Interference modeling for 3D simulator of sensor networks dedicated to smart cities

Umber Noreen

THESE DE DOCTORAT DE

L'UNIVERSITE
DE BRETAGNE OCCIDENTALE
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Umber NOREEN

Modélisation d'Interférence pour Simulateur 3D de Réseaux de Capteurs Dédiés aux Villes Intelligentes

Thèse présentée et soutenue à Brest, Pôle Numérique Brest Bouguen, le 20 décembre 2018

Unité de recherche : UMR 6285 Lab-STICC - UBO

Rapporteurs avant soutenance :

Congduc PHAM	Professeur, Université de Pau et des Pays de l'Adour (UPPA). France.
Cécile BELLEUDY	Maître de conférence HDR, Université Nice Sophia Antipolis, UNS. Laboratoire d'Electronique, Antennes et Télécommunications, France

Composition du Jury :

Président :	Olivier BERDER	Professeur, université de Rennes 1 (IUT de Lannion) / IRISA, France
Examinateurs :	Bernard POTTIER	Professeur, Université de Bretagne Occidentale, UMR6285 Lab-STICC
	Congduc PHAM	Professeur, Université de Pau et des Pays de l'Adour (UPPA). France.
	Cécile BELLEUDY	Maître de conférence HDR, Université Nice Sophia Antipolis, UNS.
Dir. de thèse :	Ahcène BOUNCEUR	Maître de conférence HDR, Université de Bretagne Occidentale, Lab-STICC
Co-dir. de thèse :	Laurent CLAVIER	Professeur, IMT Lille Douai, France.

Invité(s)

Alain PLANTEC	Professeur, Université de Bretagne Occidentale, France.

In memory of my late mother.
Dedicated to my father,
who gives me the greatest gift anyone could give another person,
he believes in me.

Acknowledgement

First and foremost, I would like to thank Almighty Allah for making all this possible for me for bestowing upon me uncountable blessings.

Completion of this doctoral dissertation was possible with the support of several people. I would like to express my sincere gratitude to all of them. Firstly, I would like to express my sincere gratitude to my advisors Dr. Ahcène Bounceur and Dr. Laurent Clavier for the continuous support, patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. They always made themselves available to clarify my doubts despite their busy schedules. I could not have imagined having better advisors and mentors for my Ph.D. study.

I thank my fellow labmates from Lab-STICC Brest and IRCICA Lille, for the stimulating discussions, for working together, and for all the fun we have had in the last three years. Also, I thank them for helping me during my initial days in France. Many thanks to Nicolas Araujom for helping me with all my French translations and documentation.

Evermore thanks go to the best friends I made during my stay in France: Sanaa Ghalem, Inderjeet Singh, Rahul Vyas, Yama Dixit, and Neha Rajpoot. They made my stay more memorable. Having friends like them made everything possible, and I appreciate their support.

I owe a special thanks to my family, my father Rashid Akbar, my brothers Sohaib and Shoaib who supported me and helped me throughout my life and during this study. I thank my sister-in-law Mehreen, and my cutest niece Hareem. I thank my uncle Ahmad Firdausi and my aunt Shaheen Firdausi for always being there for me during my Ph.D. I dedicate this work to you all. Ammi(mother), you are not with us, but I know you would be proud of me. Abbu(father) I do not know how to thank you enough for providing me with the opportunity to be where I am today. I love you so much.

Contents

List of Figures

List of Tables

1 Introduction

The evolving field of WSNs has an extensive range of potential applications in industry, science, transportation, civil infrastructure, and security, etc. WSNs comprised sensing (measuring), computation, and communication into a single tiny device called sensor node [1] [2]. WSNs typically consist in a large number of heterogeneous sensor devices that contain processing capability, sensor(s) and/or actuator(s), a power source (batteries and eventually some energy harvesting modules), multiple types of memory and a radio frequency (RF) based transceiver. This large number of sensors are densely deployed over a large field and inter-networked together. They monitor physical or environmental conditions that generate sensor readings and deliver them to a sink node to be further processed [2].

In a wireless sensor network composed of many spatially scattered wireless nodes, communication is constrained by various impairments such as the wireless propagation effects, network interference, and thermal noise. The impacts of signals propagation in the wireless environment include the attenuation of radiated signals with distance (also called path loss), the blocking of signals caused by large obstacles (also called shadowing), and the reception of multiple copies of the same transmitted signal (also called multipath fading). The network interference is due to accumulation of unwanted signals radiated by other transmitters from inside or outside of the network, which undesirably affects signal reception at receiver nodes in the network. The thermal noise is introduced by the receiver electronics and is usually modeled as additive white Gaussian noise (AWGN). However, in all WSN applications, connectivity is a crucial point. A sensor node should be low cost and consumes low power. In many cases, a node has to be as cheap as $1e$. A node also has to work with ultra-low power to allow life-time over 1 year or even several decades in situations where batteries cannot be recharged or replaced. However, those constraints must not degrade the reliability of the nodes.

The first WSN military applications can be traced back at least at the end of $60's$ (Operation Igloo White). For this, High-tech systems were developed to prevent supplies streaming into the battlefield. This system was considered technologically ahead of its time. The first significant academic project was probably Smart Dust, presented in the late 90's. It was proposed that smart dust would be made of micro-electro-mechanical systems (MEMS), which are very small sensors that can perform several functions. Since then, tremendous efforts have been made, on every component. The first significant trend was so called ad-hoc networks. It consists of a collection of mobile nodes, connected without any centralized structures or control. The idea was to transmit at short distances. This gave rise to the IEEE standard 802.15.4 in 2003, and one example of its well-known product is ZigBee. ZigBee was developed for personal area networks (PAN). At first glance, this short distance standard seemed to be a good solution, to reduce the power consumption.

Nowadays, wireless sensor network industry is gradually changing its interest towards LPWAN.

LPWAN technologies (SigFox, LoRa, Weightless-W, etc.) successfully propose wide area connectivity from a few to tens of kilometers for low data rate, low power, and low throughput applications. Their market is anticipated to be huge. Nearly a quarter of overall thirty billion IoT/M2M devices are assumed to be connected through the internet using LPWAN. IoT applications have fundamental requirements such as long range, low energy consumption, and cost-effectiveness. The widely used short-range or medium-range radio technologies (e.g., Bluetooth, ZigBee, WiFi, etc.) cannot assure either long-range transmission or low power consumption. Similarly, Cellular networks (e.g., 2G, 3G, and 4G) can give larger coverage, but on the cost of high energy. This niche can be filled with the emergence of low power wide area network (LPWAN). Low power wide are networks (LPWAN) is emerging technology used for wireless communications for IoT and M2M applications. Huge number of end-devices connect to the base station or gateways in the star topology. After that gateways transfer the uplink traffic to the server through IP based network [3]. LPWAN are supposed to operate in a wide range of IoT applications, but some common requirements should guide the design of a LPWAN network [4].

