VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Quynh Le

IEEE 802.15.4 COMMUNICATION PERFORMANCE TESTING IN INDUSTRIAL ENVIRONMENT

Technology and Communication
2021

ACKNOWLEDGEMENTS

This thesis has been done during my bachelor study at VAMK, University of Applied Science, Information Technology department.

My most sincere appreciation I would like to send to Dr. Gao Chao for his great supervision. He did not only support, guide me on this project but also encouraged me throughout my thesis implementation. He plays an important role contributing to my thesis completion.

I also want to extend my appreciation to Mr. Ari Urpiola, Laboratory Engineer, and other Technobothnia staffs for assisting me to finish my experiment in laboratory.

My big thanks I wish to send to all my teachers and classmates at VAMK, University of Applied Science for all their guidance, solution recommendations for past years, especially during the period of my thesis.

Lastly, from the bottom of my heart, I express my gratitude to my family and my friends, who always care of me, give me love and cheer me up.

I appreciate all their kind support and encouragement to me.

Quynh Le

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Quynh Le
Title	IEEE 802.15.4 Communication Performance Testing in Industrial Environment
Year	2021
Language	English
Pages	48
Name of Supervisor	Gao Chao

The purpose of this topic was to study the communication performance of IEEE 802.15.4 network in an industrial environment. IEEE 802.15.4 is a standard designed for short range and low data transmission rate wireless network, mainly found in many wireless sensor devices. These sensor networks are being used in various application areas and industry. However, the industrial environment contains numerous equipment generating electromagnetic waves or consuming frequency band 2.4GHz for their communication. These major factors affect to quality of IEEE 802.15.4 network, cause data loss during the transmission and reduce connection stability.

The study was carried out by conducting experiments in the Technobothnia laboratory where an industrial workplace was simulated. Two Digi XBee Radio Frequency (RF) modules were used to construct a simple IEEE 802.15.4 network where one module took a task of sending 1000 packets in a certain time period to the remaining module. Data was collected on the receiver side and analyzed for conclusions. By these characteristics, Python was used in this project to control XBee modules and visualize data.

Based on data gathered, the project shows how environment affects the IEEE 802.15.4 connection performance, monitors IEEE 802.15.4 network latency and concludes with data loss pattern in IEEE 802.15.4 transmission.

Keywords IEEE 802.15.4, performance testing, industrial environment

CONTENTS

ABSTRACT

LIST OF ABBREVIATIONS

LIST OF FIGURES AND TABLES

LIST OF ABBREVIATIONS

ACK	Acknowledgement
API	Application Programming Interface
AT	Application Transparent
CAP	Contention Access Period
CFP	Contention Free Period
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
FFD	Full-Function Device
IEEE	Institute of Electrical and Electronics
IoT	Internet of Thing
ISM	Industrial, Scientific, and Medical
LAN	Local Area Network
LLC	Logical Link Control
LR-WPANs	Low-rate Wireless Personal Area Networks
MAC	Medium Access Control
NFC	Near Field Communication
OSI	Open Systems Interconnection
P2P	Peer-To-Peer
PAN	Personal Area Network
PC	Personal Computer
PER	Packet Error Rate
PHY	Physical layer
PPDU	PHY Protocol Data Unit
PSDU	Physical Layer Service Data Unit
RF	Radio Frequency
RFD	Reduced-Function Device
RFD	Reduced-function device
RSSI	Received Signal Strength Indication
Rx	Receiver
SPD	Start of Packet Delimiter
Tx	Transmitter

UART	Universal Asynchronous Receiver/Transmitter
WAP	Wireless Access Point
WPAN	Wireless Personal Area Network

1 INTRODUCTION

In contemporary society, the demand of interaction is increasingly high, leading scientists continuously seek better solutions to connect people and objects together. While WiFi (also known as IEEE 802.11 standard) is a technology for high data rate wireless connection between computers and devices in a local area network (such as hospitals, schools, and offices), IEEE 802.15.4 is known as a standard mostly for low data rate devices and in close distance wireless network. IEEE 802.15.4 defines operation in only physical and data link layers in its standard and leaving upper layers for other protocol developments as Zigbee, 6LoWPAN, ISA100.11a, WirelessHART, MiWi, Thread, etc./1/ The key advantages of IEEE 802.15.4 sensor devices are low latency and long battery life. Therefore, IEEE 802.15.4 is the platform for many civil or industrial Internet of Thing (IoT) solutions implemented; for instance, smart meters, smart home (lighting, thermostats), wireless light switches, smart grid, industrial equipment monitoring, tracking device, environment monitoring, smart agriculture, automated factories and industrial plants, predictive maintenance and so on.

In the current industry 4.0 century, more and more IoT applications are leveraged to increase production efficiency and reduce machine downtime. However, the industrial workplace is a concerning environment for wireless signals. It consists of numerous causes of noise to the communication link. The major interference comes from electric equipment generating electromagnetic induction, such as transmitters, transformers, electric motors, heaters, lamps, and power supplies and from devices operating at a frequency of 2.4GHz, the same frequency band as IEEE 802.15.4, such as Bluetooth, 802.11b and 802.11g wireless devices, wireless video cameras, smart meters, car alarm, and cordless telephones. /2/ These interferences certainly challenge our connectivity, degrade the network quality, increase error rate or even lose data. Hence, studying performance of IEEE 802.15.4 connection in an industry scenario is the first need before looking for possible remedies solving this challenge.

This documentation contains six chapters where the first chapter introduces the project. The second chapter is the theorical background of IEEE 802.15.4 standard and a part of WiFi, which was used as a major cause of interference in the IEEE 802.15.4 network. The third chapter introduces equipment required for examination and testing environment. Chapter 4 describes process of experiment implementation. Chapter 5 is the presentation of data analysis and the findings while the last chapter is the experiment conclusion.

2 THEORETICAL BACKGROUND

In this chapter, we introduce the technological background of IEEE 802.15.4, including its physical layer specifications, such as channels, frequency spectrum, transmission rate, and network topology, and its data link layer specification. Then we briefly discuss some key points of WiFi, which is sharing frequency band with IEEE 802.15.4, and be one of causes interfered to IEEE 802.15.4 network. Lastly, some of wireless interference causes are highlighted.

2.1 IEEE 802.15.4 Standard

2.1.1 Overview

The first edition of IEEE 802.15.4 was released in May 2003. It is a standard intended for Wireless Personal Area Network (WPAN) which focuses on transferring data over a short distance between a group of close devices or among personal devices themselves. A WPAN is from a few centimetres to a few meters wide. IEEE 802.15.4 network usually requires little or no underlying infrastructure allowing to save cost for construction. In addition, the transmission rate defined in IEEE 802.15.4 is quite low, 250kbps at maximum, with the result that IEEE 802.15.4 devices consume very little power. Therefore, they can last for months or even for years on battery.

IEEE 802.15.4 specifies specifications in only two lower layers of the OSI model - physical and data link, while higher layers are opened freely for other standards. Two most protocols cooperating well with IEEE 802.15.4 platform are ZigBee and 6LoWPAN. Leaving upper layers for other developments allows different systems employ appropriate technology to bring the highest efficiency, increase diversity of applications.

2.2 IEEE 802.15.4 Frequency Spectrum Channels and Data Rate

Along the radio spectrum (frequencies from 30Hz to 300GHz of the electromagnetic spectrum), ISM band is a range of frequencies are initially reserved globally for Industrial, Scientific, and Medical purposes (other than telecommunications) without a license required. Some of example applications in this band are microwave ovens, industrial heaters, RF welders, medical diathermy machines. However, over the years, ISM band, especially 2.4GHz band, has been excessively used for numerous new short-range, low power wireless communications systems like cordless phones, Bluetooth, Near Field Communication (NFC), WiFi, ZigBee and many other wireless applications. Even the increasing congestion in this radio frequency band causes electromagnetic interference and communication disruption, the unlicensed use of ISM band has been still attractive and the population of "non-ISM" applications is expanding more.

IEEE 802.15.4 aligns well to the free-license ISM radio band. IEEE 802.15.4 operates on three bands, 2.4GHz, 915MHz and 868MHz with the maximum data rate of 250kbps, 40kbps and 20kbps, respectively; in the current version, 100kbps was added for 868/915MHz frequency band /1/. The summary of IEEE 802.15.4 frequency bands can be seen in Table 1. /3/

Table 1. IEEE 802.15.4 RF Channel details.

FREQUENCY BAND (MHz)	CHANNELS AVAILABLE	DATA TRANSFER RATE (kbps)	REGION USE
868 – 868.6	1	20	Europe
902 – 928	10 (2003 release) 30 (2006 release)	40	USA
2400 – 2483.5	16	250	Global

Some other frequency bands were added to IEEE 802.15.4 standard for adapting frequency bands used in specific countries, such as 314-316MHz, 430-434MHz, and 779-787MHz frequency bands allowing to use in China and the 950-956 MHz band in Japan /3/.

Among these bands, the 2.4GHz band is the most widely used. The 2.4GHz band supports 16 channels with channel spacing of 5MHz. They are numbered from 11 to 26 (or 0x0B to 0x1A in hexadecimal) and many of them overlap with the most use WiFi channels (1, 6, 11). Therefore, it is worthy to study IEEE 802.15.4 channels performance in a high interference environment of IEEE 802.11. Figure 1 illustrates how WiFi and IEEE 802.15.4 channels allocate in ISM 2.4GHz radio band.

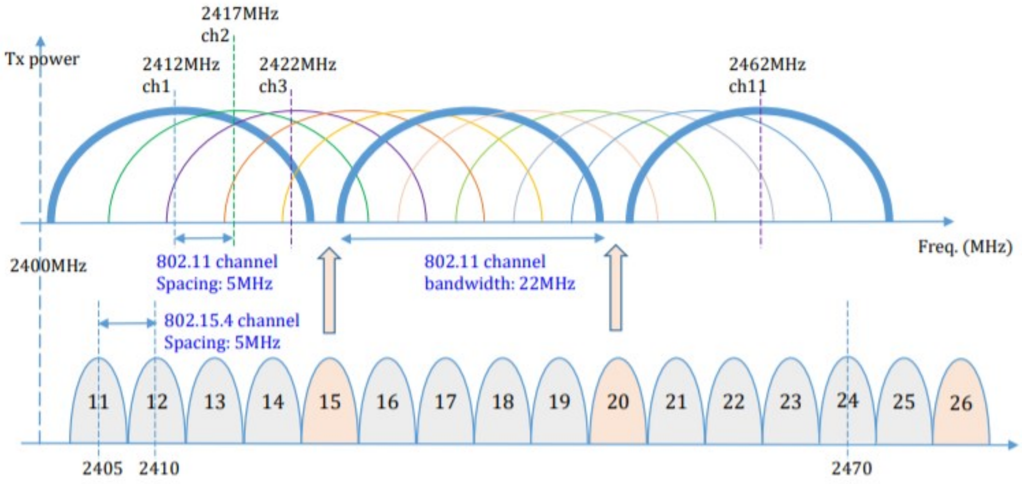


Figure 1. WiFi and 802.15.4 2.4GHz channels allocation /4/.

2.2.1 Network Nodes

There are two types of network node were defined in IEEE 802.15.4 standard:

- **Full-Function Device (FFD)**:

This node has full functionality. It is capable of communicating to any other node within the network: sending/receiving data, or routing data to other nodes. The term “coordinator” is a special form of FFD; it acts as a controller of the whole network, and sometimes it is the representative of a network.

