ASHESI UNIVERSITY COLLEGE

EVALUATING INTERFERENCE IN BLUETOOTH CLASSIC AND BLUETOOTH LOW ENERGY TECHNOLOGY

APPLIED PROJECT

B.Sc. Computer Science

Jeffrey Takyi-Yeboah

2017

ASHESI UNIVERSITY COLLEGE

Evaluating Interference in Bluetooth Classic and Bluetooth Low Energy Technology

APPLIED PROJECT

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science

Jeffrey Takyi-Yeboah

April 2017

DECLARATION

I hereby declare that this Applied Project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……………………………………………………………………………………………

Candidate's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

I hereby declare that preparation and presentation of this [capstone type] were supervised in accordance with the guidelines on supervision of [capstone type] laid down by Ashesi University College.

Supervisor's Signature:

……………………………………………………………………………………………

Supervisor's Name:

……………………………………………………………………………………………

Date:

……………………………………………………………………………………………

Acknowledgement

To my supervisor, Dr. Nathan Amanquah, I am grateful for your guidance, inspiration, and encouragement throughout the project timeline. I am thankful for your intellectual advice and time spent in helping me understand new concepts to make this project a success.

Abstract

Wireless communication is one of the dominant discoveries to mankind. Bluetooth, a wireless technology, is a crucial technology enabling wireless communications and more importantly, the internet of things. A recognized challenge in Bluetooth communication is the overlap of frequencies for specific time slots causing decreased throughput and delay of signals. This is described as Bluetooth interference. In order to improve communications in the internet of things, there is the need to evaluate and improve the Bluetooth technology. In this project, Bluetooth interference is evaluated using simulation. The two technologies involved are Bluetooth Classic and Bluetooth Low Energy. Simulation results are to help make informed decisions and recommendations in the usage of Bluetooth devices in coexistence. Findings in this study, confirm the presence of interference in the two technologies, and their relationship with the number of communicating pairs. Moreover, the incidence of interference is found to be relatively high in the Bluetooth Low Energy technology.

Table of Contents

Chapter 1: Introduction

1.1 Introduction

Wireless communication is one of the dominant discoveries to mankind. It is responsible for sharing data in ways that make them readily available and useful for our human endeavours. It is part of the norm of doing better with technology and as the years go by, new trends and ground-breaking technologies are being discovered to make our lives better. A noticeable trend is the internet of things, a concept that wireless computing devices will be connected to the internet and embedded in everyday objects to send and receive data, that will be useful in our daily lives. The future of the internet of things will be massive, as recent research has shown that more than 26 billion devices will be connected to the internet of things by 2020 (Gartner, 2013). This trend has been triggered by increasing developments in wireless communications, real-time analytics, machine learning and embedded systems.

According to the Bluetooth Special Interest Group (2014), a leading technology for connecting wireless devices in the internet of things is the Bluetooth technology. The technology is discussed in two types; the classic technology and the low energy technology. Additionally, Bluetooth is described as a standard short-range wireless interface, operating in the unlicensed 2.4GHz Industrial, scientific and medical (ISM) band. It is aimed at simplifying communications among wireless devices (Bluetooth SIG, 2014). The ISM band is a part of the radio spectrum reserved internationally for the use of radio frequency(RF) for industrial, scientific and medical purposes rather than telecommunication (Bluetooth SIG, 2014). Bluetooth uses a frequency-hopping spread spectrum mechanism, where it selects frequencies for specific times in the 2.4GHz unlicensed band to send and receive data. However, the incidence of collisions is likely to increase, as more and more devices are collocated. The collision will be as a result of two or more Bluetooth devices using the same frequency at the same time to send or receive data. The effects of collisions in wireless communications include decreased throughput, delay and loss of data.

This project seeks to evaluate and quantify how much of a problem Bluetooth devices face when collocated. The likely problem, also described as interference, if significant, will help make informed decisions on the limit of the number of collocated devices in Bluetooth communication.

1.2 Problem/Motivation

The growing conversation about Internet of things speculates that in the near future, Bluetooth wireless interfaces will be the standard for communication. A problem is likely to arise when Internet of things devices using Bluetooth, communicate while they are collocated. In Bluetooth communication, this problem is described as interference, where for certain time slots, devices hop on to the same frequencies generated by the frequency hop selection kernel. This in effect causes packet collisions. The frequency hop selection kernel is responsible for generating a set of frequencies for Bluetooth devices to send and receive data, given a set of parameters.

In such a world where internet of things is aimed at making lives better, the disadvantages of interference cannot be ignored. They pose a significant threat to connections and if not curbed, can adversely affect wireless communications in the internet of things. Some of the challenges interference pose to the internet of things for the Bluetooth interface, include slower connections in file transfer, inability to pair Bluetooth devices and poor performance. In order to avoid this, there is the need to evaluate the current Bluetooth technology responsible for connecting wireless devices. The project will be aimed at finding how much interference occurs when there are so many collocated Bluetooth devices. This is important because, in the foreseeable future, a lot more Bluetooth devices will be connected to the internet of things.

1.3 Project Objectives

The focus of this project is to quantify and evaluate interference in the Bluetooth technology: Bluetooth Classic and Bluetooth Low Energy. This is ultimately targeted at making better decisions that will improve the work of the technology in wireless communications.

The objectives of the project include:

a. Implementation of the frequency hop selection kernel for Bluetooth Classic in C++. This kernel is used to generate frequencies for communication in Bluetooth classic devices.
b. Implementation of the data channel selection algorithm for Bluetooth Low Energy in C++. This algorithm is responsible for the selection of data channels for communication in Bluetooth Low energy devices.
c. Simulation of a typical connection for Bluetooth classic and Bluetooth low energy.
d. Evaluation of interference in Bluetooth Classic and Bluetooth Low Energy.

1.4 Overview of Paper

This paper consists of five chapters. The first chapter introduces readers to the project and its relevance especially to the internet of things. Chapter 2 discusses the work of the Bluetooth technology into detail, highlighting existing work and research surrounding interference in Bluetooth and other wireless technologies. It also gives an overview of possible simulators for implementation. Chapter 3 describes the methodology of the project emphasizing the specific procedure and processes required to evaluate and quantify interference in Bluetooth. In Chapter 4, results from simulation and implementation are discussed and analysed for insights. An evaluation based on results is done to validate the research question and problem under scrutiny. Chapter 5 presents a conclusion based on results, with regards to the work of the Bluetooth and its relevance to the internet of things. Further research into the project is suggested, as limitations of the Bluetooth system are highlighted.

Chapter 2: Literature Review, Background & Related Work

This chapter provides an overview of the Bluetooth system, its types and its operation procedures in sections 2.1 to section 2.3. The next section, 2.4 discusses the relevance of the Bluetooth low energy system in the internet of things. Section 2.5 compares and contrasts the two Bluetooth systems: Bluetooth Classic and Bluetooth Low Energy. In section 2.6, the concept of interference in Bluetooth and other wireless networks is explained. The final section provides an overview of simulators and their various advantages and disadvantages.

2.1 Bluetooth Baseband Specification and Description

Bluetooth is a wireless short range communication systems intended to simplify communications among devices. The significant features of the Bluetooth wireless technology are low cost, low power consumption and robustness. Some important applications areas in Bluetooth include synchronous(streaming), asynchronous (file transfer) and state (small amounts of data sent infrequently) connections.

Other Bluetooth technologies such as the common Bluetooth synchronous connection-oriented define a synchronous system with a master and slave transmission on 1600HZ intervals. The Bluetooth asynchronous connectionless also uses the master-slave system and has a lower duty cycle depending on the traffic needs (Decuir, 2014). A connectionless service is one that does not require a session connection between sender and receiver. A master Bluetooth device is one responsible for initiating a connection between Bluetooth devices. It also generates frequencies for the slave devices to hop on. A slave device, on the other hand, listens to the requests of the master and can only transmit or receive from their master device.

The Bluetooth systems discussed in this project are Bluetooth Classic and Bluetooth Low energy technologies. The Bluetooth system provides: point to point connection and multipoint connections. The point to point connection is the physical channel that is shared between two devices with one master and one slave (Bluetooth SIG, 2014). On the other hand, multipoint connection describes the physical channel shared among several Bluetooth devices.

The Bluetooth baseband is the physical layer of the Bluetooth which manages physical channels and physical links responsible for services such as error connection, hop selection and security. It lies on top of the Bluetooth radio layer in the Bluetooth stack (Bluetooth SIG, 2014). Within the physical channel of the Bluetooth, two or more devices sharing the same channel is called the piconet. A piconet is shown below in figure 2.1. On a piconet, a total of seven slaves can be active for the Bluetooth classic technology whereas an unlimited number of slaves can be active for the Bluetooth Low Energy technology. Slaves can remain in a parked state; not active on the channel and can remain synchronized to the master. Each piconet has a master device responsible for generating its own set of frequencies, for slave devices to hop on. However, a master can be a slave in other piconets The time division multiplex scheme allows master and slave Bluetooth devices to transmit alternatively (Bluetooth SIG, 2014).

A group of piconets that have common Bluetooth devices being represented as master and slaves is described as a Scatternet. A Scatternet as shown in figure 2.1, allows multiple communication between Bluetooth devices connected to each other.

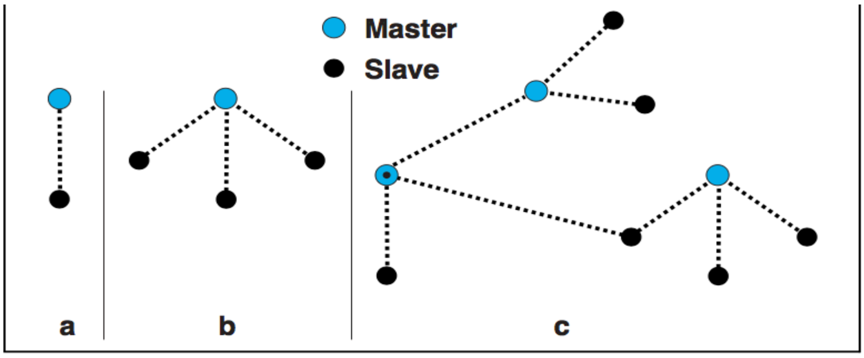


Figure 2.1: Illustration of Piconet and Scatternet (Bluetooth SIG, 2014).

a. Piconet with one slave **b**. Piconet with multiple slaves **c**. Scatternet

2.2 Bluetooth Classic

2.2.1 Physical Channel of Bluetooth Classic

Bluetooth classic is an older version of Bluetooth developed by the Bluetooth Special Interest Group for short-range wireless communications. It operates in the 2.4GHz unlicensed ISM band. The physical channel of the Bluetooth technology is described as the lowest architectural layer of the Bluetooth system. There are a number of physical channels defined and the way they are used differ in each type of Bluetooth system. They are defined by a pseudo random radio frequency (RF) channel hopping sequence, the packet time and an access code.

During communication, devices share the physical channel. For this to work, the transceivers have to be tuned to the same frequency at the same time and be within a nominal range. Independent devices sharing this physical channel are likely to cause collisions due to the limited number of radio frequency (RF) carriers. The Bluetooth device can use only one channel at a time. To ensure simultaneous operations in multiple piconets, the devices use a time division multiplexing between channels.

The Bluetooth classic has four distinct physical channels defined. Two of which, are responsible for communications between connected devices. They are the basic and adapted piconet channel. The other two channels: inquiry scan and page scan channel, are used for discovery and connecting devices respectively.