- Long range: Long range (5 to 30 km in different environment conditions) can help reducing the number of base stations and can reduce the deployment cost.
- Low power: Ultra low-power usage.
- Battery lifetime: Long network lifetime(upto 10 years) with small batteries to avoid costly battery change.
- Low cost: Inexpensive module.

Smart cities are considered as the most significant potential customers of LPWAN, which includes smart metering, smart grids, smart parking, optimised driving and walking routes, energy radiation measurements, measurements of nuclear power station radiation, weather adaptive street lighting, smart waste management, structural health monitoring, air pollution monitoring, water leakages monitoring, forest fire detection and so forth, (as also shown in Figure 1.1).

The difficulty remains, however, to predict how a network of things can behave. One primary concern is its reliability. Assume that you are in a wide area with a dense network of things. The first important aspect to the connectivity is the reliability of the radio channel. But this channel depends on the position of the transmitter and the receiver. It is essential to accurately take the environment into account if we want a good idea of the network reliability. The second aspect is the physical layer. Many solutions are available and offer a wide range of compromises between energy efficiency, reliability, and coverage, etc. The third aspect is interference. If MAC layers usually try to minimise collision, heterogeneity of networks and wireless connectivity makes the multiple access challenging to organize. Most of the existing solutions are based on random access, leaving it on the node to decide when to transmit. Even with the sensing mechanism, the risk of collision is high, especially in the ISM bands. Due to the scarcity of the radio spectrum, it is not completely possible for large wireless networks to communicate without interference. Other radio devices can make transmission using the same radio frequency band at the same time. Consequently, at the receiver, many undesired signals from interfering transmitters will add to the desired transmitter's signal. Generally, interference causes performance degradation of the communication networks [5].

The consequence is that if we want to know if our WSN application has a chance to be successful, we need to be able to predict the behaviour of the network. To accurately take the environment into

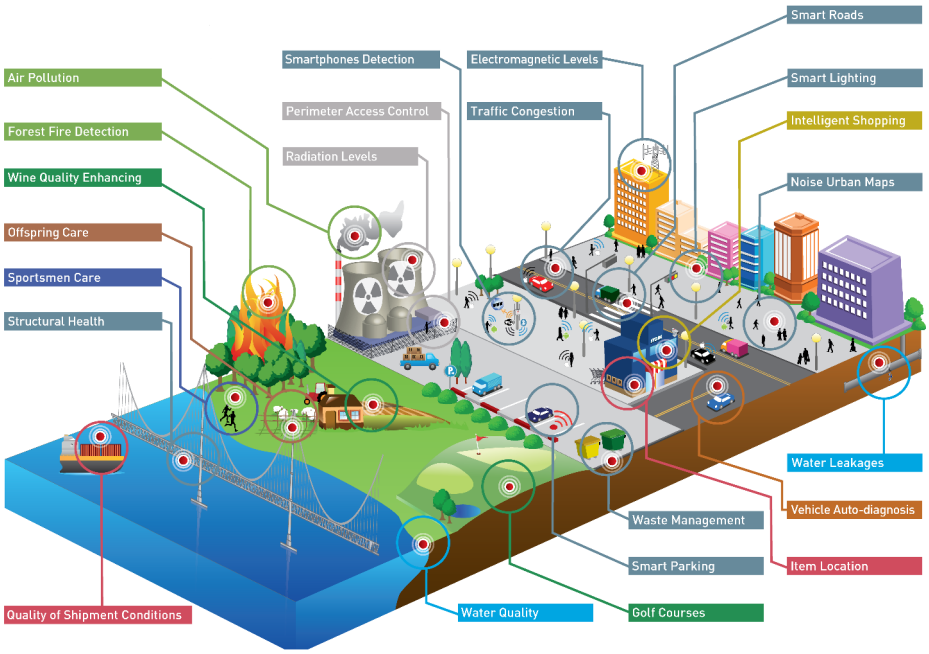


Figure 1.1: Smart city example [libelium].

account, deterministic approaches are required. But to avoid extremely long simulations, statistical approaches are also needed. Many theoretical works have addressed this problem, but many unsolved questions remain, and usually, they do not account for the specific situations encountered in real life.

1 Internet of Things (IoT)

In recent days, the Internet of Things (IoT) is gaining massive interest as the internet users are shifting from persons to things. It is envisioned to provide integration of the digital world into the physical world. It is expected to be composed of trillions of objects interacting in a heterogeneous way regarding behaviour, capabilities, and requirements. All these elements will have to send or receive (or both) information to or from the internet. Connecting everyday objects to the internet and turned them into smart objects. These smart interconnected objects are not only able to sense and gather information from the environment but also interact with and control the physical world. Most of the time this communication takes place by making use of radio frequency (RF) based wireless communications and results in spectrum overcrowding problems.

The internet of things does not comprise of a single technology. It integrates some of the already existing technologies, standards, communications and automation devices, data processing and cloud, and applications. This coordinated combination leads from the simple inter-connection of things/objects to the internet to the wide range of applications. These technologies can be stacked in three layers, which are called perception layer, network layer and application layers as shown in Figure1.2. Perception layer incorporates internet-enabled devices or objects that can per-

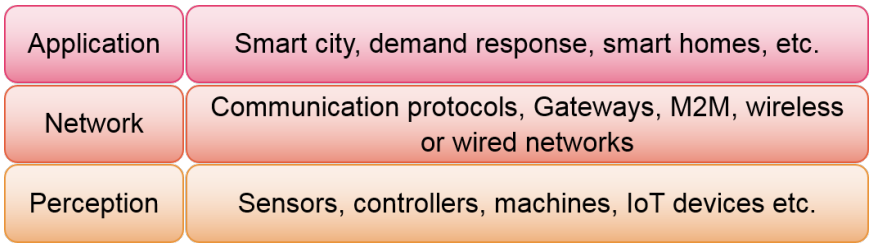


Figure 1.2: Key technologies and IoT layers.

ceive, sense, collect and exchange information with other objects through the internet. Sensors, actuators, controllers, RFIDs, GPS, etc. are a few examples of perception layer devices. Perception layer forward data to the application layer through the network layer. IoT systems use short range to long range wired or wireless technologies, such as ZigBee, Wi-Fi, LoRa, Sigfox, Cellular, Power Line Communication (PLC), etc.