- **Reduced-Function Device (RFD)**:

In contrast to FFD, an RFD is an extremely simple device or network resource. It talks only with an FFD but does not interact to any other devices. RFD spends most of its time in sleep mode, so it conserves less power than FFD.

2.2.2 Network Topologies

IEEE 802.15.4 standard supports two types of network topology:

- **Star topology**:

As the name implies, it requires one FFD, usually be called *coordinator or Personal Area Network (PAN) coordinator,* which is placed in the center of network and all nodes within the network can talk to. A simple star topology is visualized in Figure 2.

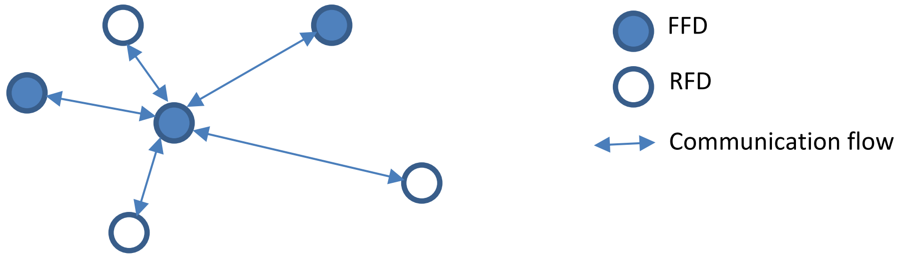


Figure 2. Star network topology.

- **Peer-to-Peer network topology**:

In a Peer-To-Peer (P2P) network topology, each FFD node can communicate with any other node directly in its radio range, while RFD performs only simple communication to the network coordinator. This kind of network topology has more advantages than the star model: network coverage is increased, and its size is easy to expand. Figure 3 gives an illustration of a P2P network topology.

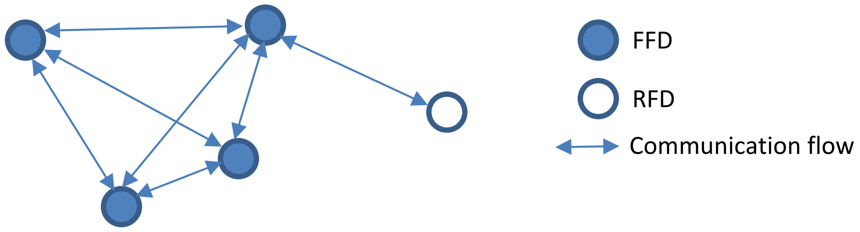


Figure 3. Peer-to-Peer network topology.

Each PAN identifies itself by an arbitrary PAN ID and each device within a network is assigned a 64-bit address or short 16-bit address (in a restricted environment).

2.2.3 Physical layer (PHY) Packet Structure

Though different frequency bands are defined, and different bit rates are used in those bands, IEEE 802.15.4 uses a common packet structure in all physical situations to interface to MAC layer. Each packet, or PHY protocol data unit (PPDU) is constructed from four fields as shown in Figure 4. Physical Service Data Unit (PSDU) field can be up to 127 bytes long including MAC layer header and tailer.

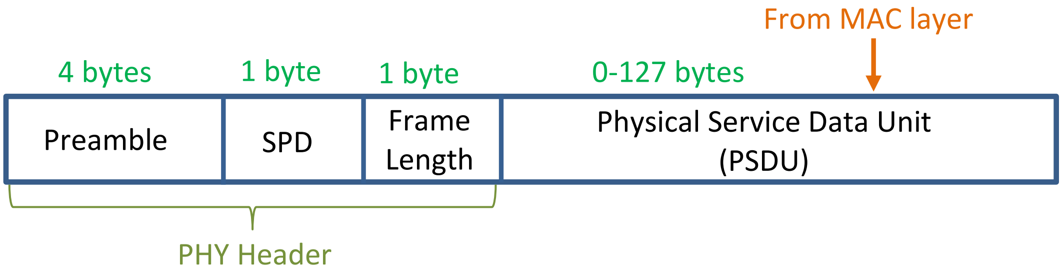


Figure 4. IEEE 802.15.4 PHY Packet structure.

2.2.4 MAC protocol in IEEE 802.15.4

As stated earlier, IEEE 802.15.4 specifies the protocol in only two lower layers: physical and data link layer or MAC sub-layer for more accuracy. According to IEEE 802, the data link layer is divided into two sub-layers: MAC and LLC (Logical Link Control). The IEEE 802.15.4 MAC layer manages the access to radio channels by employing CSMA/CA mechanism. The basic idea behind CSMA/CA is that a carrier being used is sensed to avoid collisions between nodes' traffic, a node begins do transmit only after the channel is sensed to be clear or idle /5/.

There are two variants of CSMA/CA mechanism being used in IEEE 802.15.4: the beacon-enabled access method, also called *slotted CSMA/CA*, and the non-beacon-enabled access method, also called *unslotted CSMA/CA*. If the beacon-enabled access method is being activated, the PAN coordinator periodically broadcasts a superframe with a beacon included to the devices; while in the non-beacon-enabled access method, the coordinator does not send a beacon until it receives a beacon request from a device. A superframe contains two time periods: Contention Access Period (CAP) and Contention Free Period (CFP) which enables a contention-free communication between the PAN coordinator and its slave nodes. The structure of the superframe is defined by PAN coordinator.

2.2.5 Other IEEE 802.15.4 Features:

Transmitter Power (represented in dBm) is the power level measured at transmitter. IEEE 802.15.4 standard aims to offer a solution to prolong the working period of a wireless device, so the transmitter power used in IEEE 802.15.4 is low. While IEEE 802.15.4 specifies the nominal transmitter power as -3dBm (0.5mW) /6/, other wireless application such as IEEE 802.11 at the same ISM band allows 20dBm (100mW) transmitter power /7/. This poses a great threat to IEEE 802.15.4 communication performance.

Receiver Sensitivity: According to IEEE 802.15.4, receiver sensitivity is defined as the smallest signal power that receiver results in less than 1% packet-error-rate. The receiver sensitivity for 2.4GHz band is -85dBm and -92dBm for 868/915MHz /6/.

Received Signal Strength Indication (RSSI): As its name states, RSSI – Received Signal Strength Indicator – is an indicator that measures the strength of power present in a received signal. If a transmitter is close to the receiver, the transmitted signal strength at receiver side is high, likewise, it decreases when the transmitter moves father away receiver. Measuring the signal strength on the receiver side can be determine the quality of network link. RSSI is measured in dBm.

2.3 WiFi (IEEE 802.11 Standard)

We must be very familiar with WiFi, a wireless network protocol based on IEEE 802.11 standard family, which is commonly used for digital devices in a Local Area

Network (LAN) accessing the Internet or exchanging data by radio waves. WiFi is widely used in homes and small offices to link computers (laptops, desktop computers), mobile devices (smartphones, tablets), and other equipment (printers, smart TVs, cameras) together and to a wireless router. This router acts as a Wireless Access Point (WAP) and allows WiFi devices connect to a wired network, including the Internet. /8/

There are various IEEE 802.11 WiFi standards, which are differed by different radio technologies determining its operation frequency in radio bands, the maximum ranges and speeds that it can manage. The newest WiFi versions in WiFi Alliance is branded as “WiFi 6” or “802.11ax”, which operates in the 2.4GHz and 5GHz bands with data rates speeding up to range of multi-gigabit and WiFi 6E (Extended WiFi 6) is specified at the 6GHz band. The first version of WiFi standard, known as “WiFi 1” or “802.11b”, was active at 2.4GHz in radio frequency spectrum with the speed of data rate to 11Mbps. /9/

We are focusing on WiFi 2.4GHz (802.11b) because it is more related to IEEE 802.15.4. As Figure 1 shows, IEEE 802.11 utilized 2.4GHz ISM band into 14 overlapping channels, each channel is 22MHz wide and separated from one another by 5MHz, except a 12MHz space before channel 14. Therefore, there are certain overlaps between 802.11b channels, causing signal interference on adjacent channels. The US allows to use first 11 channels, while the EU is using 13 channels and Japan has all 14 channels. The mostly used IEEE 802.11b channels are channel 1, 6, 11.

2.4 Wireless Interference

Wireless interference is always an important consideration once a wireless network is planned. Wireless interference comes from various sources. Unfortunately, interference is unavoidable but minimizing the level of interference is possible. The following are factors that can cause interference:

Physical objects include trees, buildings and other physical structure objects. The density of the materials is higher, it is harder to maintain the RF signals. A signal is particularly difficult to pass through concrete and steel walls, they may make the connection dropped or weaken.

Radio frequency interference: The frequency 2.4GHz is a favorite radio range of many devices or technologies, such as WiFi, ZigBee, IEEE 802.15.4, Bluetooth, microwaves, cordless phones, and so on. Devices that share the same frequency band certainly cause interference to each other.

Electrical interference comes from computers, lightings, or any motorized devices. The impact of electrical interference to the signal depends on the closeness of the electrical device to the wireless station. This kind of interference has been reduced in new wireless technologies and electrical devices, as well.

Environment: Lightning, fog and other weather conditions may be a great cause impact to the wireless signal.

Table 2 highlights a few example obstacles to the wireless signal. /10/

Table 2. Wireless Obstacles Found Indoor.

Obstruction	**Obstacle Severity**	**Sample Use**
Wood/wood paneling	Low	Inside a wall or hollow door
Drywall	Low	Inside walls
Furniture	Low	Couches or office partitions
Clear glass	Low	Windows
Tinted glass	Medium	Windows
People	Medium	High-volume traffic areas that have considerable pedestrian traffic
Ceramic tile	Medium	Walls
Concrete blocks	Medium/High	Outer wall construction
Mirrors	High	Mirror or reflective glass
Metals	High	Metal office partitions, doors, metal office furniture
Water	High	Aquariums, rain, fountains

3 TESTING SYSTEM AND ENVIRONMENT

The purpose of this project is to study how IEEE 802.15.4 network behaves in an industrial environment; hence the method we are applying is to create a simple IEEE 802.15.4 network, place it in an environment which simulates an industrial area, use a WiFi Access Point as the major radio interference source, and then perform experiments. In this chapter, we describe our testing system needed to perform tests, such as hardware, software, tools, and the test environment as well. We also discuss some terminologies in RF technology.

3.1 Testing System

We used two DiGi XBee S1 802.15.4 RF module to build an IEEE 802.15.4 network as P2P topology. Two modules wirelessly communicated with each other follow IEEE 802.15.4 protocol; one module acted as a transmitter (Tx) and the remaining takes receiver's (Rx) responsibility.

Each XBee module was connected to a host computer via USB interface. These two PCs were around 10 meters in distance. They had XCTU software and Digi XBee Python Library installed. The XCTU software was used to setup XBee modules for their operations and IEEE 802.15.4 radio parameters, such as PAN ID, source address, destination address, and operating mode (will be described more details later). Digi XBee Python library, published by Digi International Inc. company, was a set of useful functions for Digi XBee allows us to eliminate the need for writing codes from scratch. We used Python programming language to control XBee modules, manipulate and visualize data.

To supply radio interference, we used a Buffalo AirStation WBR-G54 router to generate WiFi radio waves at frequency 2.4GHz. The WiFi router was placed next to receiver XBee module to interfere IEEE 802.15.4 signals.

Figure 5 shows our testing system visually.

