EVENT BASED FREQUENCY AGILE MECHANISM FOR IEEE 802.15.4 INTERFERENCE AVOIDANCE

Khalid Obaid

Master of Science Thesis 2011
Embedded System
Electrical Engineering
School of Engineering
Jönköping University
EVENT-BASED FREQUENCY AGILE MECHANISM FOR IEEE 802.15.4 INTERFERENCE AVOIDANCE

Khalid Obaid

This thesis work is performed at Jönköping Institute of Technology within the subject area of embedded systems. The work is part of the master's degree program offered at the university.

The authors are responsible for the given opinions, conclusions and results.

Supervisors: Youzhi Xu, Alf Johansson
Examiner: Youzhi Xu
Credit points: 20 points (D-level)

Date:

Archive number:
Abstract

Frequency hopping is one of the most successful solutions to the issues of interference and coexistence facing wireless devices that utilize the unlicensed 2.4GHz ISM band. However, the current scheme is based on Bluetooth frequency hopping mechanism and was designed to be implemented on capable devices that are “not” characterized by limited computational power and resources. In this work we propose a new frequency hopping technique primarily for use with low cost, IEEE 802.15.4 compliant devices that can be employed in wireless sensor networks. Unlike the frequency hopping scheme developed for Bluetooth, WirelessHART and ISA-100.11a, the proposed solution is event driven, thus allowing the network to change its operating frequency in the face of interference once problems are sensed in the channel without the need for extra computational and extensive scheduling activities. The design of the proposed algorithm was carried out using specification and description language (SDL) and the simulation of the SDL model revealed promising results. The proposed algorithm is then implemented on an IEEE 802.15.4 stack that was developed in NesC and TinyOS and is implemented on Micaz motes.
Summary

Increasing activities in the unlicensed 2.4GHz ISM band is causing an increasing interference and coexistence problems that are affecting the quality and reliability of wireless devices operating in close proximity using that frequency band. Filtering methods and modulation techniques are among the first line design level solutions however, experimental results revealed that such solutions must be coupled with extra measures to ensure coexistence and immunity to noise. Among the solutions proposed to the interference and coexistence problem is frequency hopping. It’s a proven technique that is mainly described in the Bluetooth standard. Most of the frequency hopping schemes in use today are time based in which changing frequency is governed by strict time scheduling constraints. Such a scheme is not always adequate for low cost wireless devices used in wireless sensor networks that are generally characterized by limited computational and power resources.

In order to utilise the benefits of frequency agility, an event based frequency hopping scheme is developed and presented in this work. The development process was conducted on the existing IEEE 802.15.4 MAC layer specifications and is carried out using the specification and description language SDL. The simulation results show a significant improvement in the overall performance upon the utilization of the proposed algorithm. A final proof of concept implementation of the proposed algorithm was carried out using an existing IEEE 802.15.4 stack implementation on nesC and TinyOS.

Key Words
Table of Contents

1 Introduction ........................................................................................................... 5
  1.1 BACKGROUND .......................................................................................... 5
  1.2 PURPOSE AND AIMS .............................................................................. 6
  1.3 DELIMITS ................................................................................................. 6
  1.4 OUTLINE .................................................................................................... 7

2 Theoretical background .................................................................................... 8
  2.1 SPREAD SPECTRUM ............................................................................ 8
    2.1.1 Direct Sequence Spread Spectrum (DSSS) ........................................ 9
    2.1.2 Frequency Hopping Spread Spectrum (FHSS) ................................. 11
    2.1.3 Colocation ....................................................................................... 12
    2.1.4 Throughput ..................................................................................... 14
    2.1.5 Narrowband and Broadband Interference ...................................... 15
    2.1.6 Multipath fading ............................................................................. 16
    2.1.7 Diversity in Time and Frequency ............................................... 18
  2.2 INTERFERENCE IN THE 2.4GHZ ISM BAND ........................................ 19

3 Related Work .................................................................................................... 21
  3.1 COLLABORATIVE MECHANISMS ....................................................... 21
  3.2 NON-COLLABORATIVE MECHANISMS ............................................. 22

4 Development Methodology ............................................................................. 24
  4.1 ALGORITHM VALIDATION METHODOLOGY ...................................... 24
  4.2 IMPLEMENTATION METHODOLOGY ................................................. 25

5 The IEEE 802.15.4 SDL Model .................................................................. 27
  5.1 BASIC STATE DIAGRAMS ................................................................... 27
    5.1.1 PHY layer Modelling ..................................................................... 27
    5.1.2 MAC layer Modelling .................................................................... 29
  5.2 IMPLEMENTED PRIMITIVES ............................................................... 31
  5.3 DATA TRANSFER MODEL ................................................................. 32
  5.4 MODELLING FRAME TYPES .............................................................. 33
  5.5 THE CHANNEL MODEL ....................................................................... 36

6 Event Based Frequency Agility .................................................................. 38
  6.1 MODIFICATION OF THE CSMA/CA .................................................. 39
  6.2 BEACON LOST INDICATION .............................................................. 40
  6.3 COORDINATOR NOTIFICATION ....................................................... 42
  6.4 CHANNEL HOPPING ........................................................................ 42

7 Verification and Software Implementation ............................................... 45
  7.1 IEEE 802.15.4 PLANE IMPLEMENTATION ....................................... 46
  7.2 FICA ALGORITHM ................................................................................ 47
  7.3 SOFTWARE IMPLEMENTATION ......................................................... 49

8 Conclusions and Recommendations ......................................................... 51
  8.1 RECOMMENDATIONS FOR FUTURE WORK .................................... 51

9 References ...................................................................................................... 52
List of Figures

FIGURE 1: DSSS SPREADING PROCESS [8] 10
FIGURE 2: DSSS SIGNAL ENERGY SPREADING 10
FIGURE 3: FREQUENCY HOPPING DESCRIPTION 12
FIGURE 4: FREQUENCY SUB-BANDS IN THE ISM 2.4GHZ AS USED BY A) IEEE 802.11 B) IEEE 802.15.1 AND C) IEEE 802.15.4 STANDARDS 13
FIGURE 7: STAGES OF THE OVERALL DEVELOPMENT PROCESS 25
FIGURE 8: SOFTWARE IMPLEMENTATION METHODOLOGY 26
FIGURE 9: IEEE 802.15.4 DEFINED LAYERS 27
FIGURE 10: SIMPLIFIED PHY LAYER STATE DIAGRAM 28
FIGURE 11: IEEE 802.15.4 SUPER FRAME STRUCTURE 29
FIGURE 12: SIMPLIFIED MAC LAYER STATE DIAGRAM 30
FIGURE 13: PHY-MAC PRIMITIVES 32
FIGURE 14: THE DATA TRANSFER MODEL UNDER CONSIDERATION [1] 33
FIGURE 15: FRAME STRUCTURE OF THE IEEE 802.15.4 34
FIGURE 16: THE IMPLEMENTED HIGH LEVEL FRAME FORMATS 35
FIGURE 17: THE CHANNEL MODEL 36
FIGURE 18: GENERAL DESCRIPTION OF THE FICA ALGORITHM 39
FIGURE 19: THE MODIFIED CSMA 40
FIGURE 20: BL/FICA ALGORITHM 41
FIGURE 21: MODIFIED FRAME STRUCTURE WITH THE FICA BIT INTRODUCED 42
FIGURE 22: MODIFIED MAC LAYER 43
FIGURE 23: VERIFICATION MAC LAYER DIAGRAM 45


1 Introduction

The unlicensed 2.4 GHz band became one of the most active communications bands in the RF spectrum as market demand for such communication devices extended from corporate to home environment. Today, different 2.4 GHz wireless equipment are being produced and purchased in greater volumes than ever before. The operation of such devices in close proximity to each other is leading to interference issues that are affecting the overall reliability of the 2.4 GHz wireless networks. At present, such issues are seriously affecting the application of wireless technology in the industrial environment.

In this work we present a new frequency agile technique. Unlike the existing time triggered frequency hopping mechanism that are mainly based on the Bluetooth frequency hopping technique, the proposed mechanism is event driven and allows the devices operating within a network to only change frequency whenever necessary. By maintaining the same working frequency link over longer periods of time the proposed method bypasses much of the synchronization issues without significant increase in messaging, computational or even memory requirements.

1.1 Background

Research in the field of network coexistence established frequency agility as one of the leading mechanisms for managing interference scenarios. Bluetooth is perhaps one of the principal examples demonstrating how reliable communication can be when utilizing multiple channels.

Bluetooth utilizes frequency hopping spreads spectrum. Unlike the direct sequence spread spectrum employed by the IEEE 802.15.4 standard [1] that uses only one communication frequency, frequency hopping spread spectrum modulation utilizes multiple channels while communicating with other devices. This is great for avoiding Narrow band interference as the blocking single will only affect the system when it is using a subset of the frequencies on its hopping sequence. However, this scheme is time triggered, which means that it will force the device to change frequency at regular intervals with or without the presence of interference, thus, imposing aggressive utilization of bandwidth and increased computational, messaging and synchronization requirements [2].