The IoT covers different fields of applications [6] [7], most of them are mentioned in Figure1.1, among others:.

- Smart city,
- Smart environment,
- Domotics or home automation,
- Smart water management,
- Smart energy management,
- Smart agriculture,
- e-Health, and
- Smart farming.

As this thesis works and revolves around interference modeling for smart city applications and implementations, next section gives a breif discription about what smart cities are.

1.1 Smart Cities

As the world population is moving towards big cities, and these cities need to find new ways to endorse high service levels for citizens. The number of smart city projects are increasing across the world, and it is essential to discuss what are the goals and challenges involved to make a city smarter. As shown in Figure 1.3, according to a study done by Unicef (www.unicef.org/sowc2012/urbanmap), 70% of the world's population will live in cities, by 2050. These big cities will bring some intense challenges regarding their management, but these same issues make them into engines of new research ideas and innovations.

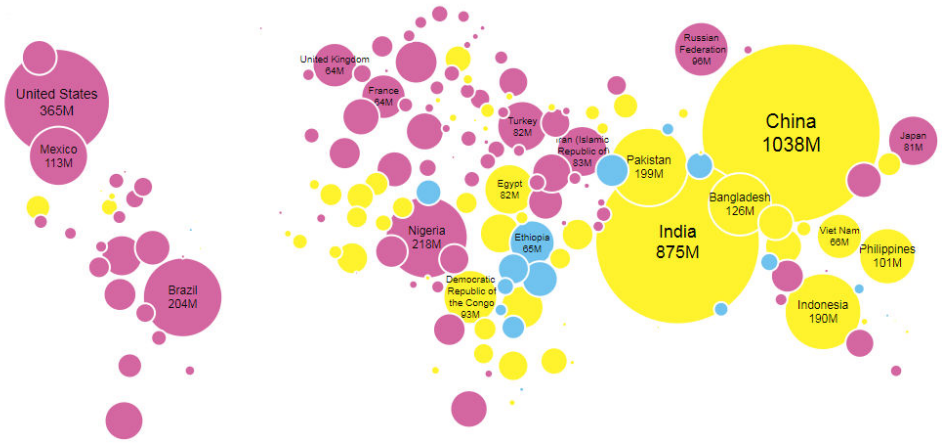


Figure 1.3: World urban population map by Unicef (www.unicef.org/sowc2012/urbanmap).

The rapid development in sensing and communication fields and the ever reducing deployment cost of WSN are allowing to embed them in masses for monitoring and controlling in the cities and make them smarter than ever. WSN has many applications for smart cities like:

- Smart parking: monitoring and managing the parking space availability.
- Urban noise: Sound monitoring in the densely populated areas.
- Structural health: Monitoring of structural material conditions of building and bridges.
- Traffic congestion: Monitoring of vehicles and pedestrian congestion to find the optimized routes.
- Smart lighting: Smart weather adaptive street lights.
- Smart waste management: Monitoring the rubbish levels in trash containers to optimize its collection routes.
- Smart highways: Smart highways with warnings signs, messages and diversion information according to unexpected events and weather conditions.

2 Wireless Sensor Network

A wireless sensor network is a self-configured and infrastructureless wireless network of nodes that can monitor environmental or physical conditions such as noise, pressure, vibration, temperature, or motion, and then take necessary decisions to control the environment. It also builds inter-connection between users and the environment [8]. A large number of end-nodes (normally equipped with sensors, actuators, controllers, transceivers and battery devices) are deployed in the monitoring area and is called a sensor field. Typically a WSN contains a large number (hundreds of thousands) of such nodes, which communicate within a network through a wireless link. An individual node in the network is usually inherently resource constrained which means they have limited storage, bandwidth an processing ability. Depending on the application requirement, these nodes can be

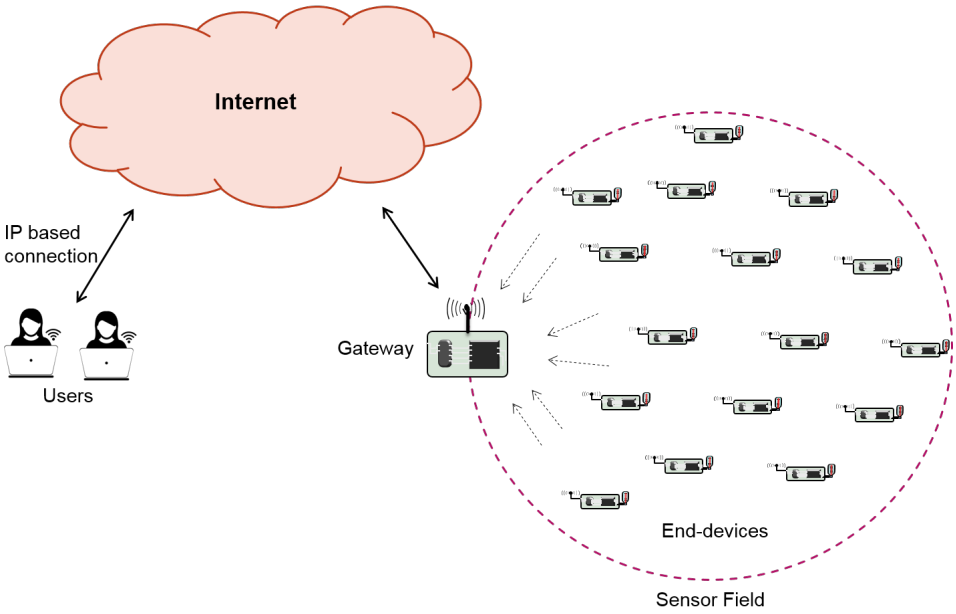


Figure 1.4: A typical Wireless Sensor Network.

static or mobile and can be categorized as continuous or event-driven. Sensor nodes can perform data collection continuously, triggered by an event, i.e., fire detector, or on demand. The end-nodes send their measured data to the collection unit called sink or gateway nodes. The gateway is used either to connect two or more dissimilar networks or an interface between network and the users. The sink or gateway node is often static; however, it can also be mobile [9].