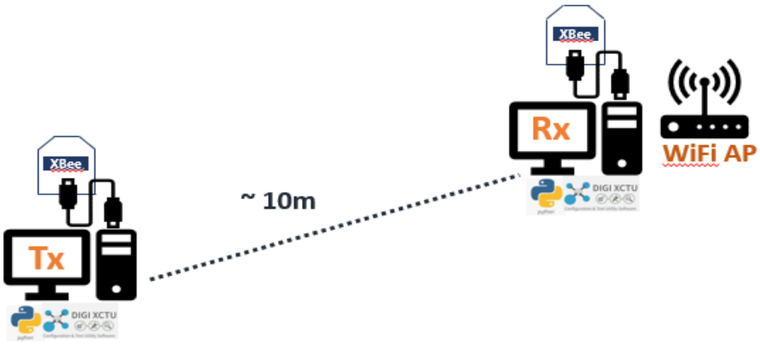


Figure 5. Testing system.

3.2 Testing environment: Technobothnia

Technobothnia is a complex laboratory unit with wide range of equipment for research and education: Automation and IoT laboratories, Electrical laboratories, Virtual and Augmented Reality laboratory, Construction Engineering laboratory, Information Technology laboratories, Energy Technology and Smart Grid laboratories, Mechanical Engineering laboratories, Environmental Laboratory. Technobothnia is an ideal environment simulating an industrial workplace. In Technobothnia, it is not difficult to find sources of wireless interference: circuit breakers, heavy-duty machineries such as CNC, motors, transformers, high-voltage electricity converters, metallic panels/fences, robots, industrial vehicles, and many WiFi networks at 2.4GHz detected (in Figure 6). Figure 7 shows PLC control panel in Technobothnia.

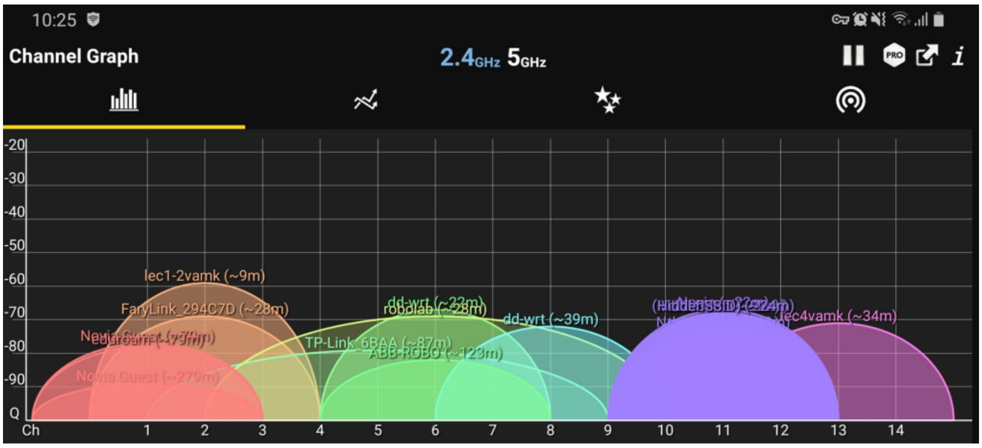


Figure 6. Wifi networks detected in Technobothnia.

Figure 7. Part of Technobothnia.

3.3 XBee S1 802.15.4 RF module

Two XBee S1 802.15.4 RF modules (in Figure 8) were used to perform testing, they are manufactured by Digi International Inc. with part number marking XB24-AWI-001 representing for Digi XBee S1 802.15.4 low-power module with wire antenna.

Figure 8. Digi XBee 802.15.4 RF Module.

Some key features of Digi XBee 802.15.4 are emphasized by the manufacturer: /11/

- It is a simple and ideal RF communication module which can be easily applied to various applications with no configuration needed.

- It supports communication in both peer-to-peer and star network topology.
- It was deployed for global 2.4GHz frequency band.
- The module integrated low-power sleep modes help to reduce power consumption.
- Multiple antenna options are provided: chip, wire whip, U.FL, and RPSMA.

The technical specifications of the module are described in Table 3 /11/.

Table 3. Digi XBee S1 802.15.4 module Technical Specifications.

SPECIFICATIONS	**DESCRIPTIONS**
RF DATA RATE	250 kbps
INDOOR/URBAN RANGE	30m
OUTDOOR/RF LINE-OF-SIGHT RANGE	100m
TRANSMIT POWER	1 mW (+0 dBm)
RECEIVER SENSITIVITY (1% PER)	-92 dBm
DIGI HARDWARE	S1
TRANSCEIVER CHIPSET	Freescale MC13212
SERIAL DATA INTERFACE	3.3V CMOS UART
CONFIGURATION METHOD	API or AT Commands, local or over-the-air
FREQUENCY BAND	2.4 GHz
INTERFERENCE IMMUNITY	DSSS (Direct Sequence Spread Spectrum)
SERIAL DATA RATE	1200 bps – 250 kbps
ENCRYPTION	128-bit AES
RELIABLE PACKET DELIVERY	Retries/Acknowledgements
IDS AND CHANNELS	PAN ID, 64-bit IEEE MAC, 16 Channels
SUPPLY VOLTAGE	2.8 – 3.4 VDC
PART NUMBER: XB24-AWI-001	Digi XBee S1 802.15.4 low-power module with wire antenna

3.4 XBee USB Adapter Board

To connect XBee modules to computers by USB interfaces, we have used SparkFun XBee Explorer USB board (in Figure 9) for an easy and simple solution. A plugging of an XBee module to the adapter board, attaching a mini-USB cable to computer gives a direct access to XBee serial pins. The FT231X USB-to-Serial converter embedded inside the board assists to translate data between computer and XBee module. Moreover, on the top side of board, there are four LEDs indicate power, RSSI, TX and RX help to debug XBee issues.

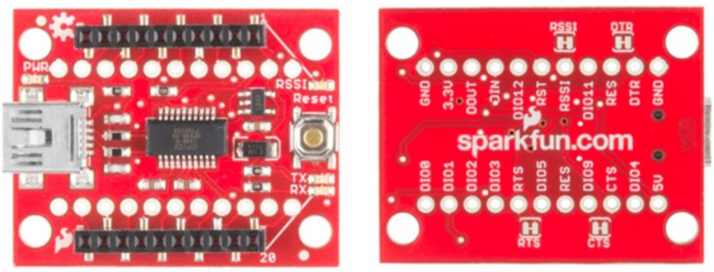


Figure 9. Font and back side of XBee USB Adapter board.

3.5 RF Terminology

3.5.1 RF Modules

A radio frequency (RF) module is a small electronic circuit designed to transmit and receive radio signals on different frequencies. The RF module we used was 802.15.4 RF module, it uses IEEE 802.15.4 specifications as a standard for its operation.

3.5.2 Radio Firmware

Radio firmware is a program stored in the memory of the module to provide control program for the device. Digi regularly releases new radio firmware versions to fix bugs and improve the module functionality. The radio firmware of the module can be updated, or the radio firmware settings can be changed by XCTU software.

3.5.3 Radio Module Operating Modes

The operating mode of RF module defines a way that a user or any microcontroller communicates with the module through the Universal Asynchronous Receiver/Transmitter (UART) or serial interface.

There are three operating modes a radio module can work in, depends on the radio firmware installed in the module:

- **Application Transparent (AT)**:

When a RF module is working in this mode, the user needs to use AT commands to control XBee. The AT command starts with "AT", then two ASCII characters for a specific command; for example, WR – Write, FR – Force Reset, DL – set Destination Address.

```
AT[ASCII command][Space (optional)][Parameter (optional)][Carriage return]
```

The structure of an AT command is:

For example, naming an identifier of a node: `ATNI MyXBee\r`

To initialize the AT operating mode on the module, a three-character command sequence needs to be sent within one second (usually "+++"). The module returns `OK\r` once the AT command mode is recognized, the command mode timer starts counting and the radio module is able to receive AT commands from that moment.

If the module does not receive a valid AT command within command mode timeout (10 seconds as default for XBee S1 802.15.4 RF module), the module automatically exists AT command mode or the AT command mode is terminated by command: `ATCN\r` /12/

- **Application Programming Interface (API)**:

This mode is an alternative AT command mode. In API operating mode, the module is communicated through a structured interface, called API data frames.

With the API mode, we are not only able to configure XBee module itself, but also are able to configure remote modules within the network. The API mode provides an easy way to detect transmission error by successful/failure status confirmation on each RF packet is sent. API mode allows to transmit data to multiple destinations or broadcast information and identify the source address of each received packet.

- **API escaped operating mode**:

It is similar to API mode except some specific data bytes of API frames must be escaped when working in API escaped operating mode.

Data bytes are escaped in API escaped operating mode are:

- 0x7E: Frame delimiter
- 0x7D: Escape
- 0x11: XON
- 0x13: XOFF

The general idea of "escaping" comes from a real case. As we know that the data unit transmitted through MAC layer is a frame. The length of a frame varies depending on its contents and settings. Supposing a receiver is receiving a stream of bytes from the serial port, how it can detect where the boundary of a frame is. 0x7E – Frame delimiter byte solves that question. A frame delimiter is a flag denoting the beginning of a frame and a frame ends before the next frame delimiter. Then another question of how system works if 0x7E appears as a piece of data (not a frame delimiter) is raised. That is the idea of escape byte (ESC). Once 0x7E is flagged to begin a frame, an ESC byte is added before 0x7E, while ESC byte is not inserted if 0x7E is in the middle of a frame. When the receiver sees an ESC byte in its stream, it knows that 0x7E is not inserted into the actual data received. In a similar way, if we want an ESC be a part of our data, another ESC is prepended to it; the receiver will ignore the first one and keep the second one in data stream.

The API escaped operating mode escapes special characters within API frames, while API non-escaped operating mode relies on the start delimiter and length bytes to distinguish API frames. Escaping mode increases the data transparency during transmission.

3.5.4 API Frames

The API frame is the structured data transmitted through radio modules serial interface when they are configured in API or API escaped operating mode. Figure 10 depicts the structure of an API frame and the description of each element in API frame can be found in Table 4:

Figure 10. Structure of an API frame.

Table 4. Explanation of element in an API frame.