2.2.2 Bluetooth Classic Operation

The Bluetooth classic version employs a frequency hopping transceiver to reduce interference (Bluetooth SIG, 2014). It has many Frequency hopping spread spectrum (FHSs) carriers. The symbol rate for the basic rate is 1 mega symbol per second supporting a bit rate of 1 megabit per second. For the enhanced data rate, the gross air bit rate is 2 or 3 megabit per second (Bluetooth SIG, 2014). A typical Bluetooth classic operation is detailed below;

a. During operation, a physical radio channel is shared by a group of devices synchronized to a frequency hopping pattern and a common clock signal (Bluetooth SIG, 2014). Master device provides synchronization reference to other devices. The devices synchronized to the master's clock and frequency hopping pattern are known as slaves.
b. Devices have a specific hopping pattern which is determined by a mathematic operation that takes the Bluetooth address and clock signal of the master as parameters. The hopping pattern is a pseudo-random ordering of 79 frequency channels in the ISM band. The adaptive hopping technique is adopted, where frequencies being used by interfering devices are excluded, to help improve Bluetooth coexistence when devices are collocated.
c. The physical channel is divided into time units known as slots. In these slots, packets are transmitted at the frequency determined by the master devices.

2.2.3 Frequency Hop Selection in Bluetooth Classic

This section describes the technique and processes used by the Bluetooth Classic technology to select frequencies for the transmissions and reception of packets. The Bluetooth classic uses 79 frequency channels with 1 MHz spacing in the 2.4 GHz unlicensed ISM band. (Bluetooth SIG, 2014).

In Hop selection, the frequency spread spectrum technique is used. The frequency hop selection kernel determines the frequency a Bluetooth classic device will use in transmission. The technique uses the Bluetooth address and clock signal of the master device as input parameters. The master node is responsible for initiating the connection. However, the Bluetooth address determines the hopping sequence whereas the clock signal determines the phase in the sequence (Bluetooth SIG, 2014).

During inquiry process, where a series of requests for connections are initiated, the master's address and clock is transmitted to slaves (Bluetooth SIG, 2014). The slave then calculates the offset between its clock and the master's. The offset used together with the master's address, allows the slave to synchronize its hop sequence with the master (Bluetooth SIG, 2014). The selection is done in a pseudorandom manner, where each piconet has a unique hopping sequence determined by master device's parameters. Interference occurs for some time slots in this pseudorandom order where frequencies match up and overlap causing packet collisions and consequently decreased throughput in these piconets.

2.3 Bluetooth Low Energy

2.3.1 Physical Channel of Bluetooth Low Energy

The core system of the Bluetooth low energy operates in a similar way as the Bluetooth classic. The Bluetooth devices share a physical channel used for communication with their transceivers tuned on the same frequency at the same time. The incidence of physical channel collision is high as the number of physical channels is limited with many devices also operating independently on the same frequency channel. To reduce collision, Bluetooth low energy generates a random access address to identify the physical link between devices (Bluetooth SIG, 2014). It essentially serves as a correlation to determine which device a communication is directed to. The Bluetooth device can use only one channel at a time. In situations of multiple concurrent operations, the devices use a time division multiplexing between channels in a way that the device appears to support connected devices while having other operations (Bluetooth SIG, 2014).

For the Bluetooth Low energy core system, there are two physical channels defined for operation. There is the Low energy piconet channel used for communication between connected devices in a specific piconet and the Low energy advertisement broadcast channel, used to broadcast advertisements to low energy devices. The advertisements include discovery, connection or sending user data to scanner or initiator devices.

2.3.2 Bluetooth Low Energy Operation

The low energy system employs a frequency hopping transceiver to minimize interference. It has many frequency-hopping spread spectrum (FHSS) carriers and the symbol rate is 1 mega symbol per second supporting a bit rate of 1 megabit per second.

It employs two multiple access schemes: The time division multiple access (TDMA) and the frequency division multiple access (FDMA). For the time division multiple access, devices transmit a packet at a predetermined time and have corresponding devices respond with a packet after the predetermined interval.

For the Frequency division multiple access scheme, 40 physical channels are used with 2 MHz spacing. Three of these channels are used for advertising whilst the other 37 channels are used as data channels. Data channels are those that can be hopped on for transmission and reception of packets. Within the physical channel, subdivided time units are known as events. Data is transmitted among the devices in packets and in the appropriate events. For the LE system, there are two types of events: advertising and connection events. The next subsection will describe the steps involved in advertising and connection events in the Bluetooth low energy technology.

2.3.2.1 Overview of Advertising

In Advertising, devices that transmit advertising packets on the same advertising channel are known as advertisers whereas devices that receive advertising packets but may have no intention of connecting to the advertising devices are known as scanners. Transmissions occur in advertising events. Detailed steps of the advertising event stage of the Bluetooth low energy technology are described below;

a. At the start of each advertising event, the advertiser sends out advertising packets on three distinct channels for scanners to respond or request to. The scanner may respond to the advertiser with a request to connect and this is followed by a response from the advertiser on the same advertising channel (Bluetooth SIG, 2014).
b. The advertising channel changes depending on the next advertising packet sent by the advertiser. The first advertising channel is used as the start of the next adverting event. Advertising events are useful in Bluetooth Low energy devices for unidirectional, broadcast connection and bidirectional communication between two or more devices using data channels (Bluetooth SIG, 2014).
c. Advertising events end when the initiator listens and receives connectable advertising packets using the same advertising channels. A request is made for the connection to be initiated. The connection events start when the request is accepted and confirmed by the advertiser. The initiator hence becomes the master in the piconet while the advertiser becomes the slave (Bluetooth SIG, 2014).

Below is a diagram to illustrate advertising events in Bluetooth low energy.

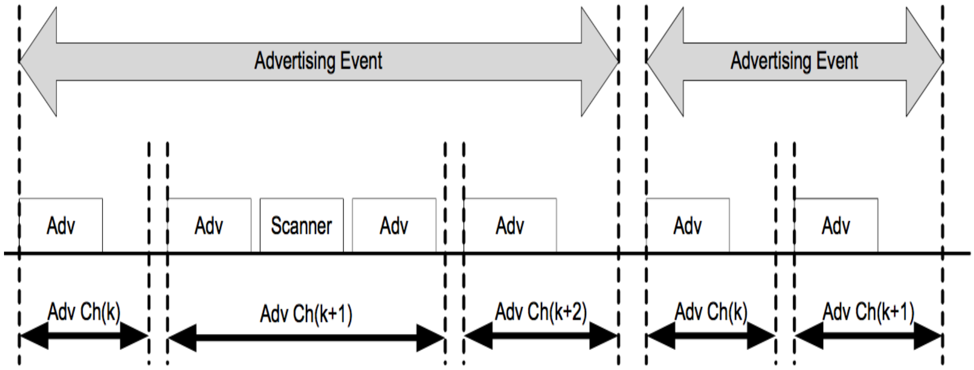


Figure 2.1: Advertising events in Bluetooth Low energy (Bluetooth SIG, 2014).

2.3.2.2 Overview of Connection Event

a. Connection event is used to send packets of data between master and slave. Channel hopping occurs at the start of a connection event (Bluetooth SIG, 2014).

b. Within the connection event, the master and slave alternate sending packets of data in the same data channel.

c. The initiator/master starts and ends the connection event.

d. For a piconet, the frequency hopping pattern is specific and generated by a field contained in the request sent by the initiator. The field contains parameters such as the channel map and hop-increment, these are used in data channel selection. The initiator also provides synchronization reference which is known as the hop interval. The frequency hopping pattern is a pseudorandom ordering of 37 frequencies in the ISM band.

Below is a diagram to illustrate connection events in Bluetooth low energy.

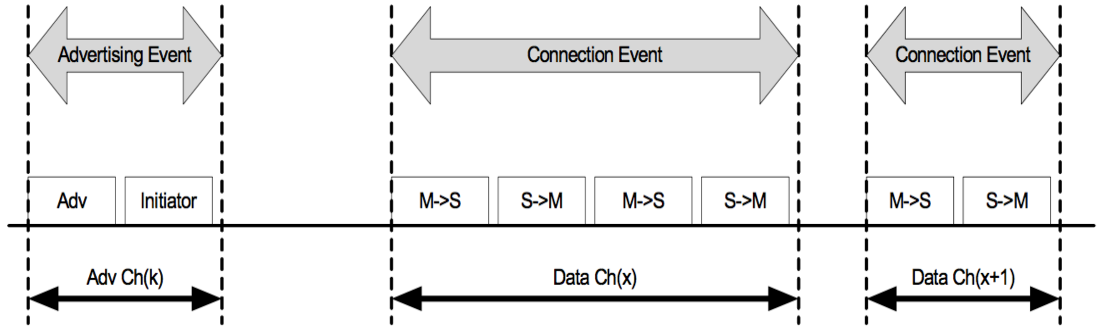


Figure 2.2: Connection events in Bluetooth Low energy (Bluetooth SIG, 2014).

2.3.3 Data Channel Selection in Bluetooth Low Energy

Bluetooth Low energy uses 40 frequency channels with 2 MHz spacing in the 2.4 GHz unlicensed ISM band. (Bluetooth SIG, 2014). 3 of these channels are set aside for device discovery and avoidance of Wi-Fi signals in the 2.4GHz ISM band. 37 of these frequency channels are used for frequency hopping or data channel selection for transmission and reception of packets.

A master Bluetooth device sends a CONNECT_REQ message which contains a channel map(ChM) field and a hop increment. The ChM contains 40 bits with 3 bits reserved for advertising. 37 bits labelled from 0 to 36, represent data channels (Kalaa & Refai, 2014). In the Bluetooth low energy channel selection, bits with value ‘1’ are used data channels whereas those with ’0’ are unused data channels, mostly reserved for advertising and discovery of Bluetooth devices (Kalaa & Refai, 2014). The minimum number of used channels for a channel map is 2.

Used data channels are those that can be used for transmission and reception of packets in BLE communication whereas unused data channels cannot be used for packet transmission. They are reserved for advertising and discovery purposes. The hop increment parameter is a random value between 5 and 16 generated to simplify calculations for the hopping frequency in the Bluetooth low energy technology. Upon successful delivery of the CONNECT_REQ message, the master and slave Bluetooth devices have the parameters needed to determine the next data channel index for devices to hop on. Interference also occurs in the data channel selection process, where two or more devices hop on to the same data channel for the same time slot. This subsequently causes packet collisions which adversely affects throughput in wireless communications.

2.4 Bluetooth Low Energy and Internet of Things

The nature of data transactions in Bluetooth communication is relevant to the internet of things(IOT). The way packets are transmitted from one Bluetooth device to the other is important because, when a lot more Bluetooth devices are collocated, there is the likelihood of transmission collisions. Bluetooth Low Energy is a leading technology in connecting the internet of things and this explains why the topic of improving the technology is essential for this project.

The improvements place emphasis on efficiency, where data transmissions are done in less time for even less power as compared to the Bluetooth Classic technology. Currently, Bluetooth Low Energy is embedded in most devices and operate slightly different from the Bluetooth Classic. Considering the fact that, most devices being produced are currently being powered by Bluetooth Low Energy, it is expected to contribute massively to the internet of things. The improvements in Bluetooth low energy are aimed at improving usability, providing a flexible model for developers and laying the foundation for the internet of things, where more work is being done on assigning IP addresses to Bluetooth devices (Decuir, 2014).

The new Bluetooth low energy technology is energy efficient, meaning it can do more work with less energy. This makes it more suitable to power wireless devices that are being connected to the internet for constant data transmissions.