Attempts to improve the performance of IEEE 802.15.4 lead to the application of frequency agile communication concepts between such devices. A number of researches such as [3], [4] and [5] were dedicated to the study and application of frequency agile algorithm similar to the one utilized by
Bluetooth in direct sequence spread spectrum devices. In all cases, the experimental results revealed a clear performance improvement over the traditional single channel systems, however, the increased computational and synchronization requirement can raise questions over the applicability of such frequency agile solution in wireless sensor networks that are, in most cases, self-organized, not restricted by a star based topology and are mainly distinguished by power, memory and computational limitations [2].

1.2 Purpose and aims

In this work we focus on the interference issues and introduce a new frequency agile modification to the MAC layer specifications for the IEEE 802.15.4 standard. Unlike the Bluetooth-based frequency hopping scheme which is time driven, the proposed frequency hopping can be described as “Event based frequency hopping”. This means that the device will change frequency only upon the detection of an event indicating that the current channel is unusable due to interference effect that can be caused due to the existence of other networks operating on the same channel within the vicinity, multipath fading or any other natural phenomena.

The work is motivated by the necessity to develop a frequency hopping algorithm that can meet the limited abilities and requirements of wireless sensor networks. The thesis work will principally focused on the development of the event based frequency agile concept from idea to implementation on Micaz motes using nesC and TineyOS. A model based development will be employed using the Specification and Description language (SDL). Using SDL, a high abstract model based on the IEEE 802.15.4 specifications will be created and used to assist the incorporation of the proposed frequency hopping technique in the IEEE 802.15.4.

1.3 Delimits

Although the proposed solution may be extended to other wireless communication standards, it is important to note that this work is focused on IEEE 802.15.4 compliant networks that include devices operating in the unlicensed 2.4GHz ISM band. For simplicity, the development work will focus on star topology networks and the solution will focus on scenarios taking place after the association operations are completed. It should be clear that this work focuses on the development procedure for the event based frequency hopping and the implementation on top of TinyOS. Extensive testing of the new protocol using practical networks is left for future work due to time limits.
1.4 Outline

Following this chapter, the theoretical background section provides the necessary detailed theatrical bases for this work. It includes a discussion on different spread spectrum flavours and compares their performance against each other. An overview of related works will be presented in chapter 3. Following that, a presentation of our development methodology will be presented in chapter 4. Chapter 5 will present our specification and description language models used in the development. This is followed by the introduction of the event-based frequency agile concept along with the proposed modification to the IEEE 802.15.4 in chapter 6. Chapter 7 presents the results of the system-level simulations along with the resulting implementation of the proposed modifications. This work will be concluded and some recommendations for future work will be presented in chapter 8.
2 Theoretical background

The Modern mobile lifestyle of millions of people around the globe is contributing to an increased deployment of low cost radio communication devices. Today, most of the low cost wireless solutions such as the ones proposed for WPANs (Wireless Personal Area Networks), WLANS (Wireless Local Area Network) and WSN’s (Wireless Sensor Networks) are utilizing the free 2.4GHz ISM (Industry Scientific and Medical) unlicensed band. This sharing of the spectrum among various wireless devices, particularly those operating in the same vicinity and sharing the same environment may lead to severe interference issues and result in significant performance degradation [6]. To tackle those problems, different coexistence schemes aimed at improving the operation of heterogeneous devices in the same locality are being continuously developed.

To better appreciate the shared environment problems, it is quite essential to understand the differences between different devices sharing this wireless band. Network oriented devices using the 2.4GHz ISM band generally follow one of three main internationally defined standards. Namely: the IEEE 802.15.1 (Bluetooth), IEEE 802.15.4 (Zigbee, WirelessHART) and the IEEE 802.11 (Wi-Fi). There are of course other devices using the 2.4GHz ISM band such as the cordless phone, the car alarm and microwave oven [2], however, our discussion in this section will be focused on providing an overview of the first three main players, particularly their properties, modulation scheme, collocation issues and the effects of different types of interference and multipath on their operation. Following that, we will move to broadly review the interference problem in the 2.4GHz band caused by the operation of multiple devices of different standards simultaneously in the same environment.

2.1 Spread Spectrum

Spread spectrum refers to any modulation technique via which a transmitted signal appears to occupy a larger bandwidth compared to what it actually require. This technique was originally developed for military applications as part of the efforts to establish stealth and jamming resistant communications. Today, two main different flavours of this technique are being used in a variety of civil communication applications including all the three IEEE standards mentioned above [2]. It is important to appreciate the difference between traditional modulation schemes and spread spectrum. Both techniques are similar in a sense that the bandwidth of the transmitted signal is a function of the message to be transmitted.
Unlike traditional modulation techniques however, spread spectrum’s required bandwidth is also a function of a spreading signal known as the spreading code which is used in different ways depending on the flavour of the spread spectrum modulation. In both cases however, the combination of both dependencies results in a transmission bandwidth that is larger than normally required. By spreading the signal in the frequency domain, Spread Spectrum modulation techniques facilitate low power density and redundancy advantages over other modulation schemes. From a theoretical perspective, low power density is a desirable property because if utilized, the signal is theoretically immune against intruders and will not interfere with the operation of other communication devices in the vicinity [7].

In spread spectrum communication, low power density is achieved because the amount of energy per specific frequency will be low compared to the total transmitted energy. The same reason provides for redundancy which is related to the high resistance to noise and interference [7]. By spreading the signal over a larger bandwidth the signal will be present on different frequencies from which it may be recovered in case or errors. And as a result, the system can recover intended transmissions even when high noise levels are corrupting the channel. As mentioned earlier there are two main Spread Spectrum modulation flavours:

1. Direct Sequence Spread Spectrum (DSSS) and
2. Frequency hopping spread spectrum (FHSS).

In case of the 2.4GHz ISM band that spreads from 2.4 GHz to 2.4835 GHz, Direct sequence spread spectrum is used in both the IEEE 802.11 and IEEE 802.15.4 standards. While Frequency hopping spread spectrum is utilized by Bluetooth (IEEE 802.15.1) [2]. Both Spread Spectrum modulation techniques are composed of two consecutive modulation processes executed on the transmitted signal. The first one is the operation to spread the code. The second is to modulate the signal. Those two processes are different for different flavors of spread spectrum and will be discussed in the following sections.

### 2.1.1 Direct Sequence Spread Spectrum (DSSS)

As mentioned earlier, DSSS is the modulation of choice used in the IEEE 802.11 and IEEE 802.15.4 standards. As mentioned earlier, there are two modulation processes being executed on the signal. The first modulation process in this case multiplies the data being transmitted by a "noise" signal which is a pseudorandom sequence of 1 and −1 values switching at a frequency much higher than that of the original signal.
This pseudorandom sequence is called the “spreading code” and the process is known as “chipping”. The spreading process is described in figure 1 for spreading code is 11 chips having the pattern 10110111000 [8].

![Figure 1: DSSS spreading process][8]

The second process is the message modulation process and here the carrier is modulated PSK of the generated spread signal. [2] As a result of the spreading operation the pulse time gets shorter and the frequency band becomes larger, therefore the energy of the original signal is spread into a much wider band as shown in figure 2.

![Figure 2: DSSS signal energy spreading][8]

Direct sequence spread spectrum achieves redundancy as a direct result of the chipping process. The chipping operation results in presence of message bits on each chip of the spreading code. This, even if reception of some chips is corrupted, the receiver can still recognize the chip sequence and reconstruct the original signal [7].
2.1.2 Frequency Hopping Spread Spectrum (FHSS)

Frequency hopping spread spectrum is the modulation of choice for the IEEE 802.15.1 (Bluetooth) compliant devices. This flavor is similar to DSSS as both use a spreading code to spread the signal over a larger bandwidth, however, unlike the DSSS, the first modulation process in frequency hopping spread spectrum (FHSS) uses this spreading code to construct a “hopping sequence” according to which a frequency agile local oscillator will be controlled and moved around in the frequency domain [9]. Thus, unlike in DSSS, the spreading process can only be appreciated if the output is viewed over a longer time periods in the range of seconds. Following this spreading process, the actual data is modulated FSK, thus generating a narrow band signal for a duration of a dwell time. The signal is transmitted over a different frequency after each dwell time while coordination between the transmitter and the receiver is done according to the following general description [9]:

1. The transmitter sends a request via a predefined frequency channel (control channel).
2. The receiver sends a number sequence, known as a seed. Or in many cases the sender has its own number sequence stored in which case this step in not executed.
3. The transmitter uses the seed as one of the inputs in a random number algorithm, which then calculates the channel sequence, i.e. the sequence of frequencies that is used for communication.
4. The transmitter sends the channel sequence, channel stay time (same for all channels) and the time when it will start transmitting the data.
5. The communication starts at the same point in time, and both the transmitter and the receiver change their frequencies according to the channel sequence the output transmission can be represented in 3d as shown in figure 3 below.

It is quite clear that the above described procedure requires strict synchronization between the transmitter and the receiver. If the synchronization is lost then communication between the transmitter and the receiver will not be possible as both will be operating on different frequencies at the same time making it impossible for them to receive transmissions from one another.
In case of FHSS, redundancy is achieved through retransmissions over different frequencies. The system may fail to send data over one frequency but as the carrier frequency is changed periodically, channel conditions may be different and the system will eventually utilize a clear link. According to the specifications, the changing of frequency should be done periodically (Time based) and is done in dwell times that are typically in the range of 100ms [7].