Field	**Description**
Start Delimiter	The first byte of a frame indicates the beginning of a data frame. It is easy to detect a new incoming frame because its value is always “0x7E”.
Length	It is two-byte value specifies total length of bytes in frame data field. It does not count Start Delimiter, Length and Checksum bytes in its value.
Frame data	Contains the API identifier and API identifier-specific data. API identifier is equal 1 for non-escape and 2 for escape operating mode. API identifier-specific data depends on API identifier.
Checksum	The last byte of the frame. It is for data integrity test and calculated by taking hash sum of API frame bytes before checksum (it excludes Start Delimiter, two bytes of length in checksum)

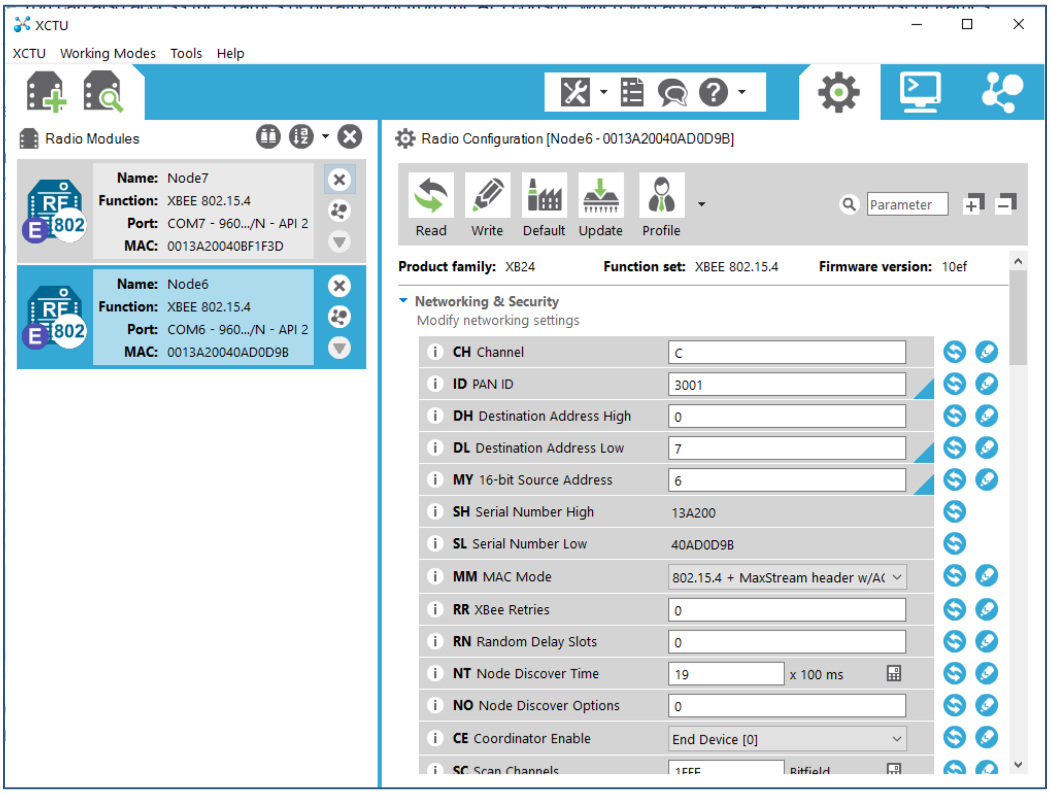
In API escape mode, Start Delimiter, Length and Checksum fields are escaped. /13/

3.6 XCTU Software

XCTU is a free multi-platform application allowing developers to interact easily with the Digi RF module. XCTU provides a graphical interface with a set of tools embedded: frames generator, frames interpreter, recovery, load console session, range test, firmware explorer. In XCTU, the user can communicate with radio devices by both AT, API and API escape operating modes.

During our experiment, XCTU was used in a few first phases: resetting manufacturing module settings, updating module firmware, initializing XBee configurations, testing XBee connections and exploring API frames in XBee using API escape operation mode.

Figure 11 captures part of XBee configurations on XCTU software.



Name: Node7
Function: XBEE 802.15.4
Port: COM7 - 960.../N - API 2
MAC: 0013A20040BF1F3D

Name: Node6
Function: XBEE 802.15.4
Port: COM6 - 960.../N - API 2
MAC: 0013A20040AD0D9B

Radio Configuration [Node6 - 0013A20040AD0D9B]

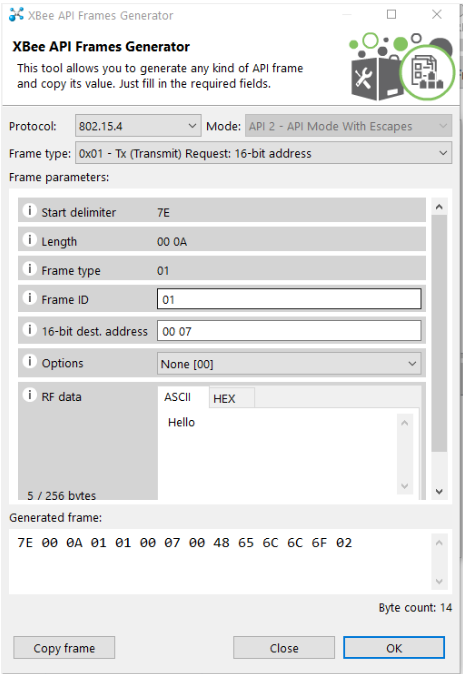
Product family: XB24 **Function set:** XBEE 802.15.4 **Firmware version:** 10ef

Networking & Security
Modify networking settings

CH Channel: C
ID PAN ID: 3001
DH Destination Address High: 0
DL Destination Address Low: 7
MY 16-bit Source Address: 6
SH Serial Number High: 13A200
SL Serial Number Low: 40AD0D9B
MM MAC Mode: 802.15.4 + MaxStream header w/AC
RR XBee Retries: 0
RN Random Delay Slots: 0
NT Node Discover Time: 19 x 100 ms
NO Node Discover Options: 0
CE Coordinator Enable: End Device [0]
SC Scan Channels: 1FFF Bitfield

Figure 11. XBee module configuration in XCTU.

Figure 12 illustrates how an API frame is generated by XCTU Generator Tool. Notice that we are using 802.15.4 protocol, API escape operating mode, and 16-bit address to communicate between XBees.



XBee API Frames Generator

This tool allows you to generate any kind of API frame and copy its value. Just fill in the required fields.

Protocol: 802.15.4 Mode: API 2 - API Mode With Escapes

Frame type: 0x01 - Tx (Transmit) Request: 16-bit address

Frame parameters:

Start delimiter	7E
Length	00 0A
Frame type	01
Frame ID	01
16-bit dest. address	00 07
Options	None [00]
RF data (ASCII)	Hello

5 / 256 bytes

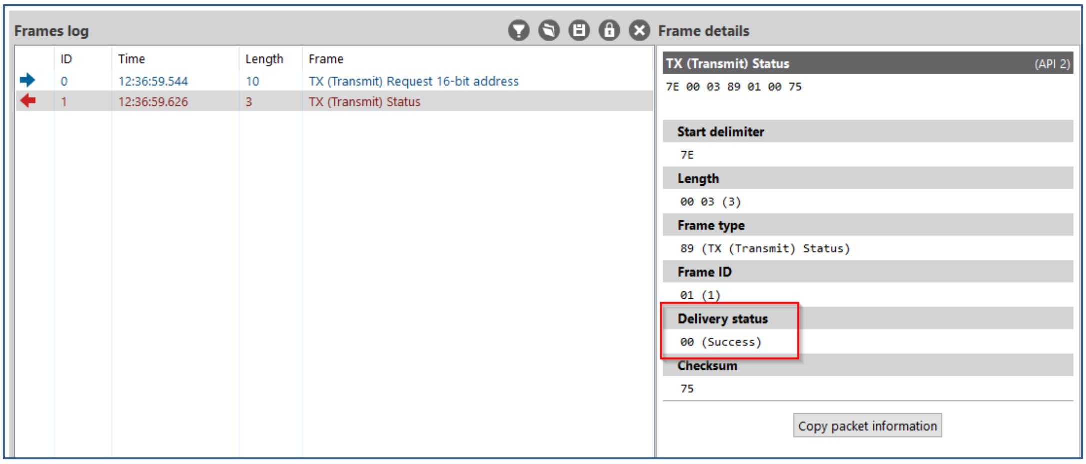
Generated frame:

```
7E 00 0A 01 01 00 07 00 48 65 6C 6C 6F 02
```

Byte count: 14

Figure 12. API Frame generator tool in XCTU.

As mentioned earlier, one of the advantages of XCTU software is a message of transmission status is delivered to sender once a packet is sent. This can be seen in Figure 13.



ID	Time	Length	Frame
0	12:36:59.544	10	TX (Transmit) Request 16-bit address
1	12:36:59.626	3	TX (Transmit) Status

TX (Transmit) Status (API 2)

```
7E 00 03 89 01 00 75
```

Start delimiter
7E

Length
00 03 (3)

Frame type
89 (TX (Transmit) Status)

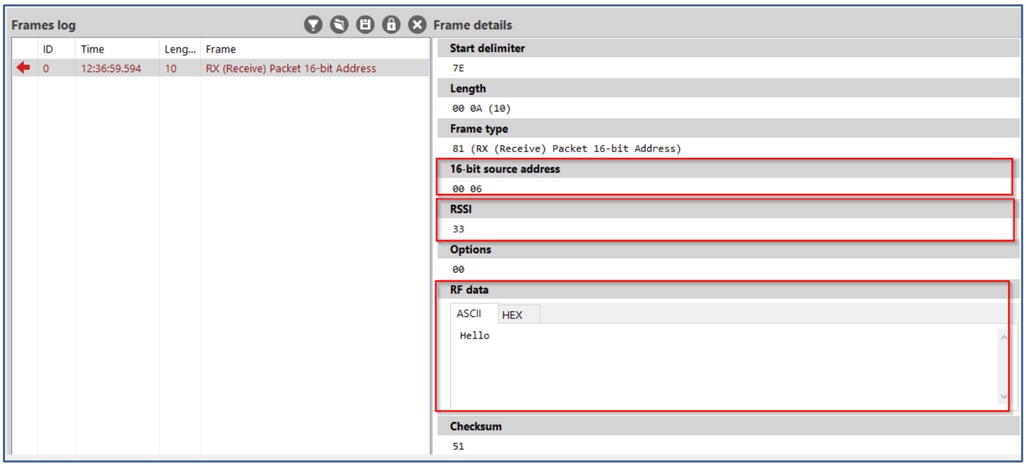
Frame ID
01 (1)

Delivery status
00 (Success)

Checksum
75

Figure 13. TX Transmit status frame details.

XCTU provides a convenient interface to explore what information is included in a frame, such as source address, RSSI, RF data, and so on. Figure 14 is a detail of a sample received frame.



ID	Time	Leng...	Frame
0	12:36:59.594	10	RX (Receive) Packet 16-bit Address

```
Start delimiter
7E
Length
00 0A (10)
Frame type
81 (RX (Receive) Packet 16-bit Address)
16-bit source address
00 06
RSSI
33
Options
00
RF data
Hello
Checksum
51
```

Figure 14. RX Frame details.

3.7 Digi XBee Python Library

Despite a friendly and convenient interface of XCTU, sending 1000 packets in a certain time period is a complicated task that XCTU cannot handle, therefore we chose the Python scripts as the alternative working method. The Digi XBee Python Library is a set of functions pre-written specially for Digi XBee RF modules. It helps to save time without needing to write everything from scratch. For example, instead of having to manually construct a frame to send, it is convenient to use `send_data` function in XBee Python Library and only need to provide the message content as a function parameter; or we used `wait_for_frame` function to read whole frame at once instead of reading each data byte then re-construct frames from received data bytes.

In order to use XBee Python Library, the following software components are required: Python 3.6 or above, PySerial 3, XBee Python library software. The installation instruction is not relevant here.

4 EXPERIMENT IMPLEMENTATION

For the experiment, we programmed the transmitter XBee to send 1000 packets at three different time intervals (0.1, 1 and 10 seconds) and the receiver to collect all necessary information from its side. All 16 channels of IEEE 802.15.4 were inspected. In this chapter, we describe steps of our experiment from setup to data manipulation.

4.1 Setup IEEE 802.15.4 Network Connection

Two computers with Digi XBee 802.15.4 S1 module attached were setup approximately 10 meters of distance away from each other in the Technobothnia laboratory, no physical obstacles obstructed their transfer path.

In order to transmit data wirelessly between two radio XBee modules, they were configured to communicate in the same frequency or channel (defined by CH parameter), be in the same network (defined by PAN ID parameter), the destination address of transmitter was the receiver's address and converse. We granted a name for each XBee for identification. Table 5 summarizes required XBees' parameter settings, while remaining parameters can be left as default. It is convenient to apply these settings by the XCTU software.

Table 5. Parameter configuration for XBee modules.