Depending on the network topology, the data from end-device to sink might be transferred directly or using any multi-hop technique. Star (used in LoRa [10] and Sigfox [11]), mesh (used in ZigBee [12]), tree (used in RPL [13]), etc. topologies are some of the commonly used network topologies. For short range technologies (ZigBee), end-nodes also act as a router to allow multi-hop communication in the network. In this case, end-nodes should have self-organizing capabilities, in order to avoid whole network failure due to router node failure. In recent years, with the emergence of LP-WAN technologies (LoRa [10], SigFox [11], etc.), which enable long-range communication and follow star topology, complex routing algorithms can be avoided. In the star topology, end-nodes directly communicate with the gateway or base station (e.g. Cellular). A local positioning algorithm or GPS can be utilized for location and node positioning information. A typical architecture of a WSN is depicted in the Figure 1.4.

2.1 Wireless Sensor Node Architecture

WSN typically consist of large number of heterogeneous sensor devices that contain processing capability, a sensor/actuator, a power source (battery), multiple types of memory and a radio frequency (RF) based transceiver as shown in Figure 1.5.

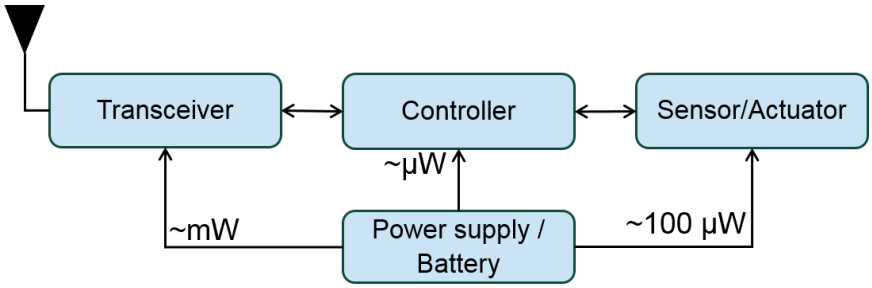


Figure 1.5: Wireless sensor node architecture.

- **Sensors/Actuators:** A node consists of different sensors or actuators to perform measurements in the network field. The type of sensors/actuators used in a node depends on the given application. The main purpose of sensor(s)/actuator(s) is to detect or measure the physical properties of a certain event and respond to it. Some examples of sensors are temperature sensor, humidity sensor, accelerometer.

- **Micro-controller:** Usually contains a microcontroller (MCU) and a small storage device. Microcontroller controls and manages all other parts of the sensor node to performs measurements and transmission of data at the right time. MCU is also responsible for the execution of any medium access control (MAC) or routing protocols with the help of internal service information. In some cases, a small memory is also attached with MCU to store the gathered data to be sent later.

- **Radio module:** Radio transceivers are used for the measured data transmission and receiving of control messages. Usually, radio modules are the most power-hungry components of a node. The lifetime of any node highly depends on the used RF module.

- **Power supply:** A sensor node is equipped with an energy source (battery). This energy resource is limited. Battery is the power supply source of a node. It is composed of an energy storage device (supercapacitor/micro battery). Sometimes it also includes energy harvesting devices (vibration energy harvesting devices, Photovoltaic (PV) cells, etc.) but it is optional since they can increase the cost of production of a node. In some cases, sensor nodes can be directly connected to the power lines. This solution can remove the energy constraint, but in this case, it reduces the number of possible applications.

To reduce energy consumption, assigned power for transmission always should remains low compared to other wireless communication systems. From the above discussion on different parts of a sensor node, one can conclude that RF modules have the maximum energy consumption. Generally, it is essential to introduce efficient power utilization techniques to regulate RF module to minimize power waists. Many RF communications related constraints affect nodes/network. These factors are briefly described in the next Section.

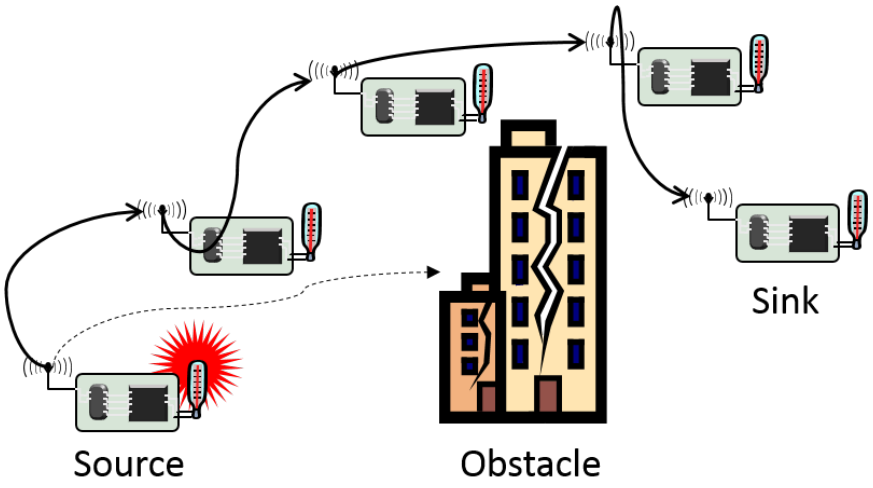


Figure 1.6: Multi-hop communication in WSN.

2.2 Wireless Sensor Network Constraints

WSN have some constraints, e.g., short radio range, low cost, limited energy, and small battery size (and difficulty to replace them in many situations). These resource constraints can impose the limitations during the application design. Moreover, some aspects of the wireless channel make wireless communication more challenging as compared to wired communication like probabilistic wireless channel behavior, limited radio range, interference from other radio devices and many more [14]. These aspects change the characteristics of the transmitted signal as it travels through the wireless channel, and can make difficult if not impossible to recover.

Multi-hop Communication

While wireless communication is a core technique in WSNs, direct communication between a sender and a receiver is faced with limitations. In particular, communication over long distances can induce high transmission power. The use of intermediate nodes as relays can reduce the total required power. Hence, for many forms of WSNs, so-called multi-hop communication is a necessary ingredient [15], which causes route management overhead. An end-node that is used as an intermediate or relay node by neighboring nodes is likely to become a performance bottleneck. A multi-hop communication scenario in WSN communication is dipicted in Figure 1.6, where due to obstacles (e.g., building etc.), direct communicatio is not possible.