2.1.3 Collocation

It is important to note how significant it is to have collocated systems without collisions. A collusion free environment has a marvellous impact on the aggregate capacity / throughput of the installation as well as its efficiency in terms of bps per Hz. In DSSS systems, it is possible to try collocating systems via using different spreading sequences for different active system. This technique is called Code Division Multiple Access (CDMA) [2]. For this technique to work however, the sequences used should be highly distinguishable from the other one (Orthogonality property ). If such orthogonal codes are employed then each receiver will be able to “read” only the information sent to it by the transmitter that is dedicated to it. Although the CDMA concept is valid in theory, however, it is important to note that the number of orthogonal pseudo-random sequences is limited and it is a function of the sequence length which is number of chips (bits) in the sequence. For the collocation of 16 systems, 255 chip (bit) long sequences should be used. In actual practice however, the number of chips used is much smaller [7]. The IEEE802.11 standard utilizes either the 11 bit long backer sequence [10] or the newer Complementary Code Keying (CCK), which consists of a set of 64 8-bit code words [11]. On the other hand, IEEE 802.15.4 2006 standard specifies the use of a set of 16 spreading codes (32-bit long each) one for each symbol value.
Due to this low processing gain (ratio of spread rate to bit rate [7]), the use of CDMA is impossible and system collocation can therefore only be achieved though the fixed allocation of bandwidth to different systems.

The IEEE 802.15.1 specifies 79 different hopes for the carrier frequencies. Basic collocation of different systems following the same standard can be achieved by using different non-colliding or minimum collisions hopping sequences. Theoretically speaking, the number of FHSS systems that can be collocated in the same vicinity is up to 26 systems. However, experimental results revealed that collisions will occur at a significant rate when this number is collocated. The experiments concluded that 15 different FHSS systems can be collocated in the same vicinity and that those systems are expected to operate with acceptable collision rates. Increasing the number of collocated systems further will cause adverse effects on the operation of all systems [7].

![Figure 4: frequency sub-bands in the ISM 2.4GHz as used by a) IEEE 802.11 b) IEEE 802.15.1 and c) IEEE 802.15.4 standards](image)

All the above is correct for the case in which the FHSS collocated systems operate independently in the absence of any form of synchronization that may govern their hopping sequences. Considering the theoretical case again, the theoretical number of FHSS systems that can be collocated will jump from 26 to 79 different systems each utilizing one of the different 79 defined
frequencies at a time if proper synchronization is utilized [7]. Although that is theoretically possible, practically speaking it would require some very expensive filtering solutions. To spin around this problem, actual products today require a 6 MHz wide partition and as a result, 12 systems may be collocated “without any collisions” [7]. Note that unlike the case of communication in licensed bands, this type of strict synchronization however is not always allowed and can hardly be achieved in the unlicensed 2.4GHz ISM band. As a result, the FHSS system may be operated in a synchronized or non-synchronized hopping sequence based on the region and located spectrum.

In the non-synchronized case, as the band is assigned dynamically among different collocated systems each using a different hopping sequences that is not synchronized with others. In this case collisions do occur, thus lowering the actual throughput of each system. The number of collisions occurring is directly proportional to the number of collocated systems in the same vicinity. For a small number of collocated systems, each additional system brings in almost all its net throughput; the amount of collisions added to the system is insignificantly low. However, when the number of collocated systems approaches 15, the amount of collisions generated by the addition of more systems is so high that in total they lower the aggregate throughput. The synchronized case is different, collisions are totally eliminated and as mentioned earlier, up to 12 systems can be collocated without any collusion. In this case, the aggregate rate and throughput is a linear function of the number of collocated systems [7].

Based on the IEEE 802.11 specifications, the maximum number of DSSS systems that can be collocated is 3, based on the IEEE 802.15.4 specifications, the number 16 systems can be collocated. These 3 collocated systems provide an aggregate rate of 3 x 11 Mbps = 33 Mbps, or a net aggregate throughput of about 3 x 7 Mbps = 21 Mbps [7]. Collisions between signals generated by collocated systems do not occur Because of the rigid allocation of sub-bands to systems; as a result, the aggregate throughput is a linear function of the number of systems.

2.1.4 Throughput

According to [18], the “Rate” of a system is defined as the amount of data (per second) carried by a system in its active state. As most communications systems are incapable of carrying data at the rate of 100%, the system’s “Throughput” was defined as the average amount of data carried by the system [18].
Theoretical Background

The average is calculated over long periods of time and obviously, it is always lower than the system’s rate. It is also important to consider the amount of overhead introduced by the communication system’s protocol while looking at the performance. For instance, Ethernet is said to have a rate of 10 Mbps but only a throughput of 3~4 Mbps [18].

Yet the number of actual data bits is even smaller. In short, the performance of a particular communication protocol is also affected by the amount of extra data it adds on to the actual data that needs to be communicated. DSSS systems following the IEEE 802.11 standard transmit at rates of up to 11 Mbps on a contiguous sub-band of 22 MHz; the efficiency of such systems is $11 \text{ Mbps} / 22 \text{ MHz} = 0.5$ bits / Hertz. A more comprehensive calculation for Similar DSSS following the IEEE 802.15.4 is given in [12].

FHSS-IEEE 802.15.1 systems transmit at rates of up to 3 Mbps with every hop utilizing a channel that is 1 MHz wide. This puts the efficiency of such a system at $3 \text{ Mbps} / 1 \text{ MHz} = 3$ bits / Hertz. The above numbers represent the rates of the systems. When we look at the throughput however, we find it around 7 Mbps for the IEEE 802.11 DSSS systems, and about 2 Mbps for 3 Mbps Bluetooth FHSS systems [7].

2.1.5 Narrowband and Broadband Interference

The discussion above established the main operating differences between Direct Sequence Spread Spectrum and Frequency Hopping Spread Spectrum techniques. In this section, we will take this discussion further and address other performance issues such as interference susceptibility of the two spread spectrum flavors. However, before diving into details of narrowband and broadband interference the sensitivity of both types of receivers should be noted. It was mentioned earlier that FSK is used with FHSS, thus, a typical FHSS receiver would operate with a signal to noise ratio (SNR) of around 18 dB; On the other hand, a DSSS based system which utilizes the more efficient PSK scheme would operate at a SNR which is as low as 12 dB [2]. It follows logically from this data that for the same level of transmitted energy, a DSSS is capable of operating at a higher efficiency than a FHSS spread spectrum under conditions of all band interference. Which also means that for the same transmission energy level a DSSS system will have a longer ranger as compared to a FHSS based system.
The above discussion may seem to favor DSSS over FHSS; however, that is only one part of the story. It is imperative to bear in mind that the “whole spectrum” in the 2.4GHz ISM band is 83.5 MHz wide. From our discussion on collocation we noted that IEEE 802.15.1 FHSS systems utilizes the whole ISM band while DSSS systems uses only a single sub-band that is 22 MHz wide in case of IEEE 802.11 and 3 MHz wide in case of IEEE 802.15.4. The probability of an interfering signal to cover a 22MHz range is higher than that of it covering the whole ISM band range whereas the chances of the same interfering signal covering a 3 MHz range is even greater.

Thus, while a 15 MHz interference signal may badly affect the operation of an IEEE 802.11 DSSS based system, it will completely block an IEEE 802.15.4 DSSS signal operating in the interference signal range. However, a FHSS system will be only be affected during about 30% of its hop sequence, thus it can still operate at about 60% of its total capacity under the said conditions. DSSS modulation scheme dictates that the receiver must receive all signals present in its 22MHz or 3MHz operating sub-band [2]. Heavy filtering at the receiver side may cause the loss of incoming transmissions that are in most cases hardly above noise level due to the low power density characteristics of DSSS. Unfortunately, this makes DSSS receivers’ susceptible to narrow band interference as such unwanted signals around a single frequency are allowed in by the filtering mechanism. If the interfering signal carries enough energy, the receiver may be totally blocked [7].

The case is different for FHSS systems on one hand primarily because they utilize a more traditional narrowband modulation mechanism as was discussed earlier. As a result, FHSS systems utilize a much narrower band pass filters than the ones used in DSSS [2]. By using such filters, the system ensures that a narrow band interference signal present on a specific frequency, will block only specific set of hopes depending on the bandwidth of the blocking signal. The FHSS transmissions will be affected during the hopes on the corrupted frequencies yet as the hopping sequence progresses to non-corrupted frequencies the narrow filter will reject the interfering signal, and the system will operate normally without disturbances.

2.1.6 Multipath fading

Transmitted signals in the wireless channel never follow a single path towards the receiver. In fact, many naturally occurring phenomena such as reflection of the transmitted signal off a reflecting object (such as a building, a wall or even a desk in an office) and refraction contribute to the corruption of the signal and even creating multiple copies of it that are shifted in phase and time [13].
Those different copies are created because the signal follows different paths of varying lengths to reach its destination. As a result, the signal propagation time is different from one path to the other and different copies of the original transmission arrive at different time intervals. The chipping process used in DSSS increases the rate of the signal to be transmitted. The symbols of chipped signal are much shorter than those generated by a FHSS system for the same data rate. According to [7] An IEEE 802.11 DSSS system utilizes 11Mchips/sec over the air, i.e. pulses of 90ns width while FHSS systems transmitting at 3 Mbps use pulses of 330ns width.