Parameters	**Transmitter XBee**	**Receiver XBee**
PAN ID (0x)	3001	3001
Channel (0x)	B	B
Node Identifier	Node6	Node7
16-bit Source Address (0x)	6	7
Destination Address High (0x)	0	0
Destination Address Low (0x)	7	6
Interface Data Rate	9600	9600
Parity	No Parity	No Parity
Data bits	8	8
Stop bits	1	1

Flow Control	None	None
Coordinator Enable	End Device	End Device
API Enable	API enabled w/PPP	API enabled w/PPP

4.2 Sending and Receiving Data

In order to have a better observation of IEEE 802.15.4 performance, we firstly performed experiments in a normal environment, which did not have any major WiFi interference.

To test packet delivery ratio, a 12-byte packet is generated, which is within the limit of 127 bytes. The receiver opened its serial port and started to listen signal/data coming, it kept the opening state for a timeout (will be described later). The transmitter also opened its serial port and began sending 1000 packets. Each packet was sent with only a sequence number (from 0 to 999) in its content and separated after a time interval, there were three different time intervals we did tests: 0.1 seconds, 1 second and 10 seconds. The transmitter continuously performed sending; after 1000 packets were sent, the transmitter sent 20 END messages (message with “END” as its content, the amount of END messages will be explained later) to inform the receiver to close connection. Figures 15, 16 show flowcharts at Transmitter (Tx) and Receiver (Rx), respectively.

Data was collected at receiver includes: the number of packets received (in 1000 packets sent), RSSI, time when a packet was received and what packets were lost during the transmission.

Tests were repeated to all 16 channels of IEEE 802.15.4. Transmitter and receiver’s channels are able to be changed by XCTU software or by Python script (it’s in my case).

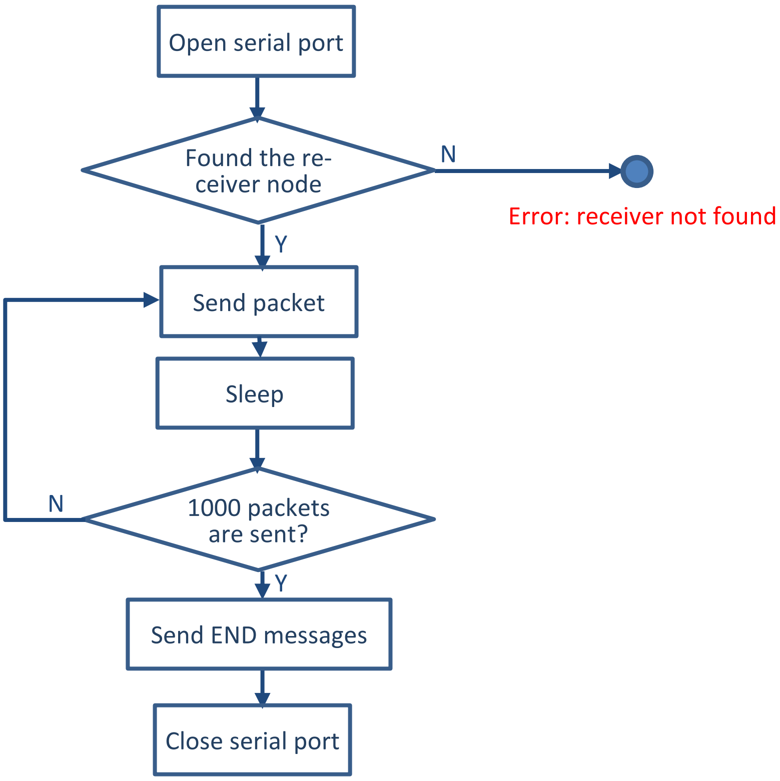


Figure 15. Flow chart at Tx side.

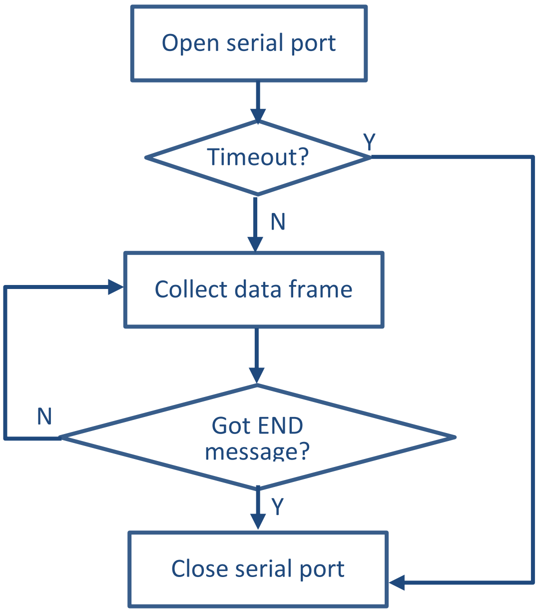
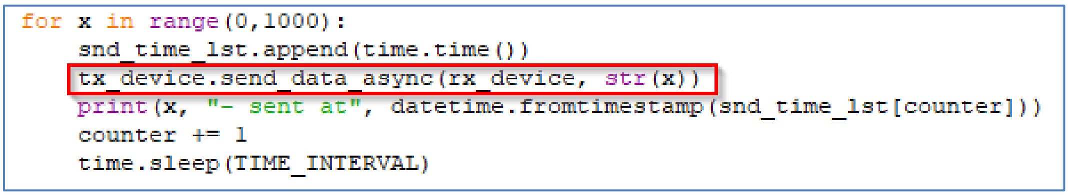


Figure 16. Flowchart at Rx side.

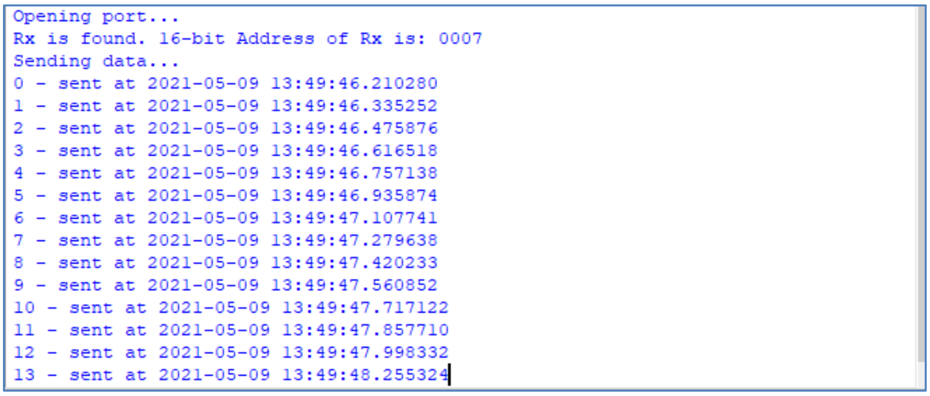
4.3 Highlighted Functions at Tx.

Firstly, the sending method used needs to be explained. XBee Python Library offers two methods to send data from a local XBee, which is referred as ‘transmitter XBee’ in this case, to a remote XBee, known as ‘receiver XBee’: they are ***synchronous*** and ***asynchronous*** operations. Both these methods transmit data as API frames. The difference between these sending methods is the state of transmitter device during data transmission: the synchronous method blocks the device for waiting a transmit status confirmation (ACK message) until it is received, or it reaches the timeout; while the asynchronous method allows transmitter continuously to send data without a successful transmission acknowledgement. Due to the characteristic of this project, we will measure the number packets lost during transmission on the receiver side, hence, the asynchronous sending method was chosen, as can be seen in Figure 17. Figure 18 captures part of screen when packets are sent from Tx.



```
for x in range(0,1000):
    snd_time_lst.append(time.time())
    tx_device.send_data_async(rx_device, str(x))
    print(x, "- sent at", datetime.fromtimestamp(snd_time_lst[counter]))
    counter += 1
    time.sleep(TIME_INTERVAL)
```

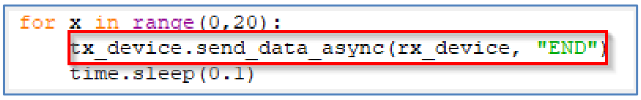
Figure 17. Asynchronous sending method.



```
Opening port...
Rx is found. 16-bit Address of Rx is: 0007
Sending data...
0 - sent at 2021-05-09 13:49:46.210280
1 - sent at 2021-05-09 13:49:46.335252
2 - sent at 2021-05-09 13:49:46.475876
3 - sent at 2021-05-09 13:49:46.616518
4 - sent at 2021-05-09 13:49:46.757138
5 - sent at 2021-05-09 13:49:46.935874
6 - sent at 2021-05-09 13:49:47.107741
7 - sent at 2021-05-09 13:49:47.279638
8 - sent at 2021-05-09 13:49:47.420233
9 - sent at 2021-05-09 13:49:47.560852
10 - sent at 2021-05-09 13:49:47.717122
11 - sent at 2021-05-09 13:49:47.857710
12 - sent at 2021-05-09 13:49:47.998332
13 - sent at 2021-05-09 13:49:48.255324
```

Figure 18. Data is sent at Rx (time interval = 0.1 second).

Secondly, the concern related to the amount of END messages are sent may be raised. According to our observation the sequence of packets was lost in each test, we have never seen 20 adjacent packets lost continuously, but it was possible to have 5 or even 10 missing packets in a sequence. These END messages do not have an impact on the number of packets received at Rx, because it is like a flag to inform the receiver that there is no waiting data to send anymore, and the END message is discarded immediately (on Rx side) since the first time Rx receives it. Figure 19 highlights process to send END messages.

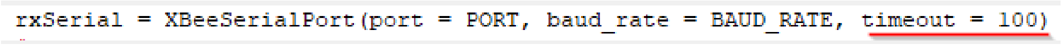


```
for x in range(0,20):
    tx_device.send_data_async(rx_device, "END")
    time.sleep(0.1)
```

Figure 19. END messages are sent.

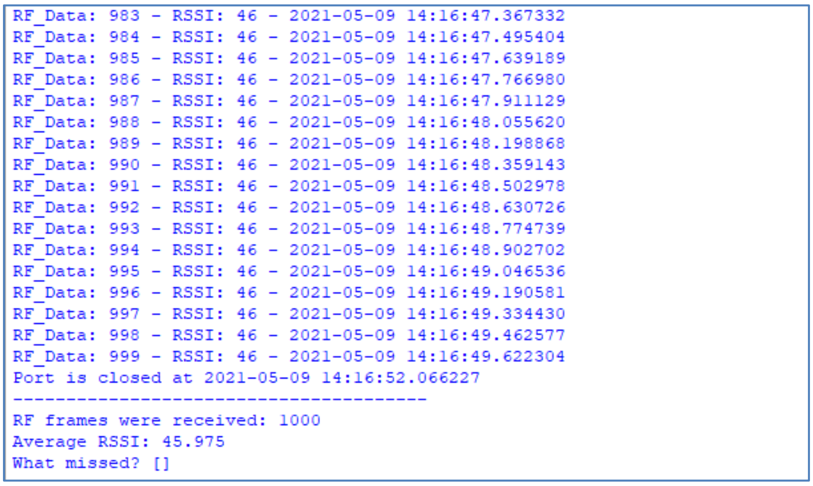
4.4 Highlighted Functions at Rx.

Unlike as in transmitter, the problem is more complicated in the receiver. The receiver serial port is expected to keep opening state to collect upcoming data packets until the whole sending process ends or there is an error in connection. These messages having an "END" flag help the receiver to recognize when the sending process ends. However, if the case in the middle of the sending process and the IEEE 802.15.4 link is dropped, in order to identify the situation, we have set a "timeout" for opening serial port as can be seen in the following code.



```
rxSerial = XBeeSerialPort(port = PORT, baud_rate = BAUD_RATE, timeout = 100)
```

The value of "timeout" depending on the Tx sending interval. A probability losing 10% packets in 1000 packets sent was assumed, meaning that 0.1 sec.*(10% * 1000) = 10 sec., 100 sec., and 1000 sec. (sending time interval = 0.1, 1 and 10 seconds, respectively). If after these timeouts, receiver does not get any data packet, the serial connection is closed automatically. Figure 20 shows data collections at Rx.