The features that make the Bluetooth smart/ Bluetooth low energy essential for internet of things are discussed below;

- New advertising mechanism for efficient device and discovery
- New asynchronous connectionless media access control (MAC) for low latency and fast transactions
- New generic attribute protocol (ATT) which in turn supports a simple client/server model
- New generic attribute profile provides an efficient way of collecting data from servers (Decuir, 2014).

The ubiquitous nature of Bluetooth Low Energy together with its unique features, describe how vital the technology is, in connecting the internet of things especially with such improvement in battery power, transmission and coverage (Decuir, 2014).

2.5 Comparing Bluetooth Classic to Bluetooth Low Energy

In this section, the two Bluetooth technologies: basic rate/enhanced data rate also called Bluetooth classic and the Low Energy technology also called Bluetooth smart, are compared and contrasted.

Table 2.1: High-level and low-level comparison between the Bluetooth classic (Bluetooth BR/EDR) and Bluetooth low energy technologies (Bluetooth SIG, 2014).

Specifications	**Bluetooth Classic**	**Bluetooth Low Energy**
Radio Frequency	2.4 GHz	2.4 GHz
Parameters for Frequency Selection	Bluetooth Address and clock signal	Channel Map and Hop Increment
Frequency Channels and Equation Specification	79 channels with 1 MHz spacing from 2400-2483.5 MHz **f = 2402 + k MHz where k = 0..78**	40 channels (Includes 3 advertising and 37 data channels) with 2 MHz spacing from 2402 MHz to 2480 MHz **f = 2402 + 2.k MHz where k = 0..78**
Nodes/Active Slaves	7	Unlimited
Physical Channels	Adapted piconet, basic piconet, page scan & inquiry scan channels	Low energy piconet & advertisement broadcast channels
Symbol Rate	1.3 Mbps	1 Mbps
Network Topology	Point-to-point, Scatternet	Point-to-point, star
Distance/Range	100 m (30ft)	50 meters (160ft)
Application Throughput	0.7 – 2.1 Mbps	Less than 0.3 Mbps
Power consumption	Low (less than 30 mA)	Very low (less than 15 mA)
Robustness	Frequency Hop Spread Spectrum (FHSS)	Frequency Hop Spread Spectrum (FHSS)
Speed	700 Kbps	1 Mbps
Primary use cases	Mobile phones, headsets, stereo, audio, automotive, PCs etc.	Mobile Phones, gaming, PCs, sports & fitness, medical automotive, industrial, automation, home electronics etc.

2.6 Interference

Interference is described in communications as anything which disrupts the correct reception of a signal at a receiver. It is also described in wireless communications as a situation in which multiple transmissions occur simultaneously over a common communication network at the same time and frequency. Over the years, a primary challenge in developing more efficient wireless communication systems has been interference, because it limits reliability and throughput of a system. (Yan Xin et al 2010). It also causes significant performance degradation when devices are co-located in the same environment (Golmie, N et al 2001).

The common types of interference include electromagnetic interference, co-channel interference, adjacent-channel interference, inter-symbol interference, inter-carrier interference, common mode and conducted interference. The increasing use of wireless networks raises the need for concern as more nodes or devices are constantly being added to the network during communication. Popular standards that make communication among these wireless devices possible include the IEEE 802.11(Wi-Fi) and Bluetooth. Unfortunately, these devices operate in the same 2.4GHZ radio frequency in the ISM band.

2.6.1 Bluetooth Interference

In Bluetooth networks, interference is described as the overlap in frequencies or the situation where two or more Bluetooth devices hop onto the same frequency for the same time slot, causing packet collision. The occurrence of a collision is as a result of the operation of the frequency hop selection kernel in the Bluetooth classic technology and the data channel selection algorithm in the Bluetooth low energy technology. The frequency or data channel selection procedures are independently responsible for generating frequencies for Bluetooth devices in a network.

2.6.2 Mutual Interference between Collocated IEEE 802.11 devices

Mutual interference between collocated IEEE 802.11 devices is described below;

An IEEE 802.11 device tries to communicate or transmit some piece of data, it checks whether the channel has already been occupied (Mathew et al., 2009). The IEEE 802.11b issues “Clear to send” and the networks adapter sends the data. While another collocated IEEE 802.11b device is using the same channel, the transmission is postponed to cater for the first transmission (Mathew et al., 2009). This instance portrays mutual interference in the IEEE 802.11 networks.

2.6.3 Interference between Collocated IEEE 802.11 and Bluetooth

For collocated IEEE 802.11 and Bluetooth, the basis of interference arises when the two networks try sending data packets and collide with each other because they hop on to the same frequency at the same time (Mathew et al., 2009). The resulting interference causes the IEEE 802.11 station to retransmit signals, causing the signals to affect the transmission. As more devices use the 2.4GHZ ISM band, the situation gets worse and has to be addressed with a congestion-free network (Mathew et al., 2009). The synchronous connection-oriented with Repeated Transmission (SCORT) packet is suggested to reduce interference significantly (Mathew et al., 2009).

Figure 2.3 below, shows the different technologies such as microwave transmission, Wi-Fi and Bluetooth transmission occupying a range of frequencies in the 2.4Ghz Industrial, Scientific and Medical(ISM) radio band. This explains the high number of radio technologies relying on the limited set of frequencies in the 2.4 GHz ISM band. The overlap in frequencies between 802.11b (Wi-Fi) signals and Bluetooth illustrates that if these different technologies are operated at the same time, there will be interference.

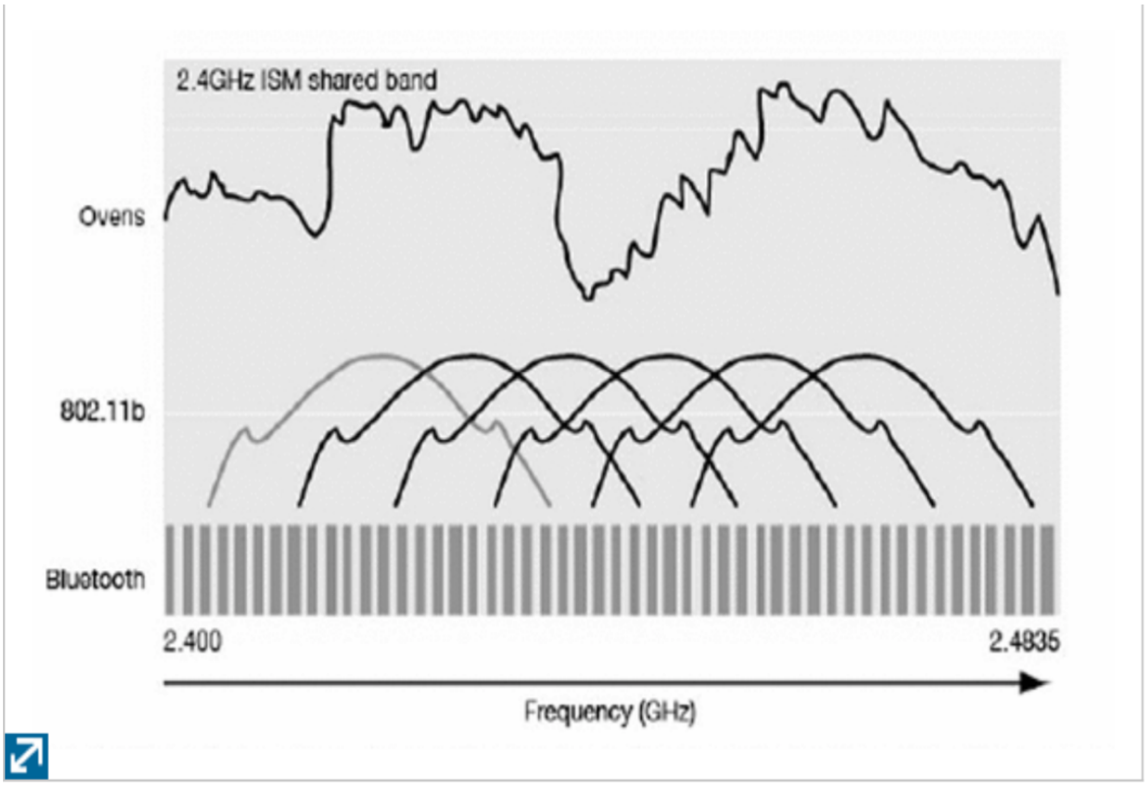
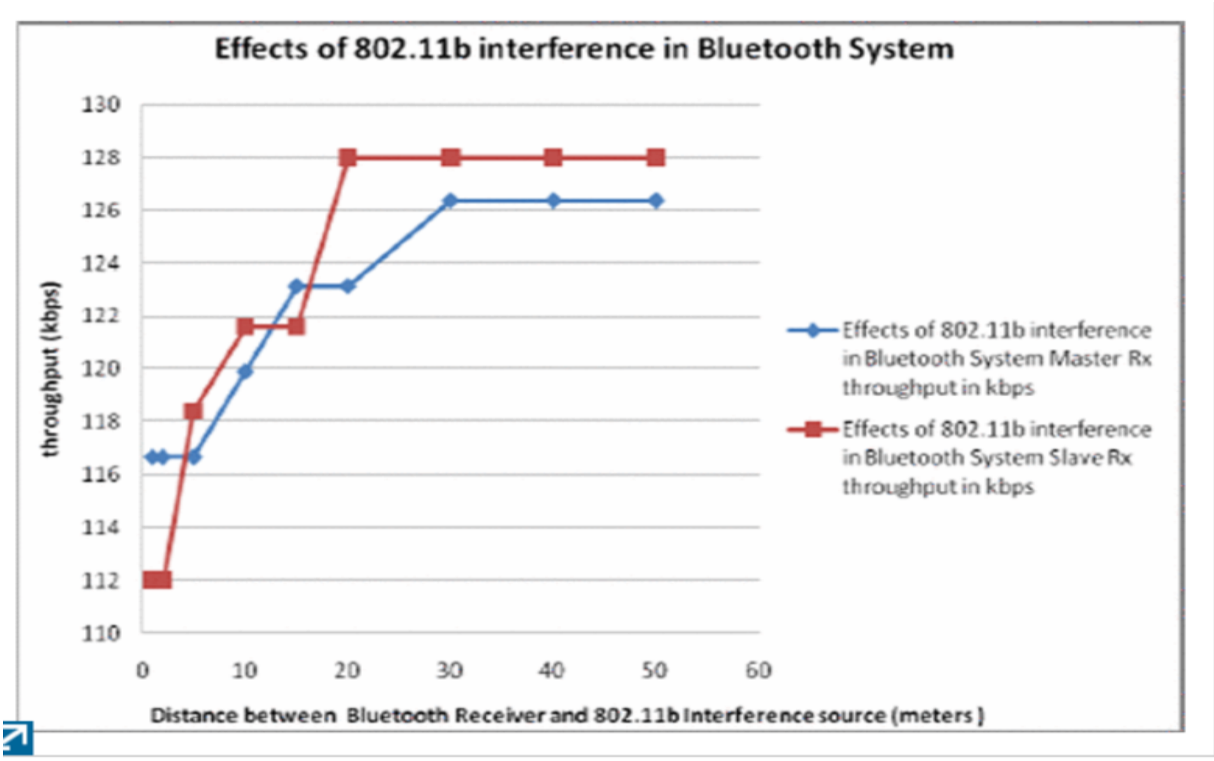


Figure 2.3: The figure describes the 2.4Ghz Industrial, Scientific and Medical(ISM) spectrum being occupied by IEEE 802.11 and Bluetooth (Mathew et al., 2009).