Fading

In different real-time environments, various objects (e.g., building, trees, mountains) causes diffraction, reflection, scattering, doppler fading of transmitted signals. Therefore, the received signal becomes a superposition of multiple delayed copies of the original signal (i.e., main signal path and its reflections). Depending on the phase shift of the signal delayed components, constructive or

destructive interference can occur at the receiver. This phenomenon is called multipath effect. If either the receiver or transmitting node is mobile, then resulting propagation phenomena shall be time-varying, and causes fading. Most of the research on wireless ad-hoc and sensor networks assumes ideal radio propagation model without inspecting the impact of fading and shadowing effects.

Collisions and Interferences

WSN consist of large number of nodes. These nodes share the same wireless medium, which generates high possibilities of simultaneous transmissions which results in interference and collisions. Collision avoidance is the basic target of MAC protocols because there is a high possibility of losing all colliding packets. This requires re-transmission and causes significant power loss. As a result, interference and collisions between transmitted packets can be a considerable barrier in achieving the required throughput. As the data load increases or in dense networks, severe performance degradation of the network can be witnessed due to congestion around sinks and burst-traffic.

In the licensed frequency bands, the traditional way to solve this problem is to assign orthogonal channels, so the users are the only ones allowed to transmit at that time using that frequency. This method has been used in AM/FM radio, over-the-air TV broadcasts and even in cellular communications. The frequency bands are auctioned to wireless telephone carriers. This approach has removed interference problem; meanwhile, it results in low utilization when the primary owner does not use the allocated frequency spectrum frequently. This disadvantage of static channel allocation has led to the use of shared or unlicensed frequency bands that can be used by multiple networks at the same time. The 2.4 GHz band is an example of this paradigm, used by 802.11 (WiFi), 802.15.1 (Bluetooth) and 802.15.4 (Zigbee) networks and even cordless telephones.

As WSN deployments increase in scale, their performance becomes more and more vulnerable to interference from internal sources because of poor coordination in the radio transmissions, and external sources. Hence, the new challenges are emerging day by day. If interference is not tackled properly, then it will become a big challenge in the coming decades for new deployments of WSN and IoT. In this thesis, we have worked on interference modeling for 802.11, 802.15.4 and LoRa systems for a simulator. But before going into the further details on these two standards and interference modeling for them, we will give a brief layout of WSN stack in next section.

3 WSN communication stack

Most common communication stack followed wireless systems is shown in Figure 1.7. Each layer in this model is responsible of some specific tasks, and each layer has some adjustable parameters, that can provide feedback to other layers. Sensor nodes behavior is adjustable on each layer to reduce the interferences.

3.1 Cross Layers

The three cross layers power management, mobility management and task management are used for network management. These layers make nodes work together in the network to increase its overall efficiency [3]. Mobility management detects and manages sensor nodes movements. Task

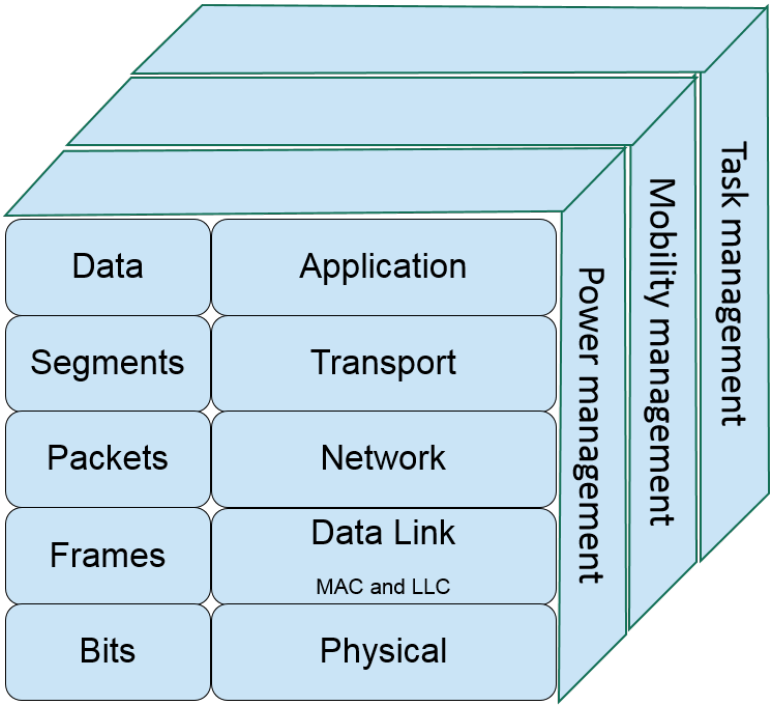


Figure 1.7: WSN communication stack.

management schedules sensing tasks in a given area. It also schedules on and off timings of the node.

3.2 MAC Layer and Upper Layers

Medium access control (MAC) layer is responsible for transmission medium control between neighboring nodes, in the network. This is one of two sub-layers of data link layer (DLL) in the OSI communication model.It is responsible for establishing the communication link between nodes and defines rules for medium sharing. MAC layer is also responsible for radio duty cycle (RDC) control. It controls the transceiver's wake up cycles.

The other sub-layer of DLC is logic link layer (LLC), which is concerned with the multiplexing of data-streams, traffic flow, error control and provides an interface to upper layers.

Examples of adjustable parameters in the DLL are:

- number of re-transmissions,
- determine whether the medium is idle or not based on energy detection threshold, and
- radio duty cycle parameters (active/sleep duration).

Some examples of feedback values provided by DLL are:

- number of transmissions before success/failure,

- connection success/failure,
- negative acknowledgments (NACK message) [16],
- DLL can be fully or partially functional through RF module or MCU.

The layer above DLL is network layer, which is in charge of transferring network packets from source to destination into or through other networks. It packages data into variable length packets and logical routing paths. Routing and delivery error reporting, data forwarding and logical connection setup are the primary responsibilities of network layer.

Some parameters that are available at the network layer are:

- maximum number of authorized hops,
- time period of sending for control messages.

Some of the feedback values available on the network layer are:

- number of hops required to reach the gateway,
- number of active neighbors or neighboring nodes that could act as a relay.

In single-hop LPWAN networks, since all nodes are supposed to communicate with the gateway directly, routing functionalities are not used. This can simplify the development of node's firmware and can save energy because radio duty cycling management becomes easier. At the same time, disabling multi-hop capabilities in a network can negatively affect the robustness of the system. For example, if for some reasons (e.g., an obstacle is placed nearby the sensor node), a node cannot reach a long distance gateway, then that node will become useless for the network. In multi-hop networks, network layer protocols can also help to reduce the interference by carefully choosing hopping nodes. But one of the main tasks in multi-hop networks is congestion control. In some networks, packet loss due to the overflow of re-transmitted packets. This also increases wireless medium usage and cause interference probability.