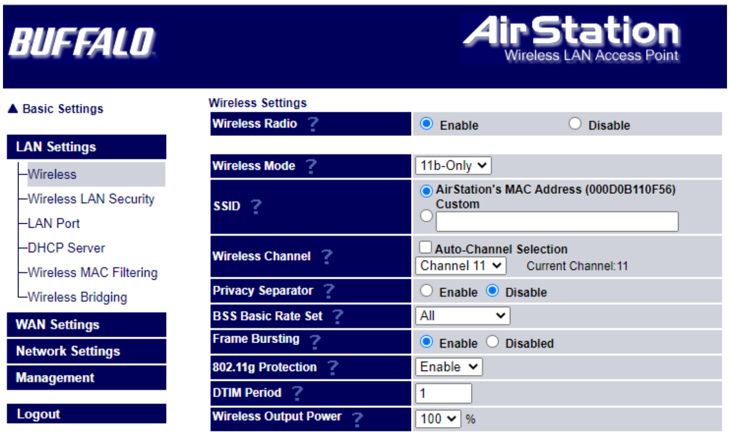
```
RF_Data: 983 - RSSI: 46 - 2021-05-09 14:16:47.367332
RF_Data: 984 - RSSI: 46 - 2021-05-09 14:16:47.495404
RF_Data: 985 - RSSI: 46 - 2021-05-09 14:16:47.639189
RF_Data: 986 - RSSI: 46 - 2021-05-09 14:16:47.766980
RF_Data: 987 - RSSI: 46 - 2021-05-09 14:16:47.911129
RF_Data: 988 - RSSI: 46 - 2021-05-09 14:16:48.055620
RF_Data: 989 - RSSI: 46 - 2021-05-09 14:16:48.198868
RF_Data: 990 - RSSI: 46 - 2021-05-09 14:16:48.359143
RF_Data: 991 - RSSI: 46 - 2021-05-09 14:16:48.502978
RF_Data: 992 - RSSI: 46 - 2021-05-09 14:16:48.630726
RF_Data: 993 - RSSI: 46 - 2021-05-09 14:16:48.774739
RF_Data: 994 - RSSI: 46 - 2021-05-09 14:16:48.902702
RF_Data: 995 - RSSI: 46 - 2021-05-09 14:16:49.046536
RF_Data: 996 - RSSI: 46 - 2021-05-09 14:16:49.190581
RF_Data: 997 - RSSI: 46 - 2021-05-09 14:16:49.334430
RF_Data: 998 - RSSI: 46 - 2021-05-09 14:16:49.462577
RF_Data: 999 - RSSI: 46 - 2021-05-09 14:16:49.622304
Port is closed at 2021-05-09 14:16:52.066227
----------------------------------------
RF frames were received: 1000
Average RSSI: 45.975
What missed? []
```

Figure 20. Data collected at Rx.

4.5 Test in WiFi Interference Environment

WiFi 2.4GHz (802.11b) and any applications operating on a frequency band of 2.4GHz are significant competitors to IEEE 802.15.4 connection.

A G54 Buffalo AirStation wireless router was used to generate WiFi signals at channel 11 (as configuration in Figure 21) to interfere IEEE 802.15.4 connectivity.



Wireless Settings

Wireless Radio ?	● Enable ○ Disable
Wireless Mode ?	11b-Only
SSID ?	● AirStation's MAC Address (000D0B110F56) ○ Custom
Wireless Channel ?	☐ Auto-Channel Selection Channel 11 Current Channel:11
Privacy Separator ?	○ Enable ● Disable
BSS Basic Rate Set ?	All
Frame Bursting ?	● Enable ○ Disabled
802.11g Protection ?	Enable
DTIM Period ?	1
Wireless Output Power ?	100 %

Figure 21. 802.11b and channel 11 was configured on wireless router.

The router was placed close to the XBee receiver. Numerous continuous 'ping -t -l 65000 [router_ip]' requests were sent from a normal PC (does not require any software installed) to the router to cause busy traffic on WiFi network. We performed the same tests as were done in No WiFi interference environment and in all channels of IEEE 802.15.4.

4.6 Data Cleaning and Analysis

The data collected from the receiver were arranged in Microsoft Excel files, which then was analyzed. The data was not always good; therefore, a cleaning task was needed. Firstly, we removed tests result of receiving below 400 packets; they could occur due to a damaged, malfunctioning sensor or a connection error. Secondly, we considered testing results of more than 400 received packets. When filtering by the same testing channel and the same sending time interval, if there was a big difference with other testing results, the investigating result was eliminated from the data collection. For example, there was a test where only 400 packets were received at channel 0x0B – the time interval was 0.1 second while other tests in the same channel, the same interval both received more than 900 packets, then this result was not accepted. However, if a test result of 400 packets received is compared with others (same testing channel, same interval) which results around 600 to 750 packets received, the examining result might be acceptable.

After the data had been cleaned and arranged into spreadsheets, they were plotted into graph by Pandas and Matplotlib.

5 ANALYSIS AND OUTCOMES

This chapter presents our data visualization and analysis results based on 313 tests done. 71 tests were performed in the environment not interfered by WiFi and 242 were performed in a high WiFi interference environment. We evaluate IEEE 802.15.4 performance in 4 aspects:

- The number of packets Rx received.
- RSSI.
- Time when packets received.
- Sequence of packets are lost.

5.1 802.15.4 Network quality Affected by 802.11 Interference

As shown in Figure 1 describing how 802.11b and 802.15.4 channels allocate in the ISM frequency band 2.4GHz, there are overlaps between channels of these two networks. In our case, we have chosen Channel 11 of 802.11b to cause interferences to IEEE 802.15.4 connectivity, meaning that the connection quality in those 802.15.4 channels overlapped by WiFi channel 11 probably were impacted.

The term “No-WiFi” will be used to differentiate two case studies: 802.15.4 tests with strong WiFi interference and tests without our WiFi setup.

Figures 22, 23, 24 give a visual comparison of the performance of IEEE 802.15.4 network in two situations: with and without strong WiFi interference and in different sending intervals.

Each dot in the graph presents an individual test result: green dots are test results in the WiFi interference environment and red dots are test results in the No-WiFi interference environment; the dots are darker; the test results are closer. Lines in the graph present for the ‘mean’ value of number packets received in each channel. Similarly, green colour is for the WiFi interfered environment and red one is for the No-WiFi environment.

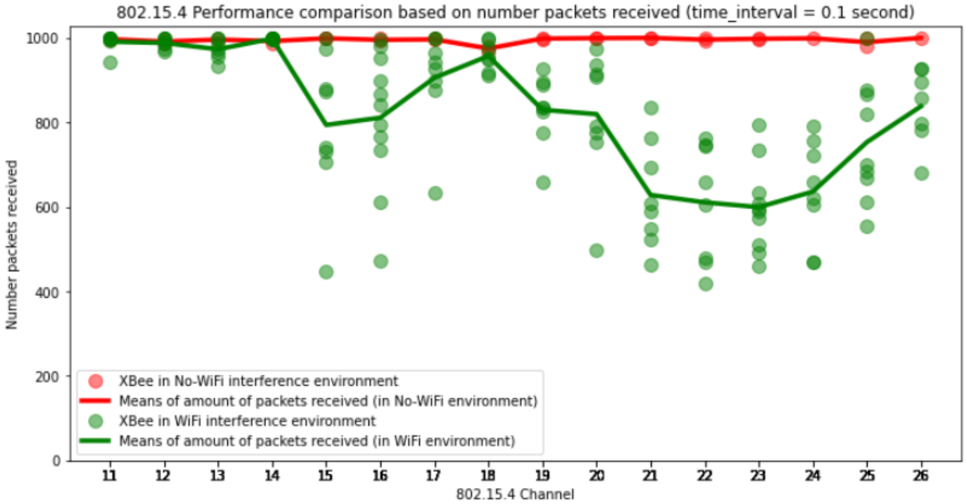


Figure 22. 802.15.4 performance comparison in two environments (TI = 0.1 sec.)

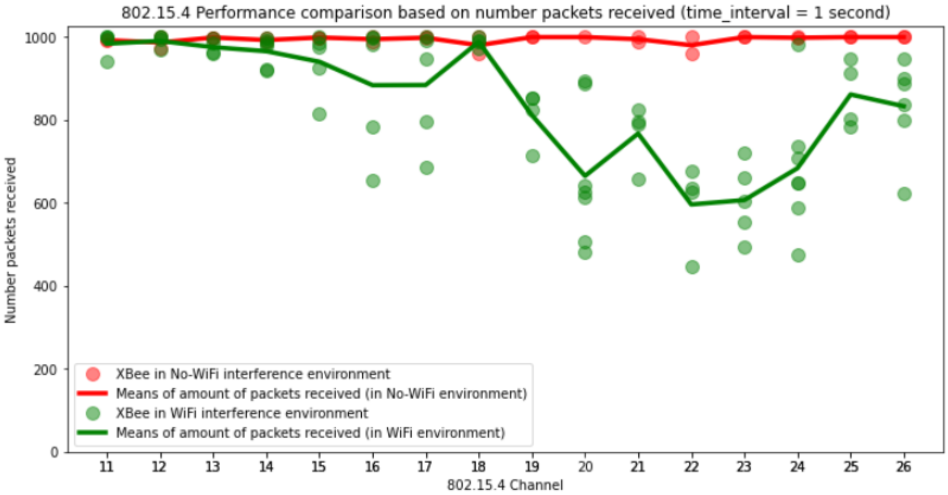


Figure 23. 802.15.4 performance comparison in two environments (TI = 1 sec.)

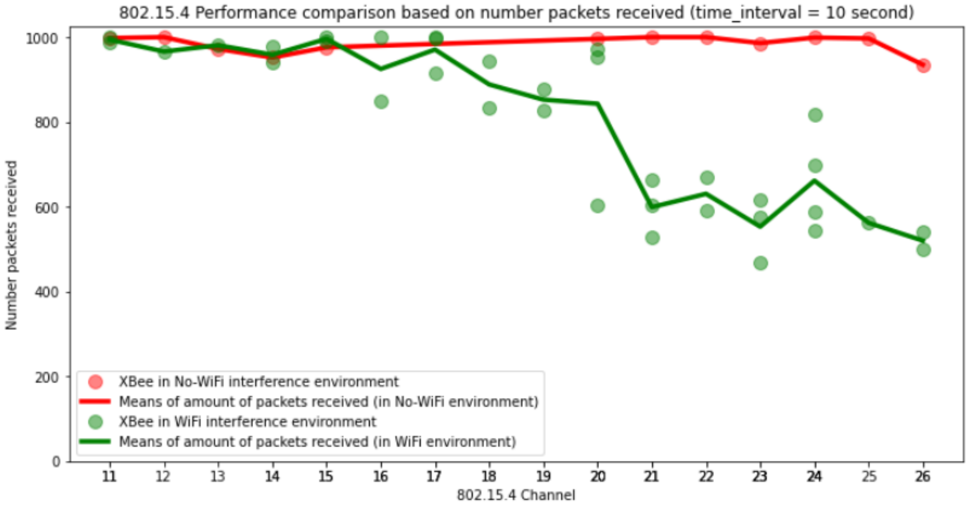


Figure 24. 802.15.4 performance comparison in two environments (TI = 10 sec.)