While a narrow pulse is quite sensitive to time shifts as compared to a wider one, it follows that the FHSS systems are more resistant to the presence of multipath effects than their DSSS counterparts. The multiple copies of the original signal arrive at the receiver with different instantaneous amplitudes and phases. Simultaneous mixing of those random signals during propagation and at the receiver will result in additive and subtractive effects in which some frequencies cancel each other, while other sum up. This is called selective fading of frequencies in the spectrum of the received signal [13].

As was discussed earlier, FHSS systems transmit a narrowband signal that is located around different carrier frequencies in different times. If at a particular time, the system is utilizing a frequency where selective fading is taking effect, the system will be affected only during that hop and the receiver will not detect enough energy to distinguish the transmission from noise. However, retransmission, probably at a different frequency, can compensate for this loss of information [7]. To the contrary, DSSS systems use a wideband signal and as was discussed earlier, the information is spread.

Unlike FHSS systems the DSSS system may be able to pick up the transmission that is significantly faded due to its spreading properties, however, there can be problems due to time shifts. In addition, when the average signal in the communication channel is very low, DSSS may not have the capability to detect the signal, and in this case, it will have to follow the same retransmission compensation policy but unlike FHSS systems that will be again on the same frequency and thus the same fading issues will still apply. The Multipath issues discussed here are all a function of the transmission rate [7], meaning that an 11 Mbps DSSS systems are much more sensitive to such effects than 2 Mbps DSSS systems, thus an IEEE 802.15.4 system should be less sensitive than the IEEE 802.11 systems when it comes to multipath. But again, the transmitted power is lower and that should be taken into consideration.
According to [7], DSSS systems can accept delay spreads that are generated by multipath in the typical range of about 500 ns when operating at 1 Mbps, going as low as 70 ns when operating at 11 Mbps. [7] provides a broad view of typical values according to which, a typical delay spread in a room is in the range of 20-30 ns, while in a shopping mall with big open spaces and reflective surfaces, the delay spread reaches 200-400 ns. Thus, as stated in [7] it can be concluded that DSSS is efficient as long as the delay spread is controlled by either

a) Allowing multipath propagation, but keeping the delay spread at low levels by using DSSS systems in small areas, such as offices, for WLAN applications. OR

b) By significantly reducing the number of reflected paths, forcing all the energy to follow just in just few of the possible paths that are of similar lengths. In this case, the multiple copies reaching the receiver are shifted only by small amounts in time. To achieve this, DSSS systems should be used with directional antennas, for point-to-point applications.

Based on the discussion in this section, it can be concluded that the FHSS is more suited for long distance point – to multipoint transmissions as in such environments; DSSS will face multipath issues that will limit its operation.

2.1.7 Diversity in Time and Frequency

As per the specification requirements, both FHSS and DSSS will have the option to retransmit their packets until receiving an acknowledgment from the intended receiver. this is part of the compensation policy for lost transmissions that was discussed earlier in the previous sections. The capability to do this is known as “Time Diversity”[7]. As mentioned earlier, the problem with DSSS is that the time diversity property is preformed utilizing only one frequency, thus all transmissions will be subjected to the same fading and multipath effects presented in the used channel. Thus a faulty retransmission is more likely and that will cause more retransmissions.

It was also noted earlier that FHSS systems use different frequencies as part of their hopping sequence causing the probability of a retransmission to occur at a different frequencies more likely. This means that the time diversity in FHSS is complemented with “Frequency Diversity” and as fading conditions vary on different channels, the probability of repeating a faulty transmission is minimized.
2.2 Interference in the 2.4GHz ISM Band

Throughout the above sections we mentioned the unlicensed 2.4 GHz ISM band. It is significant and interesting to note that this band allows for primary and secondary uses and that this thesis is focused around the Secondary use of this band that is free but regulated by rules relating to total radiated power and the use of the spread spectrum modulation schemes and defined in the Federal Communications Commission Title 47 of the Code for Federal Regulations Part 15 [14]. The given rules do not address Interference among various applications is as long as the rules are followed. Thus, sharing is the major drawback of using the free ISM band where devices must coexist and potential interference tolerated.

According to [6], spread spectrum and power rules defined in [14] are quite enough for dealing with multiple band users, provided that there exist a good physical separation between different radios. However, the case is different for closely located radios and multiple users including self–interference of multiple users of the same application. As was explained in previous sections, significant performance degradation will occur in such a case due to increased transmissions that will effectively rise the noise floor in the communication environment. Such effects become even more significant when all devices share the same sub-band in close proximity. It follows from this discussion that the interference problem is characterized by a time and frequency overlap as illustrate in figure 5 [6]. The figure demonstrates what may happen when a FHSS device that is occupying a 1 MHz band in the spectrum is overlapping with a DSSS device using a 22MHz band. It should be clear that the frequency hopping sequence and the traffic distribution of both communicating devices directly affect the collision overlap time.

![Figure 5: Time frequency overlap of Interference in the ISM band](6)
Interference among systems from the same time falls under the collocation issue that was discussed earlier in its dedicated section. It was obvious from the discussion that though such interference can be significant yet those issues are being considered early in the design process of the system and even by the international community defining the protocol [6], thus, leaving the worst realistic interference scenario to occur when heterogeneous devices are communicating in the same vicinity on the same channel. Addressing this coexistence problem in this scenario is the primary focal point of most of the research conducted today. Soon after the publication of the first IEEE standards, different attempts to quantify the impact of interference on the operation of both DSSS and FHSS devices were underway. According to [6], at least three categories can be used to classify the obtained results depending on the research methodology that was employed. This can be mathematical analysis, simulation or experimental measurements. Examples of early mathematical analysis include the works of Shellhammer [15], Ennis [16], and Zyren [17] for the WLAN packet loss and by Golmie [18] for the Bluetooth packet error.

In the viewpoint presented by [6], a first order approximation can be obtained following such analytical methods that put the performance degradation in terms of packet loss up to 25% for Bluetooth and up to 70% for WLAN [6]. Note that such analysis is normally based on a number of assumptions concerning traffic distributions and the operation of the Media access protocol, thus making them quite unrealistic. In addition, most of the analytical methods ignore mutual interference in order for the analysis to be tractable.

Results of experiments conducted by Kamerman [19], Howittt et al [20], and Fumolari [21] are considered by [6] to be more accurate but at the same time, very specific to the implementation tested while the use of simulation and modelling is considered to be a middle ground between testing and analytical results. The accuracy of this middle approach is however a function of the modelling assumptions made, just like the case of the mathematical analysis. According to [6], Some of the early simulation results were presented by Zurbes [22] for a number of Bluetooth devices located in a single room. The results of the study showed that for 100 concurrent web sessions there is 5% performance degradation. In [23] a detailed MAC and PHY simulation framework to evaluate interference problems was used. A similar result was obtained by Lansford et al [24] who use simulation and experimental measurements to quantify the interference resulting from Bluetooth and IEEE 802.11. According to [6] the simulation models were based on a link budget analysis and a Q function calculation for the channel and PHY models respectively, in addition to the MAC layer behaviour.
3 Related Work

Designers of wireless systems have always been faced with the challenge of interference from natural sources and other communication devices. As a result, the traditional wireless system design cycle always contained a step where channel impairments were either measured or predicted before selecting a modulation scheme and other signal preconditioning and processing steps at the transmitter and the receiver respectively. Most of the interference solutions proposed for the 2.4GHz ISM band however are different from the classical methods as they focus instead on signal processing control strategies including power and frequency hopping control and MAC parameters adjustment and scheduling [6].

Collaborative industrial activities focusing on coexistence issues in the 2.4 GHz ISM band lead to the foundation of the establishment of the IEEE 802.15.2 coexistence task group in order to evaluate the performance of Bluetooth devices that are likely to interfere with WLAN systems. The aim was to develop a coexistence framework consisting of a set of recommendations and best practices for better coexistence. In addition possible suggestions to modify the current IEEE 802.11 and IEEE 802.15.1 standards so as to better facilitate collaborative operation between the two heterogeneous systems are also expected from the IEEE 802.15.2 coexistence task group. The Bluetooth Special Interest Group (SIG) also formed its own task group on Coexistence concurrently with the IEEE. Both the Bluetooth and the IEEE working groups maintain liaison relations and are looking at similar techniques for alleviating the impact of interference [6]. Proposals from both groups range from collaborative schemes intended for Bluetooth and IEEE 802.11 protocols to be implemented in the same device to fully independent solutions that rely on interference detection and estimation [6]. In this section, we review the difference between the two schemes and present some of the related work done in this area.

3.1 Collaborative Mechanisms

Collaborative schemes for the coexistence of Bluetooth and IEEE 802.11 were proposed by the IEEE 802.15.2. According to [6] those are based primarily on MAC time domain scheduling to alter the transmission between the two technologies. One of the solutions assumes both technologies are implemented in the same device using a single transmitter [25]. In this scheme, voice packets transmitted via Bluetooth are given priority over data transmissions via WiFi, however, the data packets transmitted via Bluetooth will have lower priority than those transmitted over WiFi.
Another interesting recent collaborative solution was proposed in [26]. In this scheme coexistence is achieved via exploiting the cyclic nature of wireless sensor networks operation using a central arbiter that assigns medium resources according to requests coming from WSN coordinators. However, as with the case of all collaborative mechanism, the solution is expensive and requires extra hardware making it sometimes impractical depending on the application.