In figure 2.4 below, the effect of interference in the Bluetooth system (throughput on the master and slave system) is illustrated. The figure indicates that a high interference adversely affects throughput in the Bluetooth system.



igure 13 Bluetooth Master and Slave device throughput in the presence of 802.11b

Figure 2.4: The figure describes the effects of interference in the Bluetooth System (Mathew et al., 2009).

2.7 Overview of Simulators

A simulator is described as an invaluable tool for performance evaluation of algorithms and protocols in wireless networks (Imran et al., 2010). Simulators offer an option to modelling and predicting the behaviour of the real environment in different scenarios. The main reasons behind the use of simulators include low cost, efficient use of time and easy implementation (Imran et al., 2010). Network Simulators are categorized into general purpose and wireless sensor network specific simulators. Some of the common types of the general purpose simulators include OMNET++, NS-2, J-SIM GloMoSimlQualNet, Ptolemy II MAT LAB, JIST/SWAN, OPNET, NS2, WIPSIM, SSFNET etc. (Imran, M et al 2010). Some wireless sensor network specific simulators include: sensorSim, Nsrlsensorsim, Castalia, Visual Sense, Viptos, Sidh, Prowler etc. For this project, the focus will be on the general purpose simulators.

The following describe the pros and cons of a number of simulators and how they influence the choice of a simulator for this project.

2.7.1 Pros and Cons of Selected Simulators

2.7.1.1 OPNET

OPNET runs on top of a C compiler and provides a high number of editors for creating, modifying or verifying models as well as simulations and displaying results (Kuijpers & Telebit, 2003). OPNET provides a nice interface, has a large customer base, professional support, extensive documentation and a large number of built-in protocols necessary for simulation. However, OPNET has a relatively high price and complicated structure. It is also time-consuming to learn.

2.7.1.2 NS2

The NS2 is a discrete event simulator which targets networking research. It is developed by Information Sciences Institute of the University of Southern California (Kuijpers & Telebit, 2003). It creates a model for network communications. For the NS2, it has many protocols implemented, making work easier for a developer. It is easily configured, fast and has a well-documented manual. It uses two different languages such as C++ and OTCL (Kuijpers & Telebit, 2003). The cons of the NS2 include: the difficulty in evaluating small ideas quickly, since the simulation structure has to be known (Kuijpers & Telebit, 2003). It has a relatively steep learning curve.

2.7.1.3 WIPSIM

WIPSIM has source code freely available and well documented. The design of the general framework makes it easy to add new protocols. Some of the disadvantages of using the WIPSIM are: limited number of implemented protocols, slow progress in the development of user documentation and a small user base (Kuijpers & Telebit, 2003).

2.7.1.4 OMNET++

OMNET++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators (Omnetpp, 2016).

OMNET++ has well structured and open source code. It is also highly modular. In the case of large-scale simulation, simulation modules have to be hierarchical and built from reusable components (Omnet++, 2016). This gives room for easy implementation and software reuse. It is free and modular making it more advantageous than the other simulators.

2.8 Selecting a Simulator, OMNET++

The nature of this project will be best built in modules and reusable components to help save time and avoid redundant code. This forms part of the reasons for choosing the OMNET ++ simulator. Due to the fact that this project will substantially include network simulation and analysis of results, the integrated development environment provided by OMNET++ will make it a better choice. It also provides a highly interactive and user-friendly interface which will ultimately provide a good user experience.

Chapter 3: Methodology

This chapter describes the methodology used in evaluating interference in Bluetooth technology. Section 3.1 describes the research methodology. Section 3.2 explains the transition from high-level specifications to low-level specifications. In Section 3.3, the procedure involved in setting up the simulator for the project is also described. The next section, 3.4 explains the use of certain Omnet++ files in the simulation. Section 3.5 and 3.6 describes the stages involved in the implementation of the hop selection kernel and the data channel selection algorithm. Section 3.7 also explains the simulation process in Bluetooth classic and Bluetooth low energy devices. Section 3.8, focuses on how interference is evaluated while the final section 3.9 discusses certain assumptions used in the simulation process.

3.1 Methodology

In an attempt to evaluate the Bluetooth Classic and Bluetooth low energy technology, considering coexistence, transmission and reception of data, one of the important concepts to look at is interference. Interference is evident in collocated devices where hop frequencies are the same for two or more devices for a particular time. This in effect causes performance degradation which further emphasizes the need for this project.

In order to simulate and evaluate the Bluetooth technology, the Bluetooth network environment was simulated to show how a typical transmission is done both in the classic and Bluetooth low energy technology. The hop selection kernel and data channel selection algorithm are implemented for the generation of frequencies for transmission. In the simulation environment, more piconets will be created and added to the network to study the effects of an increasing number of piconets on interference. The Omnet++ simulator is used for the entire process from the implementation of the Bluetooth's algorithm for selecting frequencies, to the evaluation of interference. The subsequent sections are required steps to make this study and analysis successful.

3.2 Conversion of High-Level Specifications in Bluetooth to Low-Level Specifications

Due to the varying specifications such as the number of frequency channels for both Bluetooth technologies, different operations and parameters will be used for the simulation. Bluetooth classic uses 79 frequency channels while Bluetooth low energy uses 40 frequency channels, with 3 set aside for advertisement and avoidance of Wi-Fi channels in the same 2.4 GHz ISM band (Bluetooth SIG, 2014). Similarly, these two technologies select frequencies determined by the master node in a piconet. The Bluetooth device address and clock signal of the master node are used in the Bluetooth classic technology whereas the channel map and hop increment are used for the Bluetooth Low Energy technology. The two technologies have subtly different mechanisms and a different number of frequency channels for use. The specifications and operations have an influence in how hop selection is done in both technologies and that will be executed in the simulation.

3.3 Setting Up Omnet++ Simulator

Installing Omnet ++ requires prerequisite packages such as Java Runtime and Java runtime Environment for the Mac OS. The Omnet ++ will be set up on the Mac OS for the simulation. Additionally, the command line developer tools should be installed for the Mac OS. To enable more functionality in Omnet++, additional packages such as OpenMpi, GraphViz, and Doxygen are needed. OpenMpi is a powerful message library, GraphViz provides diagrams in HTML documentation, Doxygen also generates documentation for C++ code as part of the HTML documentation that is generated from NED files in the IDE. All these additional features are provided and need to be installed to utilize its functionality.

For debugging, Omnet++ will use Clang automatically whereas the Omnet++ IDE will use gdb as the underlying debugger. There is also an option to use Xcode for debugging.

All these options are available for debugging in Omnet++ for the mac OS. Omnet++ can be downloaded from htpp://omnetpp.org. The required environment variables through the use of command line are set to enable the application to function properly. Omnet ++ needs its directory to be in its path. This is done by editing the ".bashrc" file in the home directory. The following commands are used in the process:

a. $ touch ~/. bashrc
b. $open –e ~/. bashrc
c. export PATH =$PATH: $HOME/omnetpp-5.0/bin

After setting the environment variables, another important step to consider is to configure and build OMNET ++. In the top-level directory of Omnet++, use command $ ./configure to detect the installed software and configuration of the system. The application can be compiled by using the $ make command. Verification can be done with the following commands to check whether sample simulations run correctly.

a. $ cd samples/tictoc

b. $ ./tictoc

Omnet++ can be started from the command line by typing $ omnetpp.

3.4 Omnet++ File Structure

For the implementation process, each file used in the Omnet++ IDE plays a unique role in the success of a simulation. The main files and components used for this project include NED files, C++ files(.cc) and .ini files. The NED files are used for describing the network structure, structure of the simulation model, links etc. It defines the externally visible interface of the module. In this project, a number of NED files are used to create the network structure for both classic and low energy technologies. This file helps in the creation of nodes, piconets and connections between nodes for simulations. The C++ files are also used for functionality implementation, which in this case is the hop selection kernel and the data channel selection algorithm. These will be used to generate frequencies for nodes to hop on for transmissions in the Bluetooth classic and Bluetooth low energy respectively. The ".ini" file is also important to tell which network to run or simulate if a number of them have been created. For example, the ini file can tell the simulation IDE to run a network for the Bluetooth classic or the Bluetooth low energy.

3.5 Implementation of Hop Selection Kernel – Bluetooth Classic

In this subsection, I will discuss the most important component of this project, the hop selection kernel. The hop selection kernel is responsible for generating frequencies using the address and a clock signal of the master device. Using the Bluetooth address given as a 28-bit number and clock signal given a 27-bit number, frequencies are generated for specific time slots (Bluetooth SIG, 2014). Using the Bluetooth algorithm with its dedicated stages, frequencies can be generated for simulation. The hop selection kernel goes through five main stages: the first addition operation, the XOR operation, the permutation stage which has 7 sub stages, the second addition operation and the mapping stage which involves the use of the register bank (Bluetooth SIG, 2014). This is shown in Figure 3.1 below.

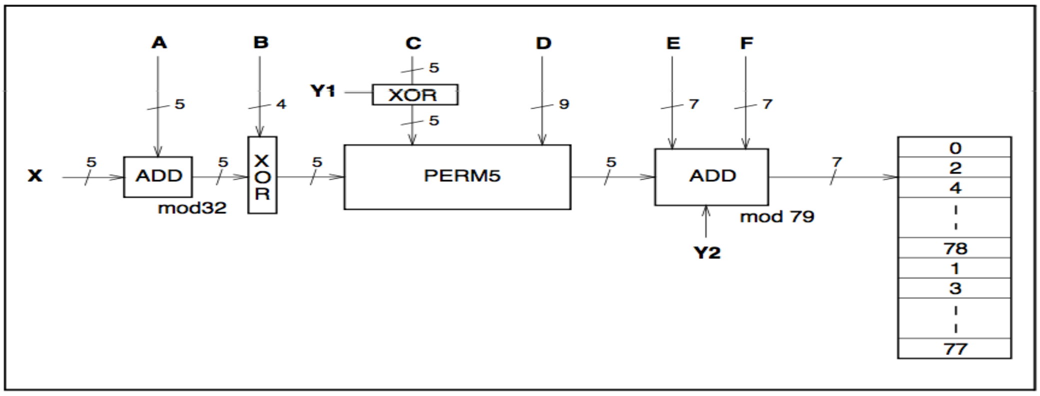


Figure 3.1: Diagram showing the various stages of the hop selection kernel (Bluetooth SIG, 2014).

These stages make use of the master address and the clock signal. The master address determines the hopping sequence while the clock signal determines the phase in the hopping sequence. The component determines the frequency given a set of 79 frequency channels. The mathematical formula for the frequency, f = 2402 + k MHz where k = 0-78. In subsequent subsections, I will describe the main inputs of the hop selection kernel and how the various stages of the hop selection kernel were implemented.

3.5.1 Description of Inputs

In the implementation process, the inputs of the hop selection kernel are represented as arrays with varied sizes. The address bits of the inputs are read from right to left and computations are done in both binary and decimal. The master address is a 28-bit number while the clock signal is a 27-bit number. Specific bits from these two inputs produce the input labels below. The next subsection provides an overview of the role of the different input labels.

Input Label	Value In
A	Address bits 27-23 XOR Clock bits 25-21
B	Address bits 22-19
C	Address bits (8,6,4,2,0) XOR Clock bits 20-16
D	Address bits 18-10 XOR Clock bits 15-7
E	Address bits (13,11,9,7,5,3,1)
F	(16 x Clock bits 27-7) mod 79
X	Clock bits 6-2
Y1	Clock bit 1
Y2	32 x Clock bit 1

Figure 3.2: Input labels of the hop selection kernel (Sandidge, 2002).

Inputs A to D as shown in Figure 3.2 determine the ordering within the segments whereas inputs E to F determine the mapping onto the hop frequencies.