Looking at red dots and red lines in these figures, even if we did not setup a WiFi as a major source of interference to 802.15.4 network, 802.15.4 was still affected by surrounding noises, causing it not always receiving 1000/1000 packets sent. However, generally, all channels received over 950/1000 packets sent.

It is distinctive to see results (in green colour) when a WiFi router (802.11b) was placed next to the receiver XBee. Much data was lost during the transmission and the performance in channels was unstable.

The network quality gradually reduces when going from channel 11 to 26 of 802.15.4. Channels performing best were channels 11, 12, 13, 14, while the worst channels were 21, 22, 23, 24 which were fully covered by WiFi channel 11; many packets were lost during transmission, the ratio of missing data can be raised up to 50% and it is impossible to completely get 1000 packets in these channels. Partly missing data transferred can be seen in those channels partially covered or neighbour of WiFi channel 11, it was still possible to get fully or almost 1000 packets in these channels, but the performance is unstable in compared with performance in channel 11, 12, 13, 14 which are entirely uncovered and far away from WiFi channel 11.

This phenomenon can be explained by the “network congestion” concept in data networking that appears when there is too much data travelling through a node or a link than it can handle, leading to packet loss, link delay or even connection blocked effects. This bad situation can be improved by retransmission mechanism or congestion avoidance techniques.

5.2 Jitter in 802.15.4 Network

5.2.1 Jitter in Telecommunications

Jitter is an important factor to assess the link performance in computer network context, which is defined as a variation in the delay of received packets. On the sending side, packets are sent as a continuous stream with a constant interval space. However, this spacing time interval cannot remain constant when packets come to receiver due to common throughput issues, such as network congestion, connection errors, configuration errors, queued link. Practically, time when a packet received was measured and the time latency between two received adjacent packets was calculated to assess the Jitter value.

5.2.2 How to Calculate Jitter

Assuming on the Rx side, the receiver receives packets at timestamp as shown below:

Packet ID	Time of arrival
1	0
2	0.99
3	2.04
4	3.01
6	5.03
7	6.04

Notice the 5th packet is lost.

Calculating time difference between two adjacent packets:

$$0.99 - 0 = 0.99$$
$$2.04 - 0.99 = 1.05$$
$$3.01 - 2.04 = 0.97$$
$$6.04 - 5.03 = 1.01$$

List of time difference between two adjacent packets [0.99, 1.05, 0.97, 1.01]

Time difference average: $\frac{0.99+1.05+0.97+1.01}{4} = 1.005$

Calculating time difference deviation:

$$0.99 - 1.005 = -0.015$$
$$1.05 - 1.005 = 0.045$$
$$0.97 - 1.005 = -0.035$$
$$1.01 - 1.005 = 0.005$$

Time difference deviation: [-0.015, 0.045, -0.035, 0.005]

RMS Jitter is calculated as following equation:

$$Jitter_{RMS} = \sqrt{\frac{(-0.015)^2 + (0.045)^2 + (-0.035)^2 + (0.005)^2}{4}} = 0.02958$$

5.2.3 Jitter Analysis

If there is a difference in performance between 802.15.4 channels when comparing by the number packets received (Figure 25), there is not much difference between channels comparing by Jitter value, as in Figure 26.

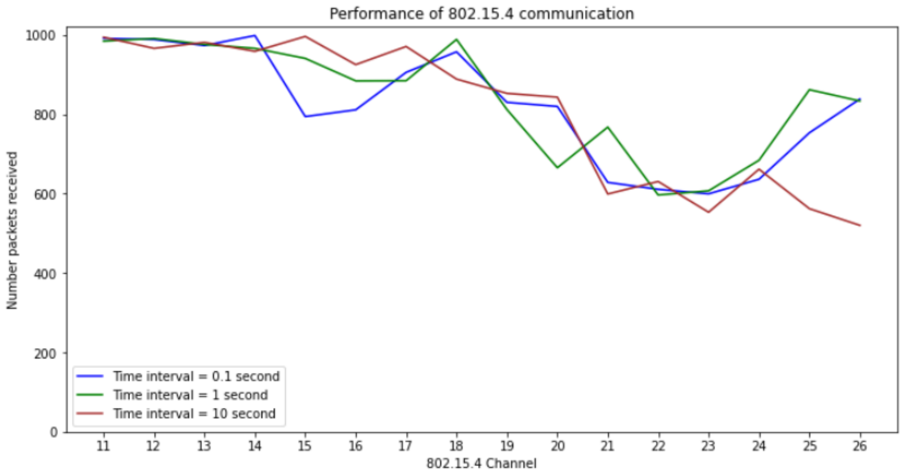


Figure 25. 802.15.4 Performance presented by 'mean' of number packets received.

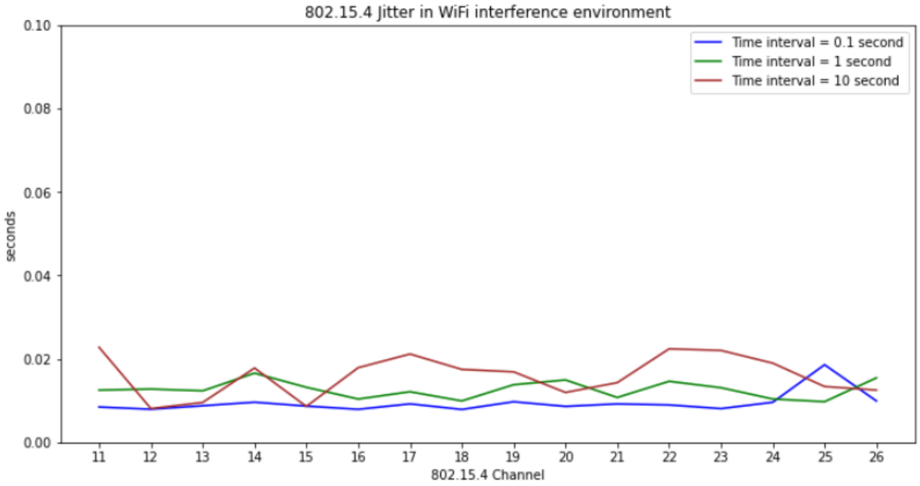


Figure 26. Performance of 802.15.4 network presented by 'RMS Jitter'.

Jitter through 802.15.4 channels does not vary much, it falls in range from 0.007 to 0.023 seconds (or 7 to 23 milliseconds). If looking at the Jitter graph in an environment not interfered by WiFi in Figure 27, generally, Jitter is in the same range, therefore, it is fair to say that 802.11 interference does not cause latency impact on 802.15.4. This might be predicted by the CSMA/CA technique used in 802.15.4 that the device senses how busy the carrier traffic is before starting transmission to avoid collisions.

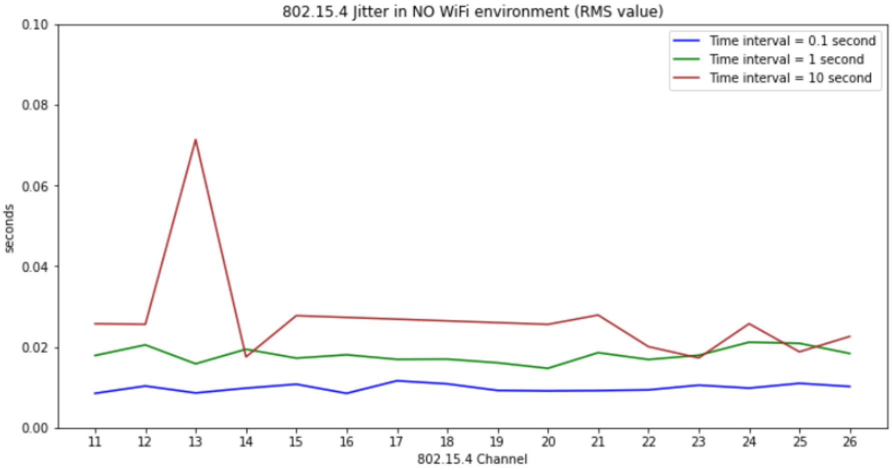


Figure 27. RMS Jitter of 802.15.4 in No Wifi interference environment.

5.3 RSSI Performance in 802.15.4 Network

RSSI in Digi XBee is presented in negative dBm, meaning that a smaller RSSI presents a stronger network link and otherwise, for instance, RSSI=50 is better than RSSI=80.

As seen earlier, the 802.11 network causes interference to the 802.15.4 connection, we possibly assume that RSSI in those channels (802.15.4) fall into the center of 802.11b channel 11 is greater than RSSI in channel 11, 12, which are not impacted much by 802.11b radio waves. Figures 28, 29, 30 give a view of 802.15.4 performance in RSSI aspect. Each color in each graph presents an individual test.

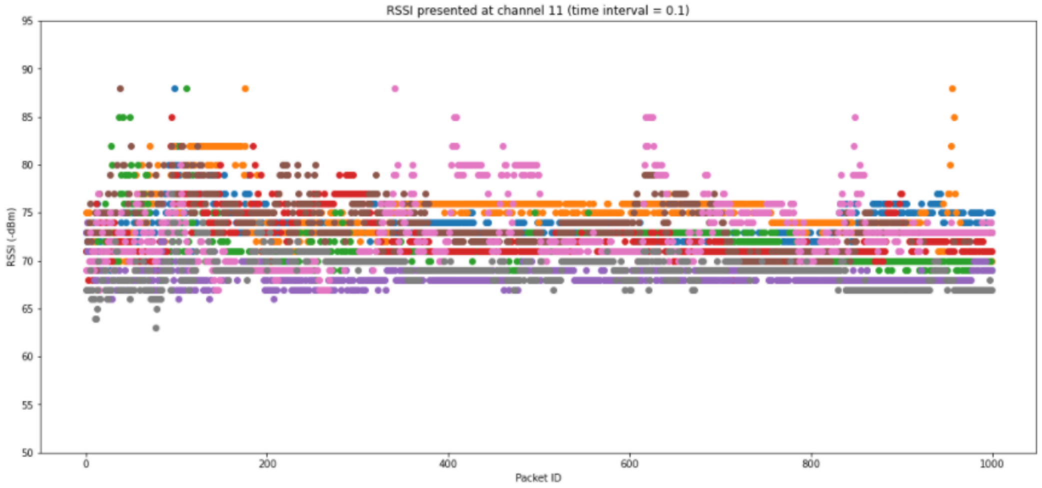


Figure 28. RSSI at channel 11 (time interval = 0.1 second).

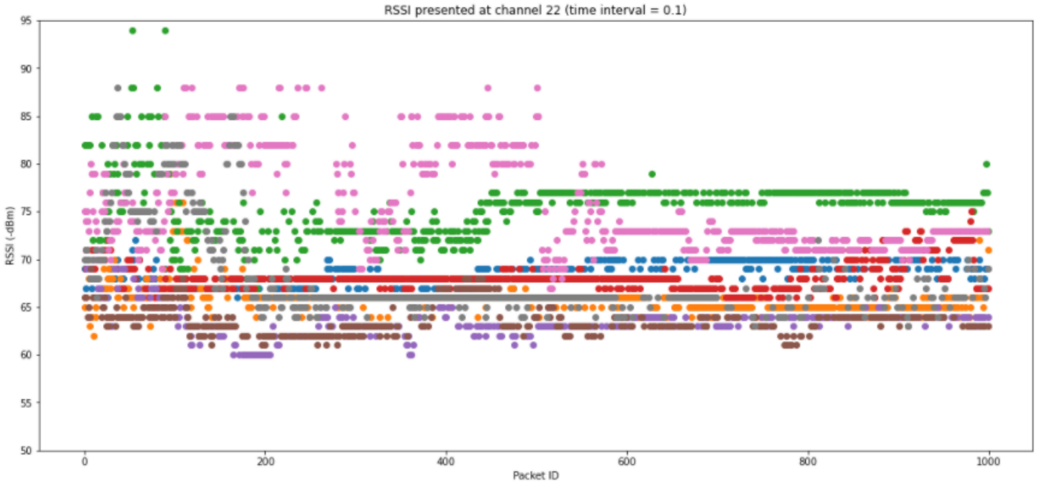


Figure 29. RSSI at channel 22 (time interval = 0.1 second).