### 3.2 Non-Collaborative Mechanisms

According to [6], non-collaborative mechanisms are a range of schemes in which heterogeneous devices will try to detect interference from other networks on their own and adapt to the situation without any collaboration with the interferers. Adaptive frequency hopping described in [27] along with packet scheduling and traffic control described in [28] are examples of non-collaborative coexistence mechanisms. Both [27] and [28] use similar techniques for channel assessment such as measuring the bit or frame error rate, the signal strength or the signal to interference ratio (often implemented as the Received Signal Indicator Strength (RSSI)). The devices will then use the collected information to modify their hopping sequences in case of adaptive frequency hopping or even chose not to transmit on certain frequencies if those are occupied as in MAC scheduling.

Both of the above mentioned technique has advantages and disadvantages. It was indicated in [6] that one of the advantages in using a MAC scheduling policy is that it does not require any changes in the FCC rules. In fact, title 47, part 15 of the FCC rules on radio frequency devices specified in [14], allows a frequency hopping system to recognize the presence of other users within the same spectrum band so that it adapts its hop sets to avoid hopping on occupied channels. Furthermore, scheduling in the Bluetooth specifications is vendor implementation specific [6]. Thus, it can be easily to implement a scheduling policy using currently available Bluetooth chip set. On the other hand, adaptive frequency hopping requires changes to the Bluetooth hopping pattern and therefore a new Bluetooth chip set design unless such mechanism is implemented in software on a higher level. While both techniques can reduce the Bluetooth packet loss and the impact of interference on the other system, yet maximization of throughput can only be achieved via adaptive frequency hopping. According to [6], only the adaptive frequency hopping technique can increase the Bluetooth throughput by maximizing the spectrum usage.

Analysis of the performance of three opportunistic multichannel MAC protocols namely the “max SNR”, “min load” and “max throughput” was carried out in [29]. All three proposed protocols associated new coming
Related Work

terminal station to one of the access points available within its transmission range by selecting the access point with best signal-to-noise-ratio transmission link, by selecting the access point with lowest load, and by selecting the access point which provides the best throughput among all access points respectively. The results of the simulations revealed that all three proposed protocols achieve tremendous gain in an opportunistic and distributed manner for a limited amount of channel state information at the transmitter and that, even for a small number of access points.

Another interesting frequency hopping scheme for the MAC layer was proposed in [4] while a cognitive framework for improving coexistence was proposed in [30], the proposed protocol in [4] will facilitate for the node to hope to a different frequency in every CSMA/CA slot. Results from [4] revealed that the proposed protocol can effectively boost throughput and remove temporal starvation. The framework in [30] uses sensing based resource management to effectively manage the available infrastructure system to suppress interference to close by ad-hoc peer to peer links. This is achieved through by utilizing its superior communication resources to estimate interference conditions and judiciously allocates transmission power such as to minimize interference while a rate constraint ensures that the infrastructure system continues to meet a specified quality-of-service level. The analysis in [30] demonstrated very promising results.

![Figure 6: Coexistence mechanism space [6]](image)

Figure 6 illustrates the coexistence mechanisms space with respect to the duty cycle or the device activity and frequency band occupancy from [6] perspective. According to the figure, as the number of interferers increase, each system is forced to transmit less often in order to avoid collisions. Thus, as the band occupancy increases, the duty cycle is reduced imposing time domain solutions. Frequency domain solutions such as adaptive frequency hopping can only be effective when the band occupancy is low [6].
Development Methodology

4 Development Methodology

In the above theoretical background discussion we have established the differences between the two major modulation schemes that are used in the 2.4GHz ISM band, namely the DSSS and the FHSS. The comparison that followed established that although FHSS employ’s an inferior FSK modulation scheme, yet its frequency agile method constitute a leading anti-interference and multipath solution. It was also established that current channel hopping schemes requires extensive time synchronization to ensure that the transmitter and its intended recipient are always operating on the same frequency while they continuously change it otherwise; transmissions will never reach their intended destinations. This problem is quite complicated for the two device case in which one device acts as the transmitter while the other is the receiver. The picture becomes even more complex when an entire network is under consideration where devices at different ends of the network need to communicate with one another. In such cases, routing protocols come into play and frequency agility tends to obscure the operation [3]. It can be argued that frequency agility is currently supported by radio chips which have now become very efficient at switching between frequencies in less than 192µs [2]. Here it should be remembered that switching at high rate will drain the device’s energy resources and that can be quite a problem in wireless sensor network applications.

Combining DSSS and frequency agility is a valid solution, however, the problem is that the frequency agile methods used today are based on the FHSS implementation and that was not developed for applications characterized by limited resources and changing topologies as in the case of wireless sensor networks. Hence, implementing Bluetooth frequency hopping solution on a technology like the one described by the IEEE 802.15.4 is not entirely appropriate. Thus an alternative algorithm should be developed for such applications. The followed methodology in this work is divided into two sections one dealing with the validation of alternative algorithm while the other deals with actual implementation. The two parts and the relationship between them are described in the following sub sections.

4.1 Algorithm Validation Methodology

The validation of the algorithm is carried out through system level simulations. This means that we opt to initially implement the relevant subset of the IEEE 802.15.4 stack in Specification and description language (SDL) and carry out
system level simulations using the IBM SDL suit. SDL models are quite popular nowadays in the development of communication protocols. In fact, according to [31a], SDL is the most widely used FDT (Formal Description Techniques) in the area of telecommunications. Unlike other FDT’s, SDL supports object-oriented software design and is very quick to learn. In addition, its graphical notation are self-documenting, and thus, easily maintainable [31]. Using SDL to develop communication software allows the developer to formally describe the functional aspects of the system, which turn facilitate the detection and remediation of functional errors early in the development cycle.

Furthermore, the high level nature of the SDL combined with its modular nature allows for the description of algorithms and systems at a higher level with minimum details. As the design process moves forward, more details can be added to better capture system performance. For all those mentioned reasons, SDL modeling technique was selected as our initial design and validation method. Therefore, for our purpose, we make use of the modular characteristics of SDL described above to implement a higher level description model which captures the relevant subset of the protocol’s stack in SDL. The model can then be used to evaluate our proposed algorithm without going through the hassle of full protocol implementation.

### 4.2 Implementation Methodology

As mentioned earlier, the purpose of the SDL simulation is to acquire an overview for the proposed algorithm and to test the logic for errors. It should be noted however, that adapted algorithm validation methodology is carried out as part of a second stage in a global software implementation methodology. The rest of the stages are described in figure 7.

![Figure 7: Stages of the Overall Development Process](image-url)
The proposed software implementation methodology can be described as being an evolutionary implementation process in which the final software is developed following different iterations undertaken to develop the best possible implementation of the proposed algorithm described in section 4. This means that the proposed software development method covers both state 2 and 3 of the overall development process. The implementation process is described in figure 8.

Following the SDL implementation and verification, an opportunistic external software reuse principle [32] is applied by which an existing IEEE 802.15.4 stack developed in nesC and TinyOS [33] by the Open-ZB community[34] is reused and modified according to developed SDL model. The output is a proof of concept implementation of the developed algorithm.
5 The IEEE 802.15.4 SDL Model

The purpose of this implementation is to develop an alternative frequency agile algorithm that can be used with low cost, limited infrastructure wireless devices. The implementation basically targeted the IEEE 802.15.4 standard and follows the development process described in the methodology section of this document. This section will present the high abstraction SDL model of the IEEE 802.15.4 that was created as part of the overall development process.

5.1 Basic State Diagrams

IEEE 802.15.4 is a standard defined for Low Rate Wireless Personal Area Networks with characteristics that are quite suitable for WSN applications. The IEEE 802.15.4 defines only the first two layers of the open system interconnection stack, namely the Physical Layer (PHY) and the Medium Access Control (MAC) (figure 11). In this section we describe our high level implementation of the subset of the described functions in this standard.

![IEEE 802.15.4 defined layers](image)

**Figure 9:IEEE 802.15.4 defined layers**

5.1.1 PHY layer Modelling

According to the IEEE 802.15.4 specifications document [1], the PHY layer, is responsible to carry out functions such as hardware control, In-channel power energy detection, Link quality indication, Communication channel selection, Clear Channel Assessment (CCA), and Packet transmission and reception through the radio channel. The constructed model should capture most of the described functions for this layer using the extended finite state machine approach which is used in SDL.
Thus, following the above described approach, the first modeling step would be to derive a state machine diagram that reflects the operation of the physical layer as per its specifications. The developed state diagram is shown in figure 10 and contains three main states namely the transceiver off state (TRX_OFF), the transmitter on state (TX_ON) and the receiver on state (RX_ON). The device is allowed to transmit or receive only in the TX_ON or RX_ON states respectively. Note that during energy detection scan the device will discard all useful received information despite having its hardware in receiver on mode [1]. The same is true when the device is carrying out clear channel assessment. For this reason, two separate states were introduced to distinguish the ED_SCAN and CCA states from the RX_ON state where useful information is received and processed.

