1 MHz).

$$SIR = 0 - -11.34 = 11.34$$

$$SINR = SNR * SIR/(SNR + SIR) = 6.26$$

$$BER_c = 0.5 \cdot \exp\left(-0.5 \cdot 10^{(6.26/10)}\right) = 0.06$$

Now we can calculate $P(EF|NC)$ and $P(EF|C)$ as shown in Equation (A3); $N_d = 366$.

$$\begin{cases} P(EF|NC) & = (1 - BER_{nc})^{N_D} = 0.9994 \\ P(EF|C) & = \frac{1}{N_D} \cdot \sum_{l=1}^{N_D} \left((1 - BER_c)^l (1 - BER_{nc})^{N_D - l}\right) \\ & = 3.3950 \cdot 10^{-13} \end{cases} \tag{A3}$$

Let's suppose that the $\sum_{i=c}^{i=c+21} P_i = 0.17$, then the probability of collision

$$P(C) = P_L \cdot (\sum_{i=c}^{i=c+21} P_i) = 0.9 \cdot 0.17 = 0.1530$$

Consequently The probability of error:

$$P(E) = 1 - (P(EF|C) * P(C) + P(EF|NC) * [1 - P(C)]) = (1 - 0.8465) = 0.1535$$

Validation of the Theoretical model

This part complements Section 4; the simulation results are compared with the theoretical model for the same topology used in the numerical example. The results for SAFH with parameter α equal to $0.5, 0.9$ and 0.1 are depicted in Figure A1, Figure A2 and Figure A3 respectively.

We consider another scenario where ($T_{SAFH} < T_{Boff}$); the mean packet length of WLAN is 1.5 ms, the mean back-off period is 2.5 ms, and the mean rate is 250, resulting in 37.5% traffic load. The results for SAFH with parameter α equal to $0.5, 0.9$ and 0.1 are depicted in Figure A4, Figure A5 and Figure A6.

Even thou the results obtained from the simulation and the simple mathematical model follow the same pattern, they do not coincide most of the time. The mathematical model needs to be investigated further.

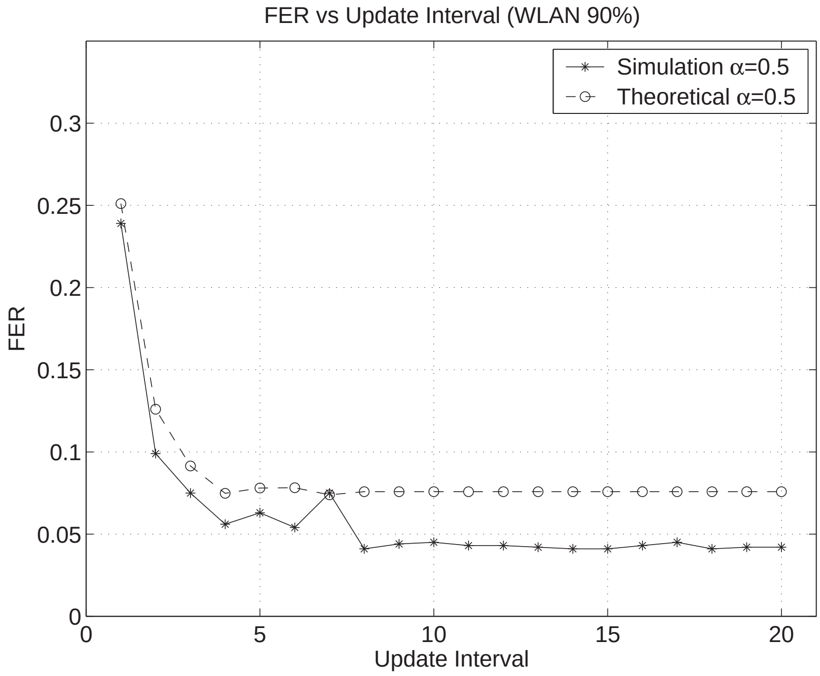

Figure A1: Theoretical vs Simulation: SAFH, $\alpha = 0.5$ (Traffic load of WLAN is 90%)