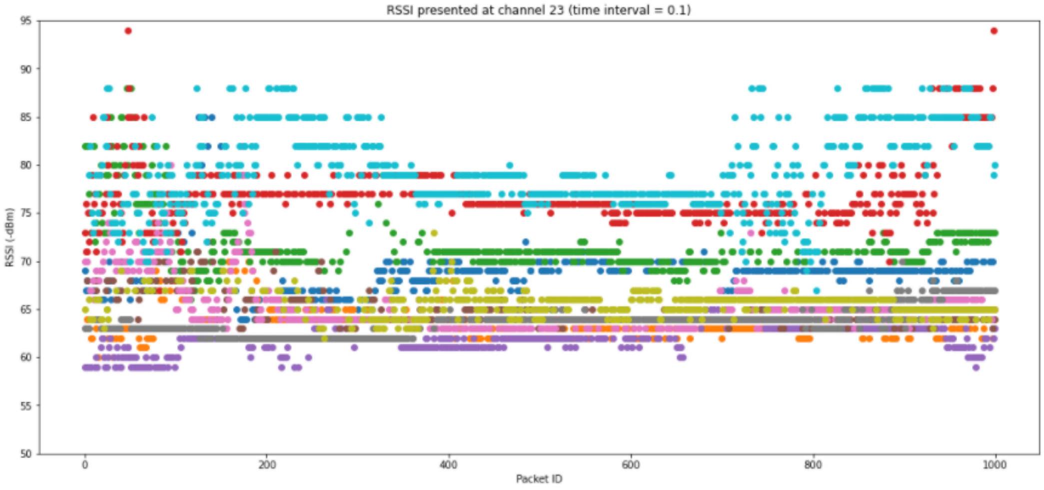


Figure 30. RSSI at channel 23 (time interval = 0.1 second).

Looking at three above graphs, some key points might be highlighted:

- Link stability: the difference of major data points in channel 11 is smaller than in channel 22 and 23 (about from -66dBm to -77dBm, -63dBm to -82dBm, and -62dBm to -85dBm in channel 11, 22, 23 respectively) meaning that the link stability in channel 11 is higher than in channel 22, 23.
- Reliability: the density of data points in channel 11 is thicker than in channel 22 and 23, meaning that in channel 22 and 23, there is more data loss possibilities.
- Receiver sensitivity: the peak RSSI in channel 11 is under -80dBm while RSSI in channel 22 and 23 raises up to -94dBm, meaning that the signal strength in channel 11 is better than in channel 22 and 23.

5.4 Data Loss in IEEE 802.15.4 Network

We have heard that surrounding noises, especially radio waves active on the same frequency band 2.4GHz affect the 802.15.4 connection quality, causing data loss during the transmission. Even in channel 11 (802.15.4), one of the best performance channels which was not impacted much by channel 11 of 802.11b network, there was still possibility of data loss. Figure 31 illustrates packets lost during the transmission in channel 11 through 8 tests.

Exploring the data loss in channel 22 (802.15.4), we see it is denser than what is seen in channel 11, as Figure 32 shows. However, we can notice that in each single test, the number packets lost, and their sequences are different. We cannot find a similar sequence of transmission loss in any channels.

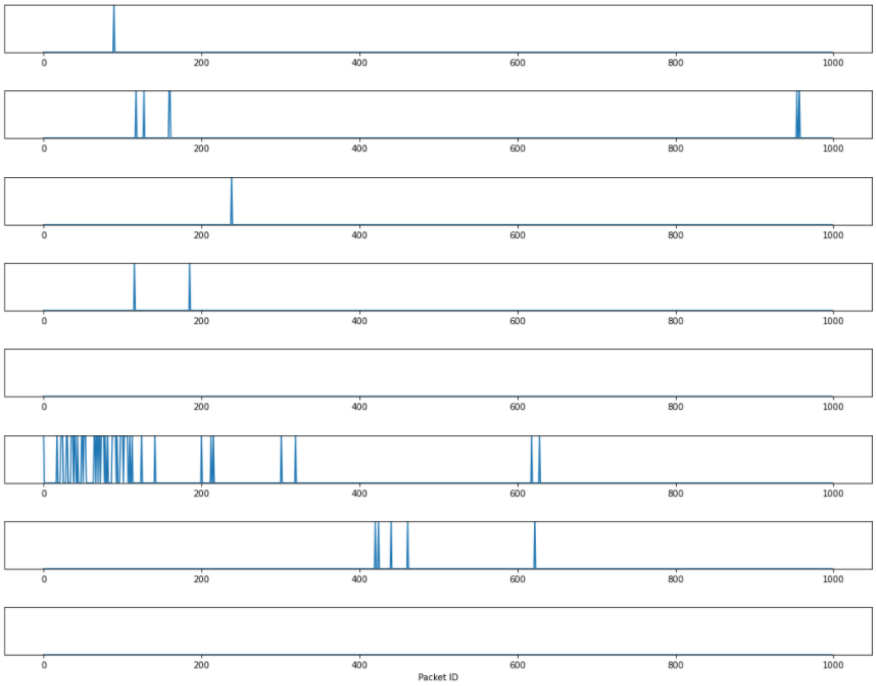


Figure 31. Data loss in channel 11 (802.15.4) - time interval = 0.1 second.

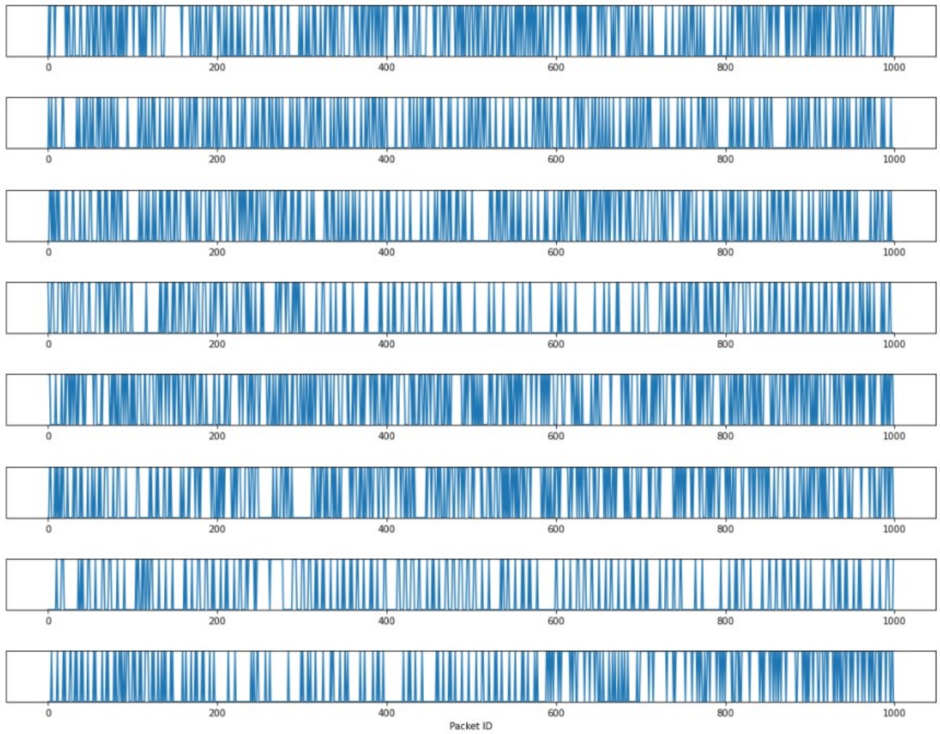


Figure 32. Data loss in channel 22 (802.15.4) - time interval = 0.1 second.

6 CONCLUSIONS

This project focuses on studying how IEEE 802.15.4 network performance in an industrial environment which consists of various radio noises. It is certain to conclude that the quality of the 802.15.4 network is impacted and decreased when there is coexistence of devices or instruments using the same radio frequency; however, time latency is kept stable. 2.4GHz IEEE 802.15.4 standard offers 16 channels allocation, hence, selecting an appropriate channel might reduce probability of data loss and increases the connection reliability. Although this thesis was done in the laboratory environment, the behavior of IEEE 802.15.4 in this simulating environment can provide predictions of issues might occur in a real industrial workplace.

The testing results will result differently depending on environment conditions, even if reproducing the same tests; moreover, the RF module functioning also bring differences. We ensure that all data used for this project was collected from our own experiments.

The analysis result may be more accurate if there are more tests performed. There is still room for further investigations based on this project idea: the WiFi channel set to channel 6 instead of channel 11 and performing test in case: one transmitter – multiple receivers.

REFERENCES

/1/ IEEE 802.15.4. Wikipedia. Accessed 5.2021. https://en.wikipedia.org/wiki/IEEE_802.15.4

/2/ Electromagnetic interference. Wikipedia. Accessed 5.2021. https://en.wikipedia.org/wiki/Electromagnetic_interference

/3/ Electronics Notes. IEEE 802.15.4 Standard: a tutorial / primer. Accessed 5.2021. https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/basics-tutorial-primer.php

/4/ Gao, C. 2019. Wireless Network. Lecture Material. VAMK University of Applied Sciences.

/5/ Carrier sense multiple access with collision avoidance. Wikipedia. Accessed 5.2021. https://en.wikipedia.org/wiki/Carrier-sense_multiple_access_with_collision_avoidance

/6/ Elahi, A. & Gschwender, A. 2009. ZigBee Wireless Sensor and Control Network. 1st ed. Indiana. Pearson Education, Inc.

/7/ Challoo, R., Oladeinde, A., Yilmazer, N., Ozcelik, S. & Challo, L. 2012. An Overview and Assessment of Wireless Technologies and Co-existence of ZigBee, Bluetooth and Wi-Fi Devices. Procedia Computer Science 12, 2012, 386-391. Elsevier B.V. Accessed 5.2021. doi:10.1016/j.procs.20122.09.091.

/8/ Wi-Fi. Wikipedia. Accessed 5.2021. https://en.wikipedia.org/wiki/Wi-Fi

/9/ PC Magazine (www.pcmag.com). 802.11 versions. Accessed 5.2021. https://www.pcmag.com/encyclopedia/term/80211-versions

/10/ Harwood, M. 2009. CompTIA Network+ N10-004 Exam Cram. 3rd ed. Pearson IT Certification.

/11/ Digi International Inc. Legacy XBee S1 802.15.4 Product Datasheet. Accessed 5.2021. https://www.digi.com/resources/library/data-sheets/ds_xbeemultipoint-modules

/12/ Digi International Inc. AT operating mode. Accessed 5.2021. https://www.digi.com/resources/documentation/digidocs/90001458-13/concept/c_at_operating_mode.htm

/13/ Digi International Inc. RF Terminology. Accessed 5.2021. https://www.digi.com/resources/documentation/digidocs/90001458-13/concept/c_concepts_terminology.htm?tocpath=RF%20concepts%20and%20terminology%7C_____0