![Figure 10: Simplified PHY layer state diagram](image)

Transition between different states in the physical layer is mainly carried out on the reception of the SET_TRX_STATE primitive from the MAC layer. The only exception is the transition to ED_SCAN and CCA states as described, those are done on the reception of the ED_SCAN and CCA primitives described in [1]. For the purpose of this high abstraction model, a subset of the primitives are implemented and those will be discussed in more details in the relevant section of this chapter.
5.1.2 MAC layer Modelling

As the name suggests, the “medium access control” (MAC) layer is responsible for regulating access to the communication medium or channel using a timed process that involves controlling the state of the PHY layer. According to IEEE 802.15.4 specifications, there are two different types of operating schemes available. The first one, called beacon enabled mode, is based on CSMA/CA with a superimposed time-slot allocation, which schedules the network access. The second one, called non-beacon mode, is based on a pure CSMA/CA protocol. This work will mainly concentrate on the beacon enabled slotted CSMA/CA mode of operation.

In the beacon enabled mode, the coordinator is required to transmit a beacon frame periodically. The period after which the device will transmit or expect a beacon is known as the super frame and it can be divided into three main sub periods namely the contention access period (CAP), the contention free period (CFP) and the inactive period. Those are shown in figure 11.

![Figure 11: IEEE 802.15.4 super frame structure](image)

During the CAP, devices compete for channel access via the CSMA/CA mechanism while during the CFP, permitted devices access the channel through time division multiple access. CFP slots are allocated from the CAP slots, thus, the CAP period will expand or shrink depending on how many devices are allocated a guaranteed time slot (GTS). Using the above described super frame structure; it is possible to construct a simplified state diagram for the MAC layer where the main states are IDLE, CFP, CAP and CSMA respectively. Transition from IDLE to CAP is done when the device receives/sends out a beacon. Following that, a counter will control the transition to the CFP depending on the CSMA/CA slots allocated to the CAP, following that a similar counter is used to keep track of the number of CFP slots before the
transition to IDLE state and the commence of the next beacon interval. The simplified state diagram is shown in figure 14. This state diagram together with the state diagram of the PHY layer shown in figure 10 are the basis of our IEEE 802.15.4 high abstraction SDL implementation.

Unlike the physical layer, transition from one state to the other in the MAC layer in the beacon enabled mode is based on timing constraints. For our implementation, we use three main timers, one is the Super frame timer that is set to the BI (Beacon Interval) while the second is the SD (Superframe Duration) timer that is set to one SD period (see figure 11).
A back-off timer is used for the CSMA/CA time keeping. In case of the coordinator, transactions are based solely on those timers while in case of End devices, the reception of a beacon will contribute to the transaction process as shown in the state diagram above.

5.2 Implemented primitives

Communication of data and other control functions are exchanged between lower and higher layers via using the concept of primitives described in [1]. As already mentioned in the above section, only a subset of the complete primitive set and services provided by the MAC and PHY layers are implemented in this SDL model. The implemented primitives are described in Table 1.

<table>
<thead>
<tr>
<th>Primitive name</th>
<th>From</th>
<th>To</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC_START_REQUEST</td>
<td>Higher*</td>
<td>MAC</td>
<td>Primitive to start operation of the MAC layer</td>
</tr>
<tr>
<td>MAC_DATA_REQUEST</td>
<td>Higher*</td>
<td>MAC</td>
<td>To request sending data</td>
</tr>
<tr>
<td>SET_TRX_REQUEST</td>
<td>MAC</td>
<td>PHY</td>
<td>To change the state of the physical layer</td>
</tr>
<tr>
<td>CHANNEL_SET</td>
<td>MAC</td>
<td>PHY</td>
<td>To change the channel</td>
</tr>
<tr>
<td>CCA_REQUEST</td>
<td>MAC</td>
<td>PHY</td>
<td>To request the physical layer to perform CCA</td>
</tr>
<tr>
<td>ED_REQUEST</td>
<td>MAC</td>
<td>PHY</td>
<td>To request the physical layer to perform ED scan</td>
</tr>
<tr>
<td>PHY_DATA_REQUEST</td>
<td>MAC</td>
<td>PHY</td>
<td>To request the physical layer to send data</td>
</tr>
<tr>
<td>CONFIRM</td>
<td>PHY</td>
<td>MAC</td>
<td>General purpose confirm from PHY to MAC</td>
</tr>
<tr>
<td>ED_CONFIRM</td>
<td>PHY</td>
<td>MAC</td>
<td>Confirmation that the ED scan is complete</td>
</tr>
<tr>
<td>PHY_DATA_INDICATION</td>
<td>PHY</td>
<td>MAC</td>
<td>Indication that a frame was received</td>
</tr>
<tr>
<td>MAC_CONFIRM</td>
<td>MAC</td>
<td>Higher*</td>
<td>General purpose indication from MAC to higher layers</td>
</tr>
</tbody>
</table>

*Higher layers, e.g. network layer

The MAC_START_REQUEST and MAC_DATA_REQUEST primitives are used by any layer higher than the MAC layer to request to MAC layer to commence operations and to transmit some data respectively. The MAC layer on the other hand will respond to such requests using the MAC_CONFIRM general indication which will indicate if the operation was successful or failure. The MAC layer can control the state of the PHY layer using the primitives SET_TRX_REQUEST, CCA_REQUEST and ED_REQUEST, while it can retune the radio to use another channel using the primitive CHANNEL_SET. Data transmissions
are requested using the preemptive PHY_DATA_REQUEST. The PHY layer on the other hand will respond to such requests using the CONFIRM primitive which is similar to the MAC_CONFIRM primitive described for the MAC layer. On the reception of data from the medium, the PHY will notify the MAC layer using the primitive PHY_DATA_INDICATION. The interaction between the PHY and MAC processes using the implemented primitives is shown in figure 13 below:

![Figure 13: PHY-MAC primitives](image)

### 5.3 Data Transfer Model

There are three types of data transfer models specified in the IEEE 802.15.4 [1]:

1. Data transfer from a device to a coordinator in
2. Data transfer from the coordinator to device.
3. Data transfer between two peer devices.

According to [1] only two of these transactions are used in a star based topology because data may be exchanged only between the coordinator and an end device. The mechanisms for each transfer type depend on the networks supports for beacons transmission. A beacon-enabled PAN is used in networks that either require synchronization or support for low latency devices. If the network does not need synchronization or support for low-latency devices, it can elect not to use the beacon for normal transfers. However, the beacon will still be required for network discovery [1].
As the main focus of this work is star based topology, Data transfer between network end devices and coordinator will be considered.

When a device wishes to transfer data to a coordinator in a beacon-enabled PAN, it first listens for the network beacon. When the beacon is found, the device synchronizes to the Superframe structure. At the appropriate time, the device transmits its data frame, using slotted CSMA-CA, to the coordinator. The coordinator may acknowledge the successful reception of the data by transmitting an optional acknowledgment frame [1]. This sequence is summarized in Figure 14.

![Image](image.png)

*Figure 14: the data transfer model under consideration [1]*

### 5.4 Modelling Frame Types

There are four types of frames defined by the IEEE 802.15.4 namely [1]:

- A beacon frame, used by a coordinator to transmit beacons
- A data frame, used for all transfers of data
- An acknowledgment frame, used for confirming successful frame reception
- A MAC command frame, used for handling all MAC peer entity control transfers
The frame structure of the IEEE 802.15.4 [1] is presented in figure 15. As can be observed, the frame control field in the MHR is common to all types, in fact according to IEEE 802.15.4 specifications, this field contains the necessary information to distinguish between different frame types along with other control information such as ACK request. The length of this field is two octets and its structure is described in table 2.

Table 2: Frame Control Field Content

<table>
<thead>
<tr>
<th>Bits</th>
<th>Frame Type</th>
<th>Security enable</th>
<th>Frame pending</th>
<th>Ack request</th>
<th>PAN ID</th>
<th>Reserved</th>
<th>Destination addressing mode</th>
<th>Reserved</th>
<th>Source addressing mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7-9</td>
<td>10-11</td>
<td>12-13</td>
<td>14-15</td>
<td></td>
</tr>
</tbody>
</table>

Figure 15: Frame structure of the IEEE 802.15.4

For the purpose of the high level SDL model, and keeping in mind the data transfer model considered in the SDL implementation, a simplified subset of
the fields shown in figure 15 were implemented using a type STRUCT in SDL. This implementation is described in figure 16. The NetType field is used to indicate the type of transmission and is used for simulation purposes only to indicate the type of modulation used by the device. Thus a frame with a NetType equal to Zigbee will be part of the IEEE 802.15.4 network that is being simulated.

Another extra field added to the frame is the Energy field which is used with the channel modeling. When the physical layer receives a frame with sufficient energy level and correct NetType, it will pass the content of the frame to higher layers, if the content of the energy is lower than the specified threshold, the frame will be discarded. If the energy is above threshold but the NetType is different, then the frame will be discarded, however, the level of the energy received will be used to assess the channel condition in the vicinity of the receiving device.