The layer above the network layer is the transport layer. Its main functionalities are reliability in the arrival of packets across the network, error checking mechanism, and congestion control. Unlike in computer networks, the sensor nodes in WSN have constrained in resources and energy consumption. Therefore, the transport layer functionalities are often transferred to lower layers.

The application layer in WSN communication stack provides software for application functionalities of the WSN. Data on this layer is to be sent to gateway nodes, through the network.

The adjustable parameters on the application layer are application specific and include:

- data sampling rate,
- data generation period,
- data compression strategy, and
- data resolution for each sample.

The feedback parameters on the application layer are to ensure the quality of service like data losses and delay.

3.3 PHY Layer

In the communication stack of WSN, physical layer is the last one. Physical layer refers to how the data or the bits get mapped onto the physical medium. In the wireless systems, that mapping maps bits onto the patterns of energy in a RF medium.

Some of the adjustable parameters are:

- central frequency,
- modulation,
- effective bitrate, and
- transmission power.

One of the main components of the PHY layer is the modulation. The modulation is the algorithm that defines how the digital values or the bits are mapped onto the RF energy. There are few parameters that can be tweak to do that, they are:

- frequency,
- phase and
- amplitude.

Some of the feedback parameters on physical layer are:

- RSSI,
- link quality indicator [16],
- result of the CRC check.

The PHY layer activities are performed by the radio chip, Thus the availability of adjustable and feedback parameters also depends on the hardware.

All the above mention adjustable parameters of all layers can be controlled in a coordinated way. Parameter controlling algorithms use the feedback values to change the adjustable parameters [17]. This thesis is mostly focused on physical layer and its implementation in the wireless sensor network simulator.

4 Characteristics of Noise in Wireless Channel

Some aspects of wireless channel make wireless communication more challenging as compare to wired communication like probabilistic wireless channel behavior, jamming and interception, limited radio range, interference from other radio devices and much more [5]. These aspects change the characteristics of the transmitted signal as it travels through the wireless channel from transmitter to receiver. Received signal characteristics depend on the distance from transmitter to receiver, multiple paths that the transmitted signal takes and its surroundings (buildings and other objects or obstacles). Besides, due to limited radio spectrum, it is not reasonable for large wireless networks to try to communicate without interference. Probably other neighboring radio devices from inside or outside of the network will transmit data using the same or nearby radio frequency band

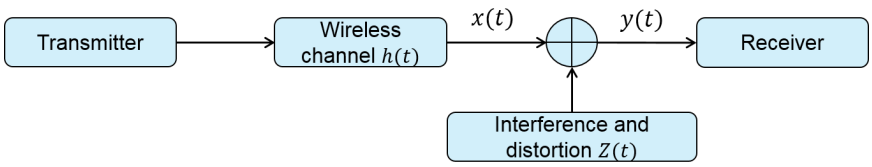


Figure 1.8: Wireless channel model.

at the same time. Consequently, at the receiver, many undesired signals from interfering nodes will add up into the desired signal. This phenomenon is called interference, and it causes performance degradation of communication networks [5]. Many WSNs use unlicensed Industrial, Scientific and Medical (ISM) frequency band that makes interference probable phenomenon on a given channel. System throughput gets influenced by the interference as it can congest wireless medium, cause packet drops, re-transmissions, link instability and inconsistent protocol behavior [18] [19] [20].

Reconstruction of the input signal at the receiver is possible by an appropriate model of the medium or channel around transmitter and receiver. This model is called channel model and it should be accurate enough to represent the characteristics of the wireless channel. To mitigate the interference effects from the received signal, the channel model plays a key role. A transmitted signal $x(t)$, after passing through the wireless channel $h(t)$, becomes $y(t)$, as:

$$y(t) = h(t) * x(t) + Z(t) \tag{1.1}$$

where $*$ is the convolution operator and $Z(t)$ is the noise, as shown in Figure 1.8.

Noise always affects the transmitted signal when it passes through the wireless channel. There are many types of noises that could add to the transmitted signal and cause the distortion in the received signal. It is necessary to categorize these noises for a better understanding of the wireless channel. Figure 1.9 shows the different type of noises that are present in the wireless environment.

Channel noise modeling or interference modeling has very long history. At first, it was done using Gaussian random process which has finite variance and very light tail [20] but what if there are much more variation in noise samples that couldn't be captured by Gaussian distribution? In a wireless network, neighboring radio devices from inside or outside of the network can transmit data using the same or nearby radio frequency band at the same time which will cause interference in the network [21] [22].

5 Interference Modeling

5.1 Gaussian Distribution

There are many ways proposed in the literature to model interference [23] [24]. The most common approach is by using Gaussian random process. This is appropriate when the interference is an accumulation of a large number of independent signals and no term dominates the sum thus the Central Limit Theorem (CLT) can be applied. In (1.1), noise $Z(t)$ can be model with Gaussian distribution, Additive White Gaussian Noise (AWGN), with zero mean and a variance N_0 : Z $N(0, N_0)$, only if

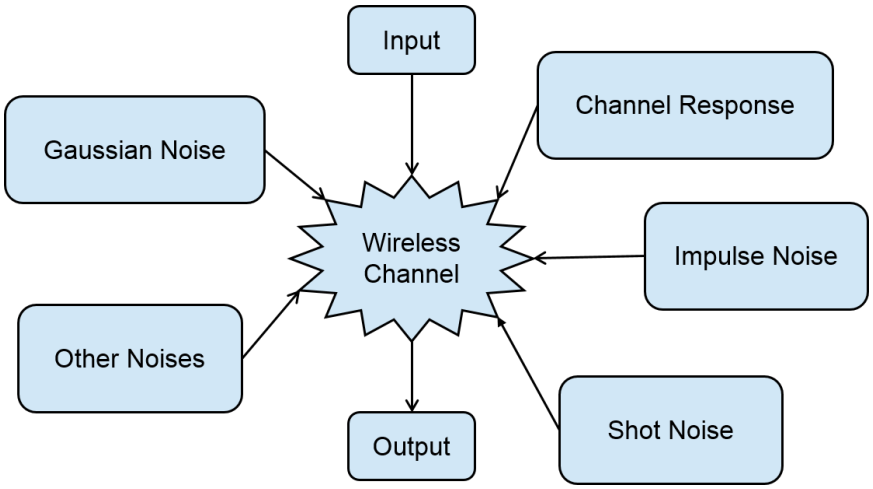


Figure 1.9: Characterization of noise in wireless channel.

the noise samples are considered independent and identically distributed (iid) [23].