- Clock bit 0 allows two or more devices to synchronize using the clock offset between master and slave. It is not used in the hop selection operation as each time slot covers two clock cycles (Bluetooth SIG, 2014).
- Clock bit 1 given as Y1 and Y2 in figure 3.2 helps to alternate between transmit and receive modes every two clock cycles (Bluetooth SIG, 2014).
- The permutation operation shown as PERM5 in figure 3.1 uses the butterfly operation where Y1, C and D inputs are involved. The butterfly operation uses inputs calculated from the first ADD block and XOR block involving X, A and B as shown in figure 3.1 and figure 3.2.
- The output from the final addition operation is applied to a modulo 79 operation as shown in the last ADD block in figure 3.1. The output is a control word which helps in selecting the frequency channel in the register. The channels in the register are sorted in a way that the first 40 code words select even channels whereas the last 39 select odd channels. This is also illustrated in figure 3.1.
- Hooping occurs sequentially through 32-hop segments. The input X as described in figure 3.2 determines the hop position within each of these 32- hop segments
- As the 32-hop segments end, clock bit-7 increments and input X resets to zero, marking the beginning of the next segment.
- No frequencies are repeated during each 32-hop segment.

3.5.2 Stages

The accuracy of frequencies produced from the hop selection kernel depends on the efficient implementation of this five-stage process.

3.5.2.1 First Addition Operation

The first addition operation shown as the ADD block in figure 3.1 involves the use of input label X and input label A, all given as 5-bit numbers. Input A is derived from the XOR operation of Bluetooth address bits 27-23 and clock bits 25-21. In the reading of bits in the Bluetooth and clock signal, bits are read from the right to the left. Using arrays to store these inputs require the reverse, bits are then read from the left to the right for easy implementation. In order to compute for Input A using the XOR operation, the input bits given in binary, are stored in an array, converted and computed in decimal.

Input label X is a 5-bit number, also derived from clock bits 6-2 as illustrated in figure 3.2, it is also stored in an array. It is converted to decimal for the first addition operation. The addition operation is computed in decimal and involves the use of the inputs derived earlier (input X and A). The resulting decimal is converted back to binary as a 5-bit number. The 5-bit number is then applied to a modulo-32 operation as illustrated in the first ADD block in figure 3.1. The result from the modulo operation is given as Z' and will be used in further operations.

3.5.2.2 XOR Operation

In the second operation of the hop selection algorithm, the first XOR block as shown in figure 3.1, the result, Z', from the first addition operation is computed with B, a 4-bit number derived from the Bluetooth address bits 22-19. Address bits and inputs are stored in arrays and used for computations. Computations are done in decimal and converted back into a binary where the specific bits can be used for further operations. A 5-bit number is produced and used for the permutation operation which consists of a 7-stage butterfly operation. The 5-bit result is denoted as Z.

3.5.2.3 Permutation Operation

The permutation operation involves the use of 4 inputs; input Z, input C, input YI and input D as illustrated in the PERM5 block in figure 3.1. The main role of the permutation operation is to randomize frequency hops. Input Z is a 5-bit number derived from the previous XOR operation. Input C is derived from a XOR operation between address bits (8,6,4,2,0) and clock bits 20-16. Input Y1, on the other hand, is derived from clock bit1. Input D is also derived from a XOR operation between address bits 18-10 and clock bits 15-7. These are illustrated in figure 3.2.

Address bits are read from right to left. Inputs are represented in arrays, forcing the implementation of address bits to be read from left to right. In the permutation operation, the 5-bit number given as Z, from the previous section is used together with the 14-bit control signal to perform butterfly operations. The control signal is denoted as P. P is derived from input D and the result from the XOR operation between inputs C and input Y1. Input D is a 9-bit number represented as P (0-8), while the result of the XOR operation between C and YI is represented as P (9-14). This makes the P control word a 14- bit number used in further computations.

The butterfly operation involves switching the Z bits given the value of each control bit of P. The permutation operation consists of 7 stages of butterfly operations that switch the input bits according to the control word. When P =0, Z bits represented in the array are not changed. When P =1, specific Z bits are swapped. This briefly describes the butterfly operation implementation and how vital it is in the hop selection algorithm.

The control signal is given as P13-P0, read from right to left. In the implementation, this is read from the left to right. The array name is given as P and the first index P[0], of the array, takes P13, P[1] takes P12, P[2] takes P11 in that order to the 13th index. The same procedure is replicated for the Z bits which are represented in an array as 5 bits. It reads from left to right hence Z0 is Z [4] in the array, Z1 is Z [3], Z2 is Z[2], Z3 is Z[1] and Z4 is Z[0], given the array name Z.

Table 3.1 below shows the specific Z bits that are interchanged when P=1 for each bit of the control signal.

Table 3.1: Butterfly Operations for Permutation Operation

Control signal	**Butterfly**		**Control signal**	**Butterfly**
P_0	$\{Z_0,Z_1\}$		P_8	$\{Z_1,Z_4\}$
P_1	$\{Z_2,Z_3\}$		P_9	$\{Z_0,Z_3\}$
P_2	$\{Z_1,Z_2\}$		P_{10}	$\{Z_2,Z_4\}$
P_3	$\{Z_3,Z_4\}$		P_{11}	$\{Z_1,Z_3\}$
P_4	$\{Z_0,Z_4\}$		P_{12}	$\{Z_0,Z_3\}$
P_5	$\{Z_1,Z_3\}$		P_{13}	$\{Z_1,Z_2\}$
P_6	$\{Z_0,Z_2\}$			
P_7	$\{Z_3,Z_4\}$			

The code shown below in figure 3.3 shows one stage of the permutation operation and how Z bits are switched based on the control signal. In this case, the control signal is given in an array called ‘perm’ and the Z bits are given in an array called ‘Zbutter’

```
void permutation1(int Zbutter [], int perm []){
  int temp, temp1, temp2, temp3;

if (perm [0]==1){

  temp=Zbutter [2];
  temp1=Zbutter [3];
 Zbutter[2]=temp1;
 Zbutter[3] =temp;
}
 if(perm [1]==1){

temp2=Zbutter[1];
temp3 =Zbutter[4];

Zbutter[1]=temp3;
Zbutter[4]=temp2;
}
 }
```

Figure 3.3: One stage of the Permutation Operation

3.5.2.4 Second Addition Operation

The output from the permutation stage is added to inputs E, F and Y2 as shown in the second ADD block in figure 3.1. The addition operation is responsible for selecting the hop frequencies used by the current segment. The 5-bit output from the permutation stage plays a role in selecting a channel within the segment. Input Y2 is used for selecting master or slave slots while input F helps to switch each segment to a new set of channels in the connection state. (Bluetooth SIG, 2014).

All inputs are represented in arrays. Input E is a 7-bit number derived from Bluetooth address bits (13,11,9,7,5,3,1). Input F is a 7-bit number derived from the result of multiplying 16 by clock bit 27-2 and applying modulo 79. Input Y2 is also derived from multiplying 32 by clock bit 1.

Inputs E, F and Y2 are therefore added and applied to modulo 79. This helps to limit the output range from 0 to 78. Inputs are converted and computed in decimal.

3.5.2.5 Register Bank

The output from the second addition operation addresses a bank of registers which contain hop frequencies from 0 to 78. The upper half of the register contains even frequency channels whereas the lower half has odd frequency channels (Bluetooth SIG, 2014). The output from the second operations is then mapped to the frequency provided in the register bank. The section of code in figure 3.4 below, illustrates the concept of the register bank where the upper half (0 – 39) contains even channels and the lower half (40-78) contains odd channels.

```
int regAddress [79];
for (int i=0, k=0; i<=39; i++) {

regAddress[i]=k;
  k=k+2;
}

for (int i=40, j=1; i<=78; i++) {
regAddress[i]=j;
  j=j+2;
}
```

Figure 3.4: Register Bank implemented in C++

3.6 Implementation of Data Channel Selection Algorithm – Bluetooth Low Energy

In this section, the implementation process of the data channel selection algorithm used in the Bluetooth low energy(BLE) technology will be discussed. The algorithm using the channel map and hop increment is responsible for the generation of frequency channels for BLE devices to hop on. For the BLE, there are 40 available frequency channels, 3 of which are reserved for advertising and discovery whilst 37 are used for frequency hop selection. The data channel algorithm is a simple straightforward algorithm which requires the use of three main inputs; the channel Map, hop increment and the lastUnmappedChannel to produce a data channel index. The channel map and hop increment are derived from the CONNECT_REQ message sent by the master device to the slave device. The channel map holds a sequence of bits as 0 or 1 to represent unused and used data channels given the data channel index.

The following steps describe the data channel selection algorithm and the implementation process.

a. The data channel index is calculated using the hop increment and the last Unmapped channel, which will be the previous data channel computed. For the first stage, the last Unmapped channel will be 0 since no data channel has been generated yet in the first stage. The hop increment is a random value generated between 5 and 16. The data channel index is hence calculated as the (last Unmapped Channel + hop Increment) mod 37.
b. Given the data channel computed in the previous stage, the channel map is used to check whether the computed data channel is used or unused. If it is used according to the channel map, the data channel is assigned for use in the particular time slot and can be used for transmission or reception of packets. If it is unused the steps below are followed;
c. The number of used channels in the channel map will be determined. A remapping index is calculated by a modulo operation of the data channel index in question and the number of used channels. That is, (data channel index modulo the number of used channels). The result is used as an index to select a new data channel using a "remapping table" which should contain all the used channels in ascending order. The remapping index is used to select the new data channel index from the remapping table containing all the used channels (Kalaa & Refai, 2014).

The snippet of code in figure 3.5 below shows an implementation of the data channel selection algorithm in C++

```
int generateFreq (int chM [], int used [], int chanIndex[]){
    int dataChannel=0;
    int used1=0;
    int reMapindex=0;
    int hopIncrement= generateHI ();

    unMappedChannel= (lastUnMappedChannel + hopIncrement) %37;


    for (int i=0; i<=39; i++) {
        if((unMappedChannel==chanIndex[i]) &&(chM[i]==1)) {

             dataChannel =unMappedChannel;
        }

        else {
            for (int j=0; j<=39; j++) {
                if(chM[j]==1) {
                    used1=used1+1;
                }

            }
            reMapindex = unMappedChannel%used1;

            dataChannel= used[reMapindex];
        }


    }
 lastUnMappedChannel = dataChannel;

    return dataChannel;

}
```

Figure 3.5: A method implementing data channel selection algorithm.

3.7 Simulation of a Transmission in Bluetooth Network

This step involves creating a network environment that imitates the real Bluetooth network environment. Omnet++, the C++ network simulation library and framework will be used for this process. This process considers the factors and conditions necessary for a successful communication between Bluetooth devices. In setting up a piconet for the Bluetooth, each piconet will include a master node and slave. At setup time, the parameters given to the hop selection kernel are the Bluetooth master's address and clock value. The hopping sequence is specific to each piconet and determined by the Bluetooth address of the master while the Bluetooth clock determines the phase in the hopping sequence. The master node uses even numbered slots starting from 0, while the slave nodes use odd numbered slots. The simulation is hence setup to match this condition. Figure 3.6 below shows a sample network structure created with two piconets and a set of master and slave nodes.