As described earlier, once a frame is received the physical layer will use the primitive PHY_DATA_INDICATION to signal the Mac layer and deliver the received content. Once the Mac layer receives the content, it will examine the frame type, and then copy the payload of the content to a Choice type (equivalent to a union in C language) which is used to implement different frame types. The content of the frame will be handled depending on its type and the data it contains.

<table>
<thead>
<tr>
<th>NetType</th>
<th>Content</th>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FrameType</td>
<td>PayLoad</td>
<td></td>
</tr>
</tbody>
</table>

Choice:

- **Beacon**
  - Source
  - Channel
  - Payload

- **Data**
  - Source
  - Destination
  - Payload

- **Mac Command**
  - Source
  - Destination
  - Command

- **Ack**
  - Source
  - Destination

*Figure 16: The Implemented high level frame formats*
5.5 The Channel Model

In order to simulate the wireless environment a simple SDL description was created that is based on a broadcast model and accounts for phenomena’s such as fading and propagation losses using the energy value stored in the transmitted frame energy field. All devices connected to the channel will transmit frames with energy value equal to 20.

Once the frame enters the channel, the channel model will generate a random number from 0-20 then subtract it from the stored energy value and broadcast the frame with the new energy value to the first device connected to the channel, the process is repeated and a new random number is subtracted from 20 for each frame sent out to each device. This process is continued until the received frame is broadcasted to all devices connected to the channel. Hence different devices will receive the same frame with different energy value. This is how multipath fading effects and propagation losses are simulated in the channel.

Once a device receives the frame from the channel the physical layer will check the energy of the device and depending on that value a decision will be made to either discard the frame if the energy level is below the threshold or to send it up to the MAC layer in case enough energy is present. The value of the received energy will be retained however to be used in clear channel assessment.

![Figure 17: The Channel Model](image-url)
This value will be decremented periodically by a random number. A clear channel is an indication that the value of the stored energy level is below the specified threshold. In our SDL implementation, 4 channels are being used, each of the channels has two states one with high interference while the other is with low interference. They differ in the limits of the random number generators. A transition between the two channels is done on the reception of a control signal that is sent randomly to one any of the channels by a channel control process.
6 Event Based Frequency Agility

To overcome the issue of continuous hopping and reduce switching requirements, while preserving the advantages of frequency agility in wireless application, we propose an “Event Based Frequency Agile” concept. We believe it is much more suitable for wireless applications with limited resource. It is a trade off between the frequency hopping scheme used in IEEE 802.15.1 and the use of a single channel with DSSS. The working of our proposed method is summarised as follows and is presented graphically in figure 18:

- The network starts communication as per the specifications. For the case of IEEE 802.15.4 devices will first search all available frequencies for the PAN then will initiate an association request, followed by normal communication.
- Once the setup phase is completed, the PAN and end devices will continue utilizing the same frequency, however, end devices will continuously monitor their communication performance.
- If an end device detects that it access to the channel is blocked due to interference or when it senses that the channel is corrupted it will send an indication to the coordinator. This is the event.
- Upon reception of such an indication, the coordinator will notify the devices in the network about the new frequency and execute a channel hop.
- End devices receive the notification about the new frequency and retune their radio receivers to the new channel. Following that, normal communication is restored until another event is detected in which case, this cycle is repeated again.

As can be seen from the flow charts in figure 18, the proposed algorithm suggests a non periodic frequency agile approach. This is based on the assumption that good system performance can still be obtained if frequency hopping is preformed only when needed due to channel conditions. The real challenging question presented here is to define events upon which a frequency hop will be initiated. In this chapter we present two proposals that are based on the active period of the MAC layer namely the Superframe duration period. The first event makes use of CSMA clear channel assessment algorithm to check for failed channel access thus covering most of the SD period. The second procedure simply keeps track of the number of lost beacons during the MAC layer state machine operation thus covering the starting period when the channel is supposed to be clear for beacon transmission.
The modifications are added to the IEEE 802.15.4 SDL model described in the previous chapter. The introduction of the modifications will change multiple access collision avoidance mechanism described in IEEE 802.15.4 into a frequency agile interference and collision avoidance (FICA) mechanism.

![Figure 18: General concept of the FICA algorithm](image)

### 6.1 Modification of the CSMA/CA

Clear channel assessment indicates if the channel access is possible or not. The operation is a function of the CCA mode in use which according to the IEEE 802.15.4 specifications can be simple ED scan with . In case the channel is blocked by interference during this procedure, the MAC layer notifies the higher layer by sending a channel access failure signal. If that happens, We propose that a Boolean introduced in the MAC layer implementation be flipped to true thus indicating that the node may be experiencing some interference issues as per the CSMA/CA algorithm assessment. This new variable will be called the CSMA-FICA bit. The modified CSMA algorithm is described in Figure 19 below:
Figure 19: The modified CSMA

6.2 Beacon Lost Indication

During beacon transmission, it is interesting to note that all other end devices are waiting and expecting a beacon. Thus during that small period of time, the quality of the received beacon frame can tell a lot about the channel conditions.
Unfortunately, the link quality indicator is not only a function of channel quality but is also a function of the transmission power, thus, the coordinator’s battery is low, the LQI will indicate weak transmissions. If the beacon is lost however, it may be because of multipath effects, collusion with other networks traffic or interference.

According to the IEEE 802.15.4 specifications, the number of beacons lost should not exceed a maximum limit otherwise the higher layer will be signalled to indicate loss of synchronization with the coordinator [1]. Thus the number of lost beacons is recorded and compared to the maximum allowed at the beginning of each beacon interval. To utilize this property, we suggest introducing another Boolean variable that can be called in this case BL-FICA bit. This bit can be flipped to true once the number of beacons lost exceeds a predefined limit thus indicating interference issues from a beacon reception perspective. This procedure is described in the Figure 20:

![Figure 20: BL/FICA algorithm](image-url)
6.3 Coordinator notification

The BL-FICA and the CSMA-FICA bits can both be used independently or combined by the node to reach a decision on informing the coordinator of possible interference issues within the channel. The advantage of using a combination of both events is that the node will only report persistent interference conditions and will discard temporary interference issues. For the purpose of this work, a positive decision will be made to report interference to the coordinator if BL-FICA and CSMA-FICA are both true at the same time.

In order to notify the coordinator, a flag bit can be introduced in the data frame format. This will be called the FICA flag and it will be set to true if the end device wishes to indicate interference issues to the coordinator. Thus, the frame structure in the SDL model can be modified as shown in figure 21:

![Figure 21: Modified frame structure with the FICA bit introduced](image)

It is interesting to note that by following this approach, the task of the sensor network is expanded from just reporting the event it was designed to monitor to include reporting the condition of the channel. The descriptions imply that following the suggested frequency agile algorithm will cause the entire network to change frequency in the face of interference.

6.4 Channel Hopping

The above procedures described one side of the story. The other side describes what happens at the coordinator end. The general proposed algorithm is quite a comprehensive in describing the operations taking place at the coordinator’s side. Once the coordinator receives a frame with a FICA flag set to true, it will initiate a frequency hop by first notifying the end devices. The notification process involves setting a FICA bit that can be introduced into the beacon frame in a fashion similar to that described in figure 21 of the data frame case.
Once the channel hopping notification is sent out the coordinator it continue operation into the next state of the MAC layer. And by the end of the CFP period it will change frequency and go to idle mode awaiting the next Superframe. The end devices will receive the notification, finish the MAC layer operating cycle and then update their channels. At the end of the Idle state, they will be waiting for the beacon’s arrival on the new channel. In terms of the State diagram of the MAC layer this means the addition of a new state that we call channel SET in which the conditions for frequency hopping will be examined and the hardware tuned to the new channel. The modified MAC state diagram is shown in Figure 22 below:

Figure 22: Modified MAC Layer
Communicating new channel information to end devices be done via utilizing the beacon payload. However, there is a possibility that the beacon will not be received and in this case, the end device will lose synchronization to the coordinator, thus making this option quite unreliable.

Different solutions to this problem were investigated via experimenting with different beacon lost solutions. The experiments were run using micaz motes running special application programs that implement each of the proposed solutions. The results of the experiments are summarized in table 3.

<table>
<thead>
<tr>
<th>Solution</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>End device has no knowledge of the hopping pattern and will try to relocate the coordinator.</td>
<td>Dynamic relocation of the coordinator</td>
<td>Large search time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the hopping interval is small the end device will fail to locate or resynchronize to the coordinator and the system will fail.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If the hopping interval is small and proper synchronization is not implemented the end device will fail to locate or resynchronize to the coordinator and the system will fail.</td>
</tr>
<tr>
<td></td>
<td>Quite reliable provided that the end device had enough time to locate the coordinator after every hop.</td>
<td>New channel must be statically specified.</td>
</tr>
</tbody>
</table>

*Table 3: Synchronization loss solutions*

The results of the experiments indicate that using a predefined hopping pattern is quite beneficial in avoiding many of the synchronization issues that may arise when frequency hopping is utilized. Furthermore, the experiments revealed that it is important for a coordinator to maintain its new frequency until all associated end devices are resynchronized on the new frequency. Such observation should be taken into consideration once the implementation is attempted.
7 Verification and Software Implementation

For verification, a small application process was also implemented in SDL. The purpose is to facilitate simulations and to log results. It sends the MAC_START_REQUEST primitive to the MAC layer by which it defines the device as a coordinator or an end device. Then it keeps track of the number of sent frames, the number of received frames, the number of received beacons, and the number of attempted retransmissions. In case of an end device, the device will wait for a beacon to be received then it will send a single data frame back to the coordinator during every Superframe duration. The coordinator on the other hand will just be sending beacons periodically to end devices.