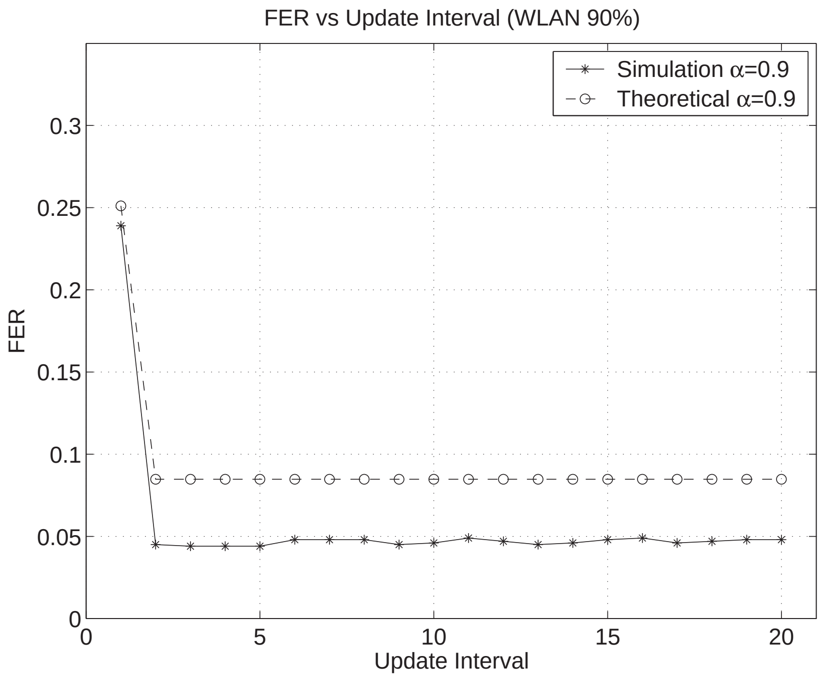

Figure A2: Theoretical vs Simulation: SAFH, $\alpha = 0.9$ (Traffic load of WLAN is 90%)

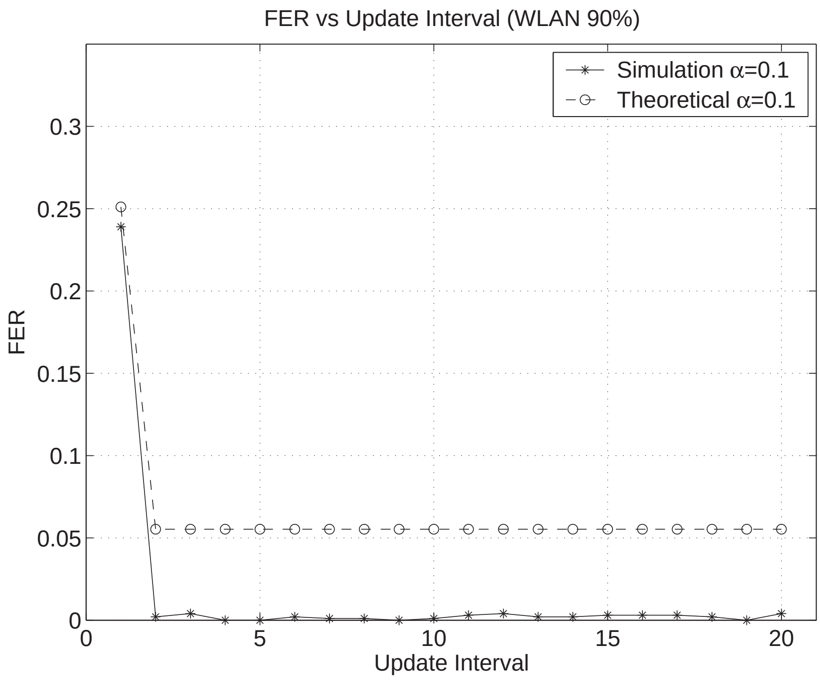

Figure A3: Theoretical vs Simulation: SAFH, $\alpha = 0.1$ (Traffic load of WLAN is 90%)

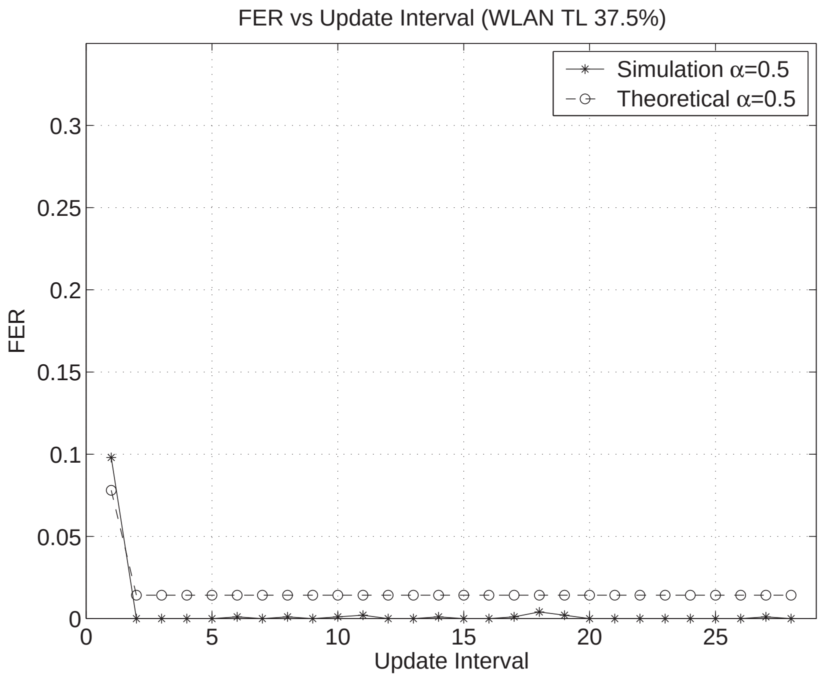

Figure A4: Theoretical vs Simulation: SAFH, $\alpha = 0.5$ (Traffic load of WLAN is 37.5%)