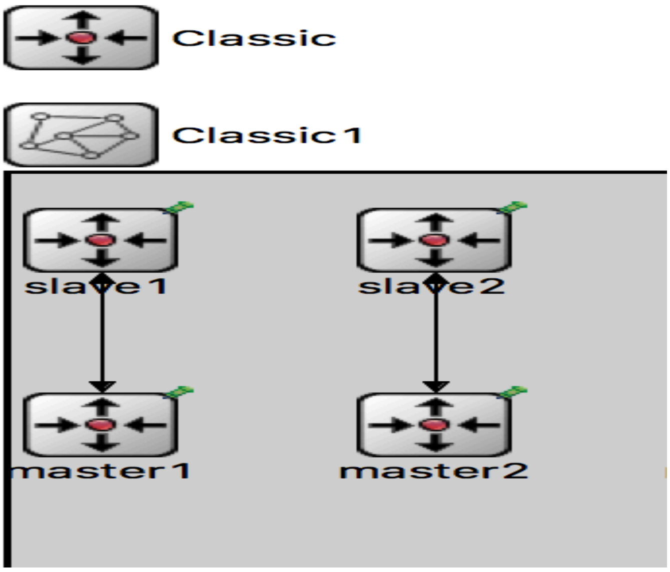


Figure 3.6: Network structure of two piconets with master and slave nodes.

The type of piconet that will be used in both simulations is the synchronized piconet. A synchronized piconet is one that ensures that two or more devices operate in the same or aligned time slot. In reality, piconets are more unsynchronized, ensuring that devices operate in different and unaligned time slots. This explains the fact that Bluetooth devices do not always start operation at the same time.

3.7.1 Bluetooth Classic Simulation

This simulation will be a simple transmission among Bluetooth classic devices in a piconet. To fully implement the Bluetooth classic system, one of the important components to utilize is the hop section kernel, operating under 79 frequency channels. Each piconet will have the master using the hop selection kernel to generate frequencies for transmissions. Random master addresses and clock signals will be chosen for the simulation. In the event of transmissions, collisions can be monitored in situations where two or more devices hop on the same frequency for a particular time slot. In this implementation, collisions will be counted at the receiving end after frequencies are generated. Collisions will be analysed in the synchronized piconets. The collision rate is calculated per the number of packets generated. The number of piconets will be increased to determine the effect of increasing collocated networks on interference. The results are quantified from the transmission, as more nodes are being added to the network.

3.7.2 Bluetooth Low Energy Simulation

Similarly, a piconet with Bluetooth low energy devices will be simulated to observe the results. The number of piconets will be increased to find out how the number of piconets impacts interference and transmission of data. As such, the simulation of the Bluetooth low energy technology will involve the use of the channel map and hop increment. It will utilize 40 frequency channels, with 3 set aside for advertising and discovery (Bluetooth SIG, 2014). Random channel maps and hop increment value are chosen for simulation. Collisions will be counted at the receiving end of the transmission for communications in the piconet.

3.8 Evaluation of Interference

There is the likelihood that for certain time slots, frequencies will match up and overlap due to how the hop selection kernel and data channel selection algorithm works. Interference of signals can occur, causing a reduction in throughput and increase in packet collisions. The interference is hereby quantified in both technologies to allow analysis and conclusions to be made. Interference is said to occur if two or more devices hop on to the same frequency for the same time slot. This is also described as collision and in this case, a collision counter can be used to track the packet transmissions that collide or hop onto the same frequency in the same time slot.

After simulations, results and insights will be visualized using graphs and the network environment, provided by Omnet++. A graph is plotted, mapping the quantified interference/collision against the number of piconets in simulations. The condition to illustrate the effect of an increasing number of piconet will be considered.

The graph should be able to visualize a quantified interference if there is any, against any other condition that is used during the experiment. Drawing from the graph will be insights that will eventually help to evaluate the Bluetooth technology with respect to how it works and its constraints.

3.9 Assumptions

The following assumptions were made during the simulation process of the Bluetooth technology;

a. All nodes/piconets are within transmission range of each other
b. Slaves are synchronized to the master device
c. Attenuation of signals because of the distance of master and slave has not been considered.
d. All nodes/piconets are in a steady connection state.

Chapter 4: Results

The chapter discusses the results from simulations in Bluetooth classic and BLE. Section 4.1 describes the configurations and parameters used for the Bluetooth classic simulation. Section 4.2 discusses simulation results from Bluetooth classic communication against parameters such as the number/type of piconets and collision rate.

4.1 Overview of Configuration and Parameters for Simulation

This section discusses the specific configurations and parameters needed to make the simulation successful. The configurations range from the number of piconets, type of piconet, the number of master and slave nodes in each piconet, the time slots, the specific clock signal and Bluetooth addresses. In the simulation, terms such as collision, time slots and collision rate are described. Collision is described as the incidence where two or more devices/nodes hop on to the same frequency for the same time slot. Time slots can also be explained as the total time span used by nodes for communication.

This aspect of the project considers all the nodes in the network, how long they are simulated for and the number of collisions. Collision rate describes the collision regarding all packets transmitted /received given the time slot. Collision rate is calculated as the total number of collisions divided by the number of time slots given as a percentage. The total time slot describes the time span involved in sending and receiving of packets/messages, Subsequent sections will have the simulation results after a set of tests varying the number of piconets and time slots for the Bluetooth classic and Bluetooth Low Energy technology.

4.2 Simulation Results for Bluetooth Classic

A number of piconets is simulated for the Bluetooth Classic technology. Using random Bluetooth addresses and clock signals, six sets of network piconets are tested out for interference between 60-600 timeslots in simulation time. The time slots are increased in steps of 60 to the last time slot of 600. Each piconet in a network has one master Bluetooth address given as a 28-bit number which will be responsible for generating the frequencies. It also has the Bluetooth slave device, that is the receptive part of the communication where the collision will be quantified. The total number of nodes in a given piconet will be 2.

The set of results in this section will be for a synchronized piconet, where the starting clock signals are the same for the master nodes generating frequencies. This is necessary to ensure a mutual start and end time for the connection of nodes in the network The starting 27-bit clock signal is also given as 0x0000000 for all master addresses in the various piconets.

The following piconets are simulated for the Bluetooth Classic; 2-piconet network, 3-piconet network, 4-piconet network, 5-piconet network, 6-piconet network and 10-piconet network. The various Bluetooth master addresses used in these simulation are randomly chosen hexadecimal numbers given as 28 bits. They include; 0x00000000, 0xA96EF25, 0x587CBA9, 0x2422025, 0xB001913, 0x58A494A, 0x88FC14F, 0xD134F00, 0x8B6B4F and 0xC4A496B. These Bluetooth addresses are used for each master in a piconet to generate frequencies for their respective slave devices.

Tables 4.1- 4.6 given below show sample data collated from simulating Bluetooth Classic network transmissions between 60 to 600 time slots in the 2-piconet network, 3-piconet network, 4-piconet network, 5-piconet network, 6-piconet network and 10-piconet network respectively. The respective collision rates and averages are also shown in the table.

Table 4.1: Collision rates for a 2-piconet Bluetooth Classic network between 60-600 simulated time slots

Two Piconet Network		
Time Slots	Number of Collisions	Collision Rate %
60	0	0
120	0	0
180	0	0
240	0	0
300	0	0
360	0	0
420	0	0
480	0	0
540	0	0
600	0	0
	Average Collision Rate	**0**

The above table 4.2 showed that given a Bluetooth Classic network with two piconets made up of four nodes; two masters and two slaves, with each master generating frequencies for each piconet, the incidence of collision or interference is 0% given the time slots between 60 to 600.

Table 4.2: Collision rates for a 3-piconet Bluetooth Classic network between 60-600 simulated time slots

Three Piconet Network		
Time Slots	Number of Collisions	Collision Rate %
60	2	3.333333333
120	4	3.333333333
180	6	3.333333333
240	8	3.333333333
300	10	3.333333333
360	12	3.333333333
420	14	3.333333333
480	14	2.916666667
540	16	2.962962963
600	18	3
	Average Collision Rate	**3.221296296**

The 3-piconet network consists of 6 nodes; 3 master nodes responsible for generating frequencies and 3 slave nodes, hopping onto these frequencies. Table 4.2 above shows an average collision rate of 3.22% given the duration of 60 – 600 simulated time slots for the 3-piconet network for Bluetooth classic devices.

Table 4.3: Collision rates for a 4-piconet Bluetooth Classic network between 60-600 simulated time slots

Four Piconet Network		
Time Slots	Number of Collisions	Collision Rate %
60	2	3.333333333
120	6	5
180	8	4.444444444
240	12	5
300	18	6
360	24	6.666666667
420	30	7.142857143
480	30	6.25
540	36	6.666666667
600	40	6.666666667
	Average Collision Rate	**5.717063492**

The number of piconets given in table 4.3 is 4. Each piconet has 2 nodes making a total of 8 nodes in the network. 4 of these nodes are master nodes, generating frequencies while the other four are slave nodes, hopping on to these frequencies. Table 4.3 above shows the various collision rates and the average for the specified time slots after simulation.

Table 4.4: Collision rates for a 5-piconet Bluetooth Classic network between 60-600 simulated time slots

Five Piconet Network		
Time Slots	Number of Collisions	Collision Rate %
60	2	3.33333333
120	6	5
180	8	4.44444444
240	12	5
300	18	6
360	24	6.66666667
420	32	7.61904762
480	34	7.08333333
540	40	7.40740741
600	46	7.66666667
	Average Collision Rate	**6.02208995**

The 5-piconet network consists of 10 nodes (5 masters and 5 slaves) in communication between 60-600 simulated time slots. An average of 6.02% was recorded, as shown in table 4.4 above.

Table 4.5: Collision rates for a 6-piconet Bluetooth Classic network between 60-600 simulated time slots

Six Piconet Network		
Time Slots	Number of Collisions	Collision Rate %
60	6	10
120	10	8.33333333
180	16	8.88888889
240	22	9.16666667
300	26	8.66666667
360	30	8.33333333
420	34	8.0952381
480	38	7.91666667
540	42	7.77777778
600	46	7.66666667
	Average Collision Rate	**8.48452381**

The 6-piconet network consists of 12 nodes; 6 master nodes responsible for generating frequencies using their Bluetooth address and 6 slave nodes, hopping onto these frequencies. Table 4.5 above shows an average collision rate of 8.48% given the duration of 60 – 600 simulated time slots for the 6-piconet network for Bluetooth classic devices.

Table 4.6: Collision rates for a 10-piconet Bluetooth Classic network between 60-600 simulated time slots

Ten Piconet Network		
Time Slots	Number of Collisions	Collision Rate %
60	8	13.3333333
120	15	12.5
180	24	13.3333333
240	31	12.9166667
300	35	11.6666667
360	42	11.6666667
420	51	12.1428571
480	59	12.2916667
540	108	20
600	136	22.6666667
	Average Collision Rate	**14.2517857**

The 10-piconet network consists of 20 nodes; 10 master nodes responsible for generating frequencies and 10 other slave nodes, communicating with these master nodes. Table 4.6 above shows an average collision rate of 14.25% given the duration of 60 – 600 simulated time slots for the 10-piconet network for Bluetooth classic devices.

4.2.1 Relationship between Number of Piconets and Collision Rates in Bluetooth Classic

This subsection shows the relevance of the data collected in tables 4.1- 4.6 above as the number of piconets were increased for the simulation. The results and observations show a fascinating relationship between the number of piconets and the average collision rate for a specific number of time slots in a synchronized Bluetooth classic network.