Let us understand the Gaussian distribution in the context of a wireless communication channel. Consider a sensor network with n transmitting nodes and a receiver. If the node with transmitter T_0 wants to send data to the node with receiver R_x, the received signal will be:

$$ReceivedSignal \quad R_x = h_0 X_0 + \sum_{i=1}^{n} h_i X_i \tag{1.2}$$

where X_0 is desired transmitted signal multiplied with channel response h_0. X_i is an interfering signal comming from the user i through channel h_i. $\sum_{i=1}^{n} h_i X_i$ is the accumulated interfering signals coming from n wireless devices. If we assume that these received samples $(\sum_{i=1}^{n} h_i X_i)$ follow a Gaussian distribution then, they can be mathematically expressed by the following probability density function (PDF):

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1.3}$$

where μ is the mean and σ is the noise variance. For zero mean noise $(\mu = 0)$, (1.3) will become:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \tag{1.4}$$

Gaussian distribution has many well suited properties in the modeling of the wireless channel, but there are several scenarios where CLT is not applicable. The Gaussian model has shown to be useful since it results in best and simplified performance for communication systems when signals and noise fit this distribution. However, real-time systems may exhibit performance degradation when the signal and noise distributions are non-Gaussian. In many cases, the perceived noise distribution differs from the Gaussian assumption and it exhibits impulsive behavior. It is when [19]:

- the number of interferers are not large in number, and/or
- there are some dominant interferers present in the environment and makes noise non-Gaussian or impulsive in nature.

5.2 Alpha-Stable Distribution Model

The α-stable distribution was first put forward by Paul Levy in 1925, but it was not truly well known for modeling until the last decades of the 20th century. It was mainly because, α-stable distributions do not have closed-form expressions for their probability density functions (PDFs). However, they do have closed-form characteristic functions. Due to the unavailability of PDF function, computationally intensive methods are required to generate α-stable random numbers from the characteristic function. Nowadays, with the significant computing power available in personal computers (PCs) and generic digital signal processor (DSP) devices, computationally intensive methods to handle α-stable distributions are a no more insurmountable obstacle.

The CLT states that the normalized sum of a large number of i.i.d. random variables of finite variance converges in distribution to the Gaussian distribution. When the constraint of finite variance is excluded from the CLT (i.i.d. random variables may have either finite or infinite variance), It becomes the generalized CTL and the sum converges in α-stable distribution. Therefore, the Gaussian distribution is a special case of α-stable distributions. Impulsive noise is a noise which often occurs in wireless and wireline communications in the indoor and outdoor environments. Impulsive noise or non-Gaussian noise can be a major source of error in data transmission and also can be a contributor to the increase of total error rate of the system. In more real wireless environmental circumstances, some noises are impulsive in nature; e.g. atmospheric noise which is caused by lighting, radar noise and so on [25]. In particular, these rare events with large amplitude have a high impact on communication links but cannot be modeled with the Gaussian process. In this case, another mathematical model is required.

α-Stable distribution that satisfies Generalized Central Limit Theorem and presents heavy tails [26] [27], allows to take these rare events into account in a more accurate way. Except the Gaussian distribution, stable distributions have huge amount of scatter and infinite variance.

α-stable distribution has infinite variance and heavy tail expect the case when $\alpha = 2$, it becomes Gaussian distribution [28]. α-stable distribution is a heavy tail distribution. There is no closed form expression or probability density function of the α-stable distribution, but it can be described by its characteristic function [28], which is:

$$\phi(\theta) = \begin{cases} exp\{-\sigma^{\alpha}|\theta|^{\alpha}\big(1 - i\beta sign(\theta)tan(\frac{\pi\alpha}{2})\big) + i\delta\theta\} & \alpha \neq 1 \\ exp\{-\sigma|\theta|\big(1 + i\beta\frac{2}{\pi}sign(\theta)ln|\theta|\big) + i\delta\theta\} & \alpha = 1 \end{cases} \tag{1.5}$$

where

$$sign(\theta) = \begin{cases} 1 & \text{if} \quad \theta > 1 \\ 0 & \text{if} \quad \theta = 1 \\ -1 & \text{if} \quad \theta < 1 \end{cases}$$

and α, β, σ and δ are four parameters of α-stable distribution and can be explained as [28]:

- $\alpha \in (0, 2]$ index of stability: sets the degree of the impulsiveness of the distribution. At $\alpha = 2$ distribution becomes Gaussian distribution.
- $\beta \in [-1, 1]$ skewness parameter: specifies distribution curve is skewed towards right or left.
- $\sigma \in [0, \infty)$ scale/dispersion parameter: measures the spread of the noise samples around the mean.

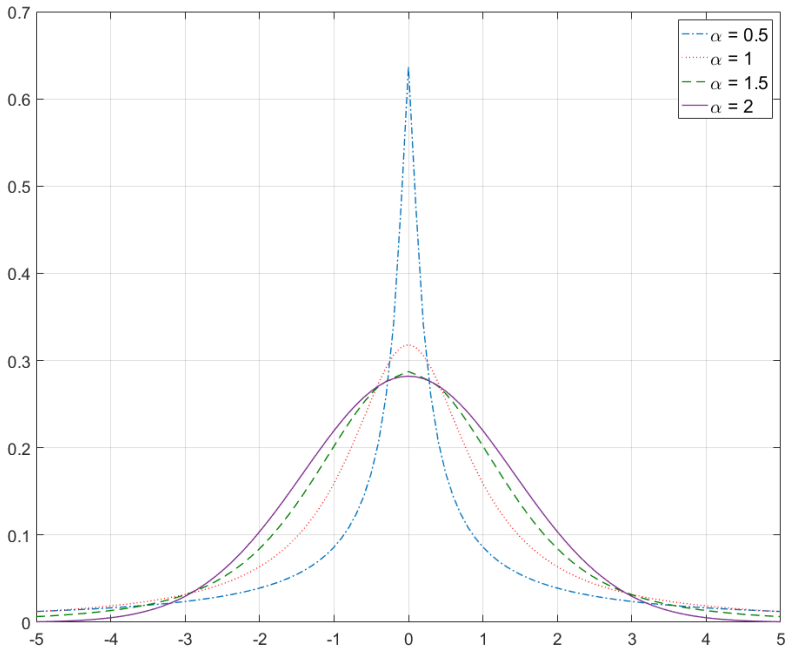


Figure 1.10: Examples of stable PDFs for $\beta = 0,\ \sigma = 1,\ \delta = 0$, and different values of α.