The application process was used with both the plan high level implementation of the IEEE 802.15.4 and with both proposed event based algorithms. The results are described in the following sections. The SDL model comprises of an adjustable number of End nodes and one coordinator node connected together through a channel model and is shown in Figure 23 below:

![Figure 23: Verification system diagram](image-url)
### 7.1 IEEE 802.15.4 Plane Implementation

The SDL system was simulated using a coordinator node communicating with five other end devices. The network was established using a star topology first using a channel that is continuously operating in the low interference state, following that, the test was repeated while the channel is continuously in the high interference state. The SDL was conducted for a period of 50 Beacon Intervals. The results are displayed in table 4 and 5 respectively.

The results of the simulation in the normal case reflect the effects interference may have on devices operating in wireless environments. As can be seen, the number of successfully sent frames is lower when devices are operating in high interference environments while using CSMA-CA algorithm.

The results are consistent with the experimental results given in [ ] where the authors concluded that devices operating without CSMA have a higher throughput than those with CSMA because the CSMA algorithm may cause a transmission back off because it sensed the channel busy as a result of slight interference that may not really affect the transmission.

<table>
<thead>
<tr>
<th>Device type</th>
<th>Beacons Lost</th>
<th>Successful Data Sent</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinator</td>
<td>-</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>End 1</td>
<td>0</td>
<td>48</td>
<td>-</td>
</tr>
<tr>
<td>End 2</td>
<td>0</td>
<td>45</td>
<td>-</td>
</tr>
<tr>
<td>End 3</td>
<td>0</td>
<td>46</td>
<td>-</td>
</tr>
<tr>
<td>End 4</td>
<td>0</td>
<td>45</td>
<td>-</td>
</tr>
<tr>
<td>End 5</td>
<td>0</td>
<td>44</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 4: Simulation results, normal implementation, low interference channel*
### Verification and Software Implementation

<table>
<thead>
<tr>
<th>Device type</th>
<th>Beacons Lost</th>
<th>Successful Sent</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinator</td>
<td></td>
<td>- 117</td>
<td></td>
</tr>
<tr>
<td>End 1</td>
<td>2</td>
<td>46</td>
<td>-</td>
</tr>
<tr>
<td>End 2</td>
<td>6</td>
<td>42</td>
<td>-</td>
</tr>
<tr>
<td>End 3</td>
<td>4</td>
<td>37</td>
<td>-</td>
</tr>
<tr>
<td>End 4</td>
<td>8</td>
<td>39</td>
<td>-</td>
</tr>
<tr>
<td>End 5</td>
<td>7</td>
<td>41</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 5: Normal implementation results at High interference channel*

#### 7.2 FICA Algorithm

The same simulation setup structure was used, this time with the modified version of the implemented IEEE 802.15.4 to account for the proposed modifications. This time, multiple channels were used starting with one at high interference state. The others were at low interference, but can be switched to a higher state. The results are displayed in the following tables 6.

<table>
<thead>
<tr>
<th>Device type</th>
<th>Beacons Lost</th>
<th>Successful Sent</th>
<th>Received</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinator</td>
<td></td>
<td>- 219</td>
<td></td>
</tr>
<tr>
<td>End 1</td>
<td>0</td>
<td>43</td>
<td>-</td>
</tr>
<tr>
<td>End 2</td>
<td>2</td>
<td>43</td>
<td>-</td>
</tr>
<tr>
<td>End 3</td>
<td>0</td>
<td>43</td>
<td>-</td>
</tr>
<tr>
<td>End 4</td>
<td>0</td>
<td>47</td>
<td>-</td>
</tr>
<tr>
<td>End 5</td>
<td>1</td>
<td>46</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 6: FICA algorithm verification*
As can be seen from the simulation results, the performance of the system improved as compared to single channel implementation. The number of beacons lost is reduced and so is the number while the number of successfully sent and received frames improved as compared to the numbers achieved with a high interference channel. A graphical comparison of results is presented in Figure 24 and Figure 25 respectively.

Figure 24: Data frames received comparison

Figure 25: Beacon Lost comparison
7.3 Software Implementation

Following the development, evaluation and simulation work, the final step is to implement the developed algorithm on nesC over an existing IEEE 802.15.4 stack. After research, it was decided to adapt the IEEE 802.15.4 stack implemented by the open Zigbee organization [34]. Their released code is described in [35]. In this section we describe the mapping from SDL to nesC and the resulting code. While the FICA bit can simply be added to the MAC layer implementation as a new variable of Boolean, Following our study of the IEEE 802.15.4 frame structure, it was decided to implement the FICA flag as part of the reserve bits in the frame control field.

Originally, in the used implementation, the frame control field is populated using the set_frame_control function in which all reserved bits are ignored. To access one of the reserved bits, the function was modified to account for the newly defined FICA flag. Both the original implementation and the modification are given as follows:

```c
// original Implementation

uint16_t set_frame_control(uint8_t frame_type,uint8_t security,uint8_t frame_pending,uint8_t ack_request,uint8_t intra_pan,uint8_t dest_addr_mode,uint8_t source_addr_mode)
{
    uint8_t fc_b1=0;
    uint8_t fc_b2=0;
    fc_b1 = ( (intra_pan << 6) |(ack_request << 5) |
            (frame_pending << 4) |(security << 3) |
            (frame_type << 0) );
    fc_b2 = ( (source_addr_mode << 6) | (dest_addr_mode << 2)));
    return ( (fc_b2 << 8 ) | (fc_b1 << 0) );
}
```
The function set_frame_control() helps to write the value of the FICA flag, reading the value of the FICA flag was not addressed in any of the implemented functions simply because the bit was ignored in the implementation. To facilitate reading this new flag bit a new function, (get_fcl_fica) which is similar to the other functions used to read the other bits in the frame control field was introduced, the syntax is as follows:

```c
// FICA flag Added
// Modified by Khalid Obaid

uint16_t set_frame_control(uint8_t frame_type,uint8_t security,uint8_t frame_pending,uint8_t ack_request,uint8_t intra_pan,uint8_t fica,uint8_t dest_addr_mode,uint8_t source_addr_mode)
{
    uint8_t fc_b1=0;
    uint8_t fc_b2=0;

    fc_b1 = ( (fica << 7) | (intra_pan << 6) |
            (ack_request << 5) | (frame_pending << 4) |
            (security << 3) | (frame_type << 0) );

    fc_b2 = ( (source_addr_mode << 6) | (dest_addr_mode << 2)));

    return ( (fc_b2 << 8 ) | (fc_b1 << 0) );
}
```

For the rest of the software implementation, the two functions were used to implement the same logic that was presented in the Model implementation chapter by modifying the relevant parts of the MAC layer implementation. Testing for the FICA flag and setting the FICA bit whenever needed.
8 Conclusions and Recommendations

The purpose of this thesis was to introduce a new frequency agile MAC based scheme for the IEEE 802.15.4 devices which, unlike the existing frequency hopping schemes, is “Event Based”, the thesis main contribution came in the development of the Frequency agile interference and collusion avoidance concept (FICA). The system level simulations results conducted during the development process revealed an evident performance improvement when the system utilized more than a single channel while the performance of both proposed FICA methods indicated excellent potential for interference avoidance.

The results of the high abstraction simulations support our initial assumption that periodically changing frequency is not necessary to achieve good wireless performance and channel hopping can be performed when necessary as dictated by channel conditions. Actual implementation was carried out using TinyOS and nesC and the code was compiled and implemented on Micaz motes.

8.1 Recommendations for future work

The validation method used in this work was based on SDL simulations to check if the whole system will work as intended. Further verification is possible using a well know simulation tool such as the network simulator 2 may further help locate some minor issues. Such simulations may help refine the proposed CSMA/FICA and Beacon Loss/FICA methods. It could be a nice idea to simulate the implementation in NS2 however, Experimental analysis is required, and is preferable, to completely verify the performance of the pressed MAC layer modifications. Preferably, the experiments should be designed to check the industrial environment in order to foresee all possible issues that may arise when implementing industry specific solutions.

It is possible for the coordinator to use the CSMA/FICA algorithm and assess channel conditions around its location. In this case the coordinator will follow the same steps described in Figure 19 except that the coordinator will, in this case directly indicate a channel hop to the rest of the devices using the same procedure described earlier. This possibility combined with GTS traffic to replace the use of beacon can be investigated to arrive at a comprehensive channel assessment at the coordinator side.
9 References


References


CCK : Solutions to avoid mutual interference », IEEE P802.11 Working Group Contribution, IEEE P802.11- 00/162r0, July 2000.


[33] TinyOS website: http://www.tinyos.net/

[34] Open ZB website: http://www.open-zb.net/

[35] IP Hurray IEEE 802.15.4 implementation documentation.