Table 4.7: Number of piconets and average collision rates recorded for simulated time slots between 60-600 in Bluetooth Classic network

Number of Piconets	Average Collision Rate
2	0
3	3.22
4	5.72
5	6.022
6	8.48
10	14.25

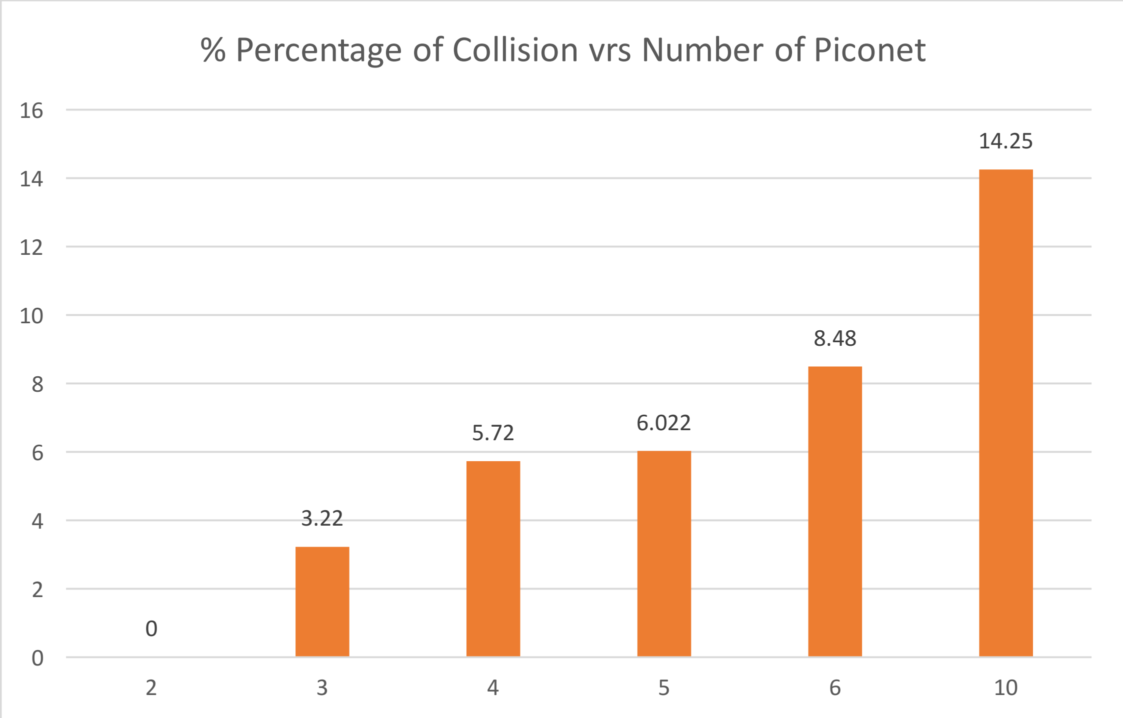


Figure 4.1: Graph showing the percentage of collision rate(y-axis) against number of piconets(x-axis) in the Bluetooth classic network

The number of piconets in each network considers the number of nodes in use. In this simulation each piconet has two nodes: a master and a slave, so given the number of piconets, the total number of nodes is doubled.

In figure 4.1 above, it can be seen that the average collision rate for the Bluetooth classic increases as the number of piconets increases. Collision rates increase from 0 to 14.25% as the number of piconets increases from 2 to 10. This shows a positive correlation between the two factors. This explains that in the transmission of packets in the Bluetooth classic technology, one of the important factors to note is the number of piconets and the number of nodes in use. Clearly, this plays a huge role in interference. The level of interference is therefore high, given an increase in the total nodes and number of piconets. This shows that when a lot of Bluetooth classic devices or nodes coexist in a network environment, the frequencies being generated by the hop selection kernel are more likely to be overlapped given a high number of piconet and nodes.

4.3 Simulation Results for Bluetooth Low Energy

In the simulation of the Bluetooth Low energy(BLE) technology, the channel map and hop increment value are used as parameters to generate frequencies for BLE devices to communicate. The channel map is a bit sequence of 40 bits randomly selected for each BLE master device to generate data channels for the BLE slave devices to hop on. It is made of used channels, which are available for devices to hop on and unused channels which are reserved or unavailable for use. Every data channel is represented with bit ‘0’ or ‘1’as per the data channel index in the channel map to indicate unused and used channels respectively. Another parameter vital to BLE connection is the hop increment, which is a random value generated between 5 and 16 to determine the next hop frequency. Using a random channel map and hop increment for each master BLE device, six sets of network piconets are simulated to check for interference between 60-600 timeslot. The time slots are increased in steps of 60 to the 600th time slot. Each piconet in a network has one master and slave BLE device. The total number of nodes in a given piconet is 2.

The results in this section will contain data collated from different number of piconets. The following piconets are simulated for the Bluetooth Low energy; 2-piconet network, 3-piconet network, 4-piconet network, 5-piconet network, 6-piconet network and 10-piconet network. The channel maps(ChM) used for the BLE master device in these simulation are randomly chosen hexadecimal numbers given as 40 bits. They include; 0x7FF7FFFFFE, 0x5DF65AB5F8, 0x3BF7FBEFFC, 0x33F7F7FFFE, 0x13F7F7FFFE, 0x 73E7FAFEBA, 0x3DF6F5EF7C, 0x151655C63A, 0x7F01FD5FFC and 0x3FB7FFFFFC. These channel maps are used in each BLE master device in a piconet to generate data channels or frequencies for their respective slave devices.

Tables 4.7-4.12 given below show sample data collated from simulating BLE network transmissions between 60 to 600 time slots in the 2-piconet network, 3-piconet network, 4-piconet network, 5-piconet network, 6-piconet network and 10-piconet network respectively. The respective collision rates and averages are also shown in the tables.

Table 4.7: Collision rates for a 2-piconet Bluetooth Low Energy(BLE) network between 60-600 simulated time slots.

Two Piconet BLE Network		
Time Slots	Number of Collisions	Collision Rate %
60	0	0
120	0	0
180	0	0
240	0	0
300	0	0
360	0	0
420	0	0
480	0	0
540	0	0
600	0	0
	Average Collision Rate	**0**

The above table 4.7 shows that given a BLE network with two piconets made up of four nodes; two masters and two slaves, with each master generating frequencies for each piconet, the incidence of collision or interference is also 0% given the time slots between 60 to 600.

Table 4.8: Collision rates for a 3-piconet Bluetooth Low Energy(BLE) network between 60-600 simulated time slots.

Three Piconet BLE Network		
Time Slots	Number of Collisions	Collision Rate %
60	1	1.666666667
120	5	4.166666667
180	11	6.111111111
240	11	4.583333333
300	11	3.666666667
360	13	3.611111111
420	15	3.571428571
480	17	3.541666667
540	18	3.333333333
600	21	3.5
	Average Collision Rate	**3.775198413**

The above table 4.8 for a BLE network with three piconets made up of six nodes; three masters and three slaves, the average collision rate recorded was 3.78% given the time slots between 60 to 600.

Table 4.9: Collision rates for a 4-piconet Bluetooth Low Energy(BLE) network between 60-600 simulated time slots.

Four Piconet BLE Network		
Time Slots	Number of Collisions	Collision Rate %
60	6	10
120	12	10
180	17	9.444444444
240	27	11.25
300	32	10.66666667
360	38	10.55555556
420	42	10
480	48	10
540	55	10.18518519
600	66	11
	Average Collision Rate	**10.31018519**

The above table 4.9 for a BLE network with four piconets made up of eight nodes; four masters and four slaves, the average collision rate recorded was 10.31% given the time slots between 60 to 600.

Table 4.10: Collision rates for a 5-piconet Bluetooth Low Energy(BLE) network between 60-600 simulated time slots.

Five Piconet BLE Network		
Time Slots	Number of Collisions	Collision Rate %
60	8	13.3333333
120	13	10.8333333
180	23	12.7777778
240	31	12.9166667
300	45	15
360	54	15
420	67	15.952381
480	73	15.2083333
540	87	16.1111111
600	119	19.8333333
	Average Collision Rate	**14.696627**

The above table 4.10 for a BLE network with five piconets made up of ten nodes; five masters and five slaves, the average collision rate recorded was 14.69% given the time slots between 60 to 600.

Table 4.11: Collision rates for a 6-piconet Bluetooth Low Energy(BLE) network between 60-600 simulated time slots.

Six Piconet BLE Network		
Time Slots	Number of Collisions	Collision Rate %
60	13	21.6666667
120	22	18.3333333
180	25	13.8888889
240	34	14.1666667
300	47	15.6666667
360	68	18.8888889
420	78	18.5714286
480	86	17.9166667
540	91	16.8518519
600	128	21.3333333
	Average Collision Rate	**17.7284392**

The above table 4.11 for a BLE network with six piconets made up of twelve nodes; six masters and six slaves, the average collision rate recorded was 12.73% given the time slots between 60 to 600.

Table 4.12: Collision rates for a 10-piconet Bluetooth Low Energy(BLE) network between 60-600 simulated time slots.

Ten Piconet BLE Network		
Time Slots	Number of Collisions	Collision Rate %
60	18	30
120	27	22.5
180	34	18.8888889
240	56	23.3333333
300	65	21.6666667
360	89	24.7222222
420	92	21.9047619
480	105	21.875
540	190	35.1851852
600	210	35
	Average Collision Rate	**25.5076058**

The above table 4.12 for a BLE network with ten piconets made up of twenty nodes; ten masters and ten slaves, the average collision rate recorded was 25.51% given the time slots between 60 to 600.

4.3.1 Relationship between Number of Piconets and Collision Rates in Bluetooth Low Energy

This section shows how an increasing piconet number affects collision or interference. From the tables given above, it can be seen that as the number of piconets was increased for the Bluetooth low energy simulation, the collision rate increases. The average collision rates for the time slots between 60-600 were computed for the piconet types and shown in table 4.13 below. This is to help plot a graph to illustrate the relationship between the number of piconets and collision rates as shown in figure 4.2 below.

Table 4.13: Number of piconets and average collision rates recorded for simulated time slots between 60-600 in a BLE network

BLE Network	
Number of Piconets	Average Collision Rate %
2	0
3	3.78
4	10.31
5	14.69
6	17.73
10	25.5

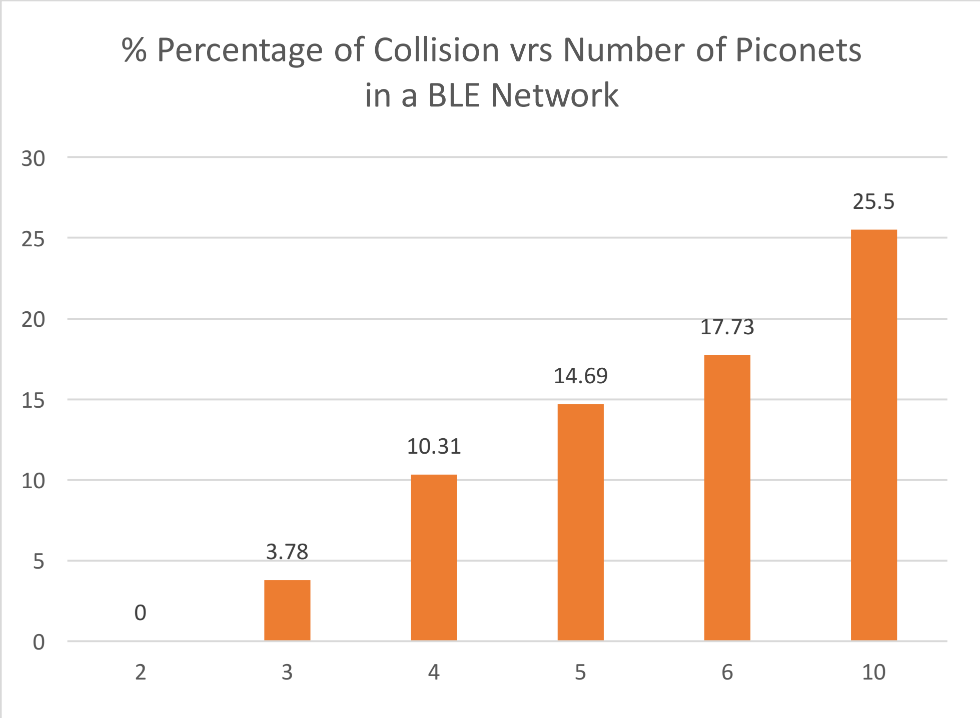


Figure 4.2: Graph showing the percentage of collision rate(y-axis) against number of piconets(x-axis) in Bluetooth low energy network

The Figure 4.2 above shows a positive correlation between the number of piconets and the collision rate. Collision rates increase from 0 to 25.5% as the number of piconets increases from 2 to 10. This is to say that, as the number of piconets in a Bluetooth Low Energy (BLE) network increases, the collision rate also increases. This is influential in BLE coexistence especially with relation to the internet of things. This is because, many wireless devices which will be connected to the internet of things, will be powered by the Bluetooth low energy technology and while these devices coexist, collision or interference is 0 to 25.5% likely, given a piconet number between 2 and 10.