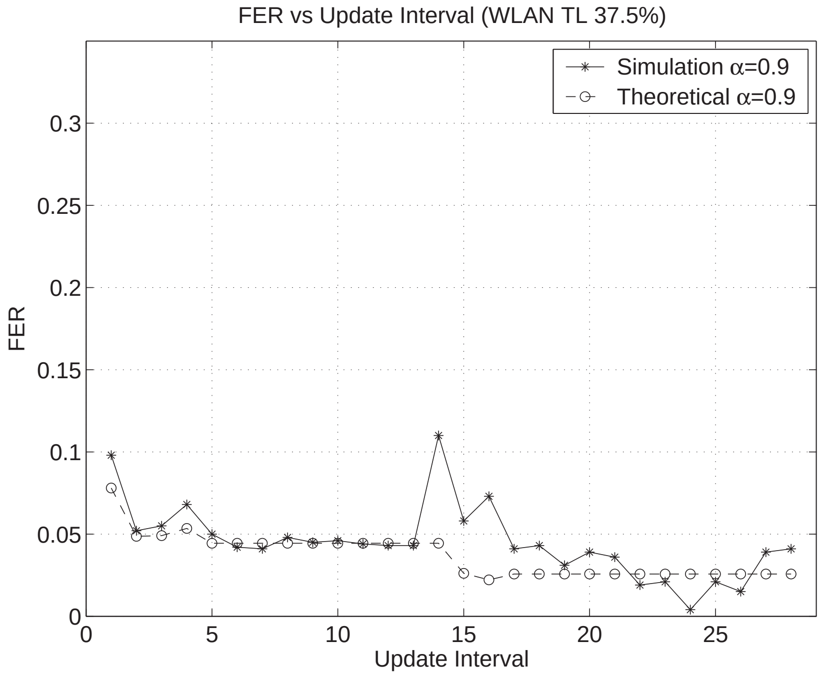

Figure A5: Theoretical vs Simulation: SAFH, $\alpha = 0.9$ (Traffic load of WLAN is 37.5%)

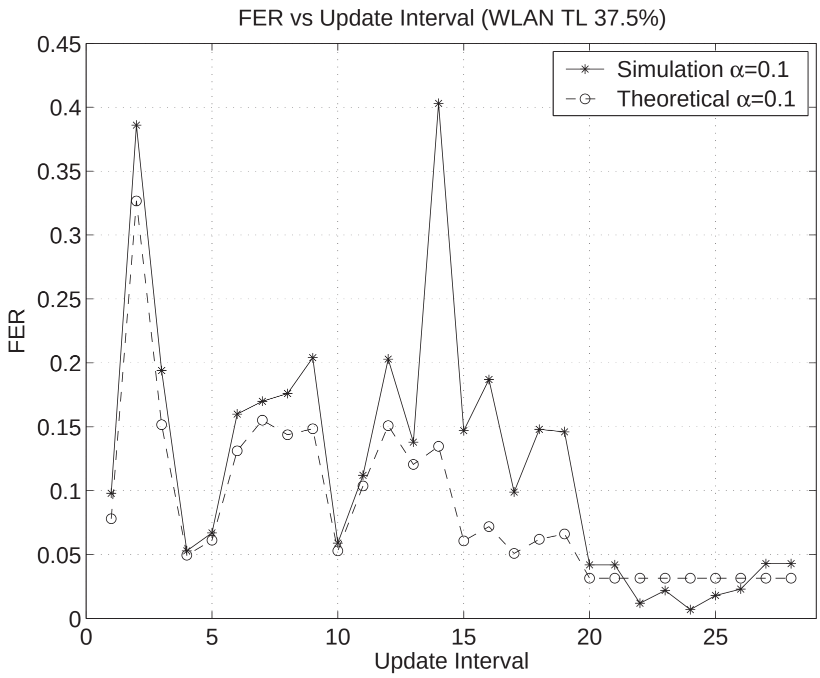

Figure A6: Theoretical vs Simulation: SAFH, $\alpha = 0.1$ (Traffic load of WLAN is 37.5%)