4.4 Interference in Bluetooth Classic and Bluetooth Low Energy

In this section, interference or collision is discussed with respect to the two Bluetooth technologies; Bluetooth Classic and Bluetooth Low Energy. Given the simulated timeslots between 60 and 600 and a given set of piconets, the collision rates for the two technologies are discussed.

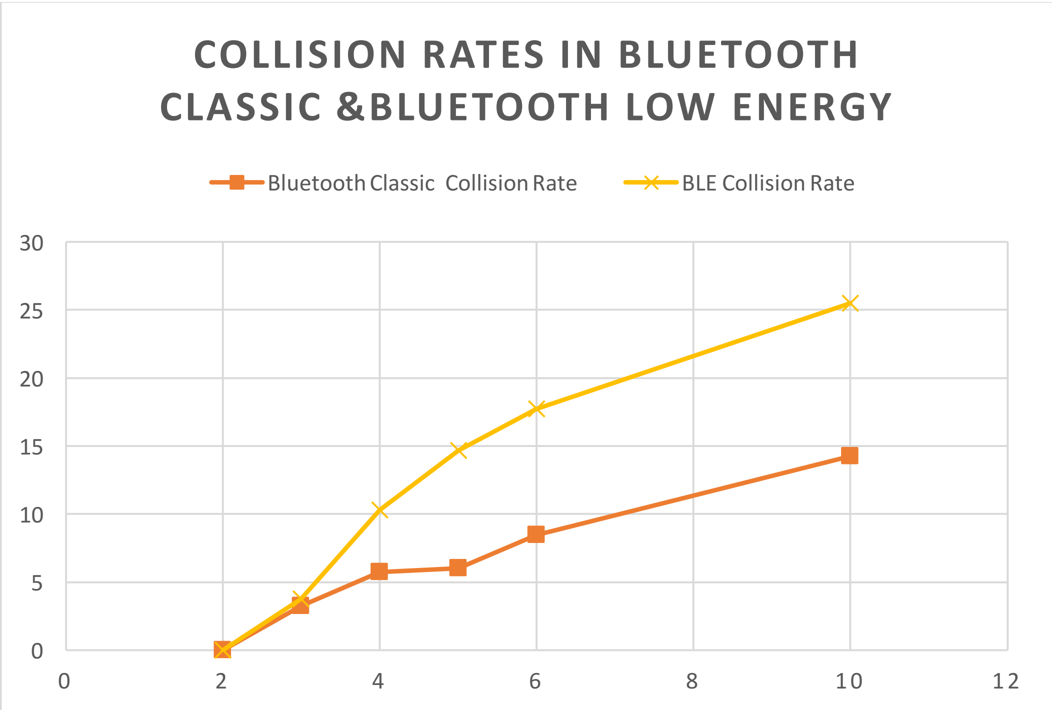


Figure 4.3: Graph showing collision rates(y-axis) against the number of piconets in Bluetooth Classic and Bluetooth Low Energy.

After collating the simulated results in the Bluetooth classic and Bluetooth Low energy technologies, the collision rates were analysed against the increasing piconet number. The Bluetooth Low Energy recorded a positive correlation between the number of piconets and the collision rates. The collision rates recorded, range from 0% to 25.5% in the Bluetooth Low Energy technology.

Also, the Bluetooth Classic recorded a positive correlation between the number of piconets and the collision rate. Collision rates recorded for Bluetooth Classic range from 0% to 14.25%. However, the collision rates in Bluetooth low energy were found to be higher for the given simulation time and piconet numbers. From figure 4.3 above, it can be seen that the collision rates in Bluetooth Low energy, from piconet number 4 to piconet number 10, were almost as double the collision rate in Bluetooth classic.

This therefore shows that, given a set of piconets from 2 to 10 and a simulated time slots between 60 to 600, the incidence of interference or collision in Bluetooth Low energy is higher as compared to interference in Bluetooth Classic.

Chapter 5: Limitations, Conclusion and Recommendation

In this project, interference or collision has been quantified and evaluated for the two Bluetooth technologies; Bluetooth Classic and Bluetooth Low Energy. For a given Bluetooth network with the piconet number say 2 to 10, interference or collision is likely to happen at the rates of 0% to 14.25% in the Bluetooth Classic technology and 0% to 25% in the Bluetooth Low Energy technology for simulated time slots between 60 to 600. From the results, it can be said that the Bluetooth Low energy has a relatively high incidence of interference as compared to the Bluetooth Classic technology.

5.1 Limitations

The evaluation of interference in Bluetooth is critical in how packets are transmitted among Bluetooth devices. The most critical part of the Bluetooth Classic system is the hop selection kernel which computes frequencies using a master node's address and a clock signal. The data channel selection algorithm is also essential in how Bluetooth Low energy devices communicate. The accuracy and efficiency of both algorithms depend a lot of how logically the system is implemented. The logic of the two algorithms is complicated for implementation and makes it difficult to developers to correctly imitate and verify frequencies being generated by the Bluetooth Special Interest group.

Another limitation for this project is the fact that simulations do not efficiently imitate real systems. Simulations are almost always likely to ignore certain factors which could affect and skew the accuracy of results. Technological advancements have always given us reasons to do better and the level of change that is made each day makes it difficult to keep up with. This introduces a lot of factors and complicates the world of wireless communications. All these changes and updates are difficult to cater for and cause a limitation in the execution of the project.

In terms of implementation and evaluation, one of the key limitations to the project is coexistence and presence of many devices or nodes operating in 2.4Ghz ISM band. This project does not feature coexistence between Bluetooth Classic and Bluetooth Low Energy devices and their effect on interference. Signal strength for transmissions also decreases with increasing distance between Bluetooth nodes. This can have an effect on the results obtained. The simulation does not fully replicate this situation. Also, results obtained in this project are for worst case scenarios.

For the two Bluetooth technologies, all the other states and operations involved in the Bluetooth system such as discovery, advertising etc are ignored. The only state considered during simulation in both technologies is the connection state. Most Bluetooth devices work in an unsynchronized fashion especially in relation to the use of the clock signals and this aspect is not featured in the implementation of Bluetooth Classic and Bluetooth Low Energy for this project.

5.2. Recommendation/Future Work

Based on the limitations discussed above, further work will be focused on a simulation between Bluetooth Classic and Bluetooth Low energy devices transmitting in coexistence. More work will be done in both Bluetooth classic and Bluetooth Low Energy technologies to see how much throughput is affected by interference during communication.

5.3 Conclusion

The impact of the internet of things cannot be underestimated and constant development is needed to make the best out of this revolution. The Bluetooth technology is essential in the internet of things and this project has been able to evaluate interference in the Bluetooth technology considering its adverse impact on the internet of things. Although work has been done to minimize the level of interference, the situation seems to be inevitable and is causing a lot of disruptions in wireless communications.

The project helped to evaluate interference in the Bluetooth classic technology and Bluetooth Low Energy technology. In this evaluation, simulations were done to appropriately imitate a real Bluetooth network environment using OMNET++. Various factors such as the number of nodes/devices and number of piconets were tested in a set of simulations to find out the relationship between these factors and interference in Bluetooth. The aim was to evaluate and quantify interference in Bluetooth technology (Bluetooth Classic and Bluetooth Low Energy) and this has been done effectively.

In the results discussed earlier, collision rates were produced given the number of piconets and nodes. Additionally, interesting insights were gained; it was observed that interference/collisions in Bluetooth Classic and Bluetooth Low Energy increases as the number of piconets/nodes for communication increases. This is shown in figure 4.1 and 4.2. The Bluetooth Low Energy technology also has a relatively high incidence of interference or collision as compared to the Bluetooth Classic technology given a set of piconets. This can be seen in figure 4.3. This explains that for an efficient level of transmission in Bluetooth technology, fewer devices or nodes should be collocated to each other for less interference.

References

Bluetooth. (2013). [Online]. Available: www.bluetooth.com Evans, "The Internet of things: How the next evolution of the Internet is changing everything," CISCO, San Jose, CA, USA, White Paper, 2011

Bluetooth SIG. (2014). Specification of the Bluetooth System Version 4.2: Architecture. Bluetooth SIG. (Vol. 2), 66 –100.

Decuir, J. (2014). Introducing Bluetooth Smart: Part II: Applications and updates. IEEE Consumer Electronics Magazine, 3(2), 25-29. doi:10.1109/mce.2013.2297617

Gartner. (2013). Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020. Retrieved April 17, 2017, from http://www.gartner.com/newsroom/id/2636073

Golmie, N., Van Dyck, R., & Soltanian, A. (2001). *Interference of Bluetooth and IEEE 802.11: simulation modeling and performance evaluation* (1st ed., pp. 11-18). Rome: ACM New York, NY, USA. Retrieved from http://dl.acm.org/citation.cfm?id=381598&CFID=848276760&CFTOKEN=4822552

Imran, M., Said, A. M., & Hasbullah, H. (2010). A survey of simulators, emulators and testbeds for wireless sensor networks. *2010 International Symposium on Information Technology*. doi:10.1109/itsim.2010.5561571

Kalaa, M. O., & Refai, H. H. (2014). Bluetooth standard v4.1: Simulating the Bluetooth low energy data channel selection algorithm. 2014 IEEE Globecom Workshops (GC Wkshps). doi:10.1109/glocomw.2014.7063519

Mathew, A., Chandrababu, N., Elleithy, K., & Rizvi, S. (2009). IEEE 802.11 & Bluetooth Interference: Simulation and Coexistence. *2009 Seventh Annual Communication Networks and Services Research Conference*. doi:10.1109/cnsr.2009.41

Omnetpp. (2016). Why Choose Omnet++ ? Advantages of Omnet++. (2016). Retrieved November 7, 2016, from http://omnetplusplus.com/omnet/

R. Heydon, *Bluetooth Low Energy,* The Developers Handbook, R. Heydon, Ed. Englewood Cliffs, NJ: Prentice-Hall, 2012.

Sandidge, J. M. (2002). Bluetooth Hop Prediction. 1-32. Retrieved from https://ttu-ir.tdl.org/ttu-ir/bitstream/handle/2346/19895/31295017082172.pdf?sequence=1.

Xin, Y., Wang, X., Leus, G., Yue, G., & Jiang, J. (2011, March 6). Interference Management In Wireless Communication Systems: Theory and Applications. *EURASIP Journal on Wireless Communications and Networking, 2010*(1), 687649. doi:10.1155/2010/687649