- $\delta \in \mathbb{R}$ location parameter: corresponds to the median for $0 < \alpha < 1$, and to the mean for $1 < \alpha \leq 2$.

Figure 1.10, shows the comparison of the probability density functions of α-stable distribution with four different values of α. One case corresponds to $\alpha = 2$ and the other three correspond to lower values for α. It can be observed that the lower the value of α, the heavier the tail of the PDF. Figure 1.11 shows the tail decay of these cases. Skewness parameter $\beta = 0$ for a symmetrical distribution. Dispersion parameter σ is used to scale the random variable to different magnitudes whereas location parameter δ is used to control the location of the variable. It is worth mentioning that for an α-stable distribution with $\alpha > 1$, δ is equal to its mean value. A relevant member of the family of α-stable distributions is when $\alpha = 2$, which leads to the Gaussian distribution. In this case, σ^2 is not equivalent to the variance. Nevertheless, they are related since $\sigma^2 = VAR(.)/2$. Figure 1.12 shows the distribution of the noise samples with different values of α. It can be observed that higher the value of α, higher the impulsive noise.

6 System Model

We model the spatial distribution of the nodes according to a homogeneous Poisson point process in the two dimensional infinite plane. The probability of n nodes being inside a region $\mathbb{R}$ depends

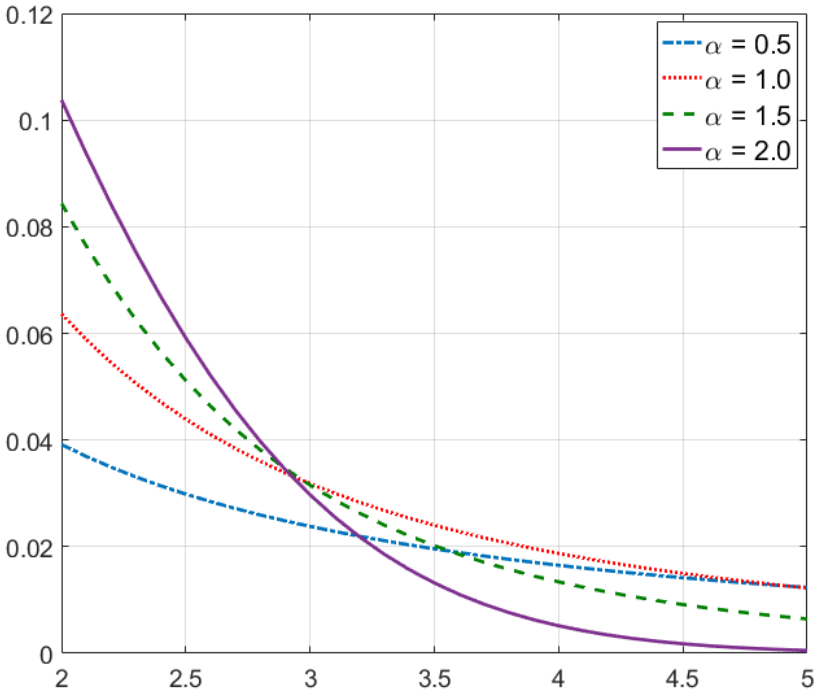


Figure 1.11: Tail decay for the same cases. Note the heavy tail in the cases where $\alpha < 2$.

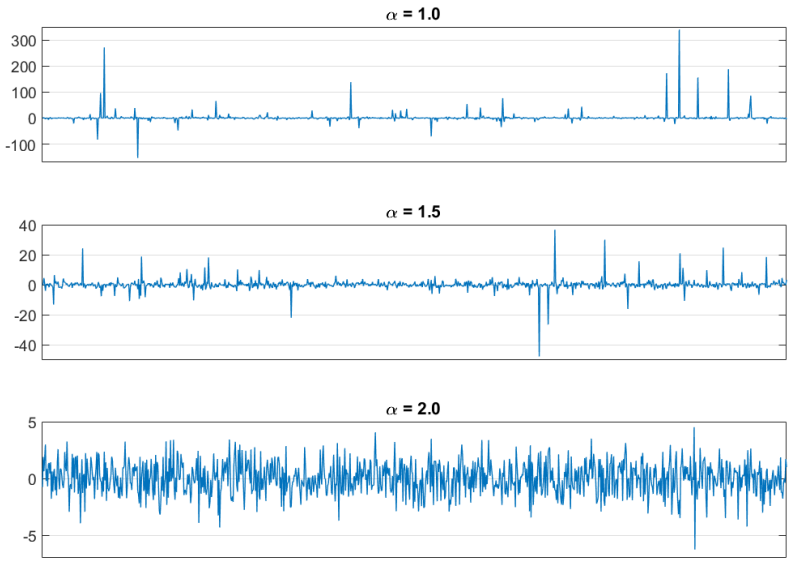


Figure 1.12: Distribution of noise impulses using different values of α.

on the total area $A_{\mathbb{R}}$ of the region and is given by:

$$\mathbb{P}\{n\ in\ \mathbb{R}\} = \frac{(\lambda A_{\mathbb{R}})^n}{n!} e^{-\lambda A_{\mathbb{R}}} \tag{1.6}$$

where λ is the spatial density of the nodes, in nodes per unit area. We define the interfering nodes to be the terminals that are transmitting during the time interval of interest, within the frequency band of interest,and hence are effectively contributing to the total interference. Then, irrespective of the network topology (e.g., pointto-point or broadcast) or multiple-access technique (e.g.,time or frequency hopping), the above model depends only on the density of interfering nodes.

Using the general expression from (1.2), we now characterize the interference coming from the interferer nodes. Let a set of distances $\{R_i\}_{i=1}^n$ between the receiver node and several interferer nodes within the specified area. An arbitrary random quantity $Q_i = [Q_1, Q_2, ..., Q_n]$ is associated with each interferer i. Q_i can be defined as $Q_i = k_i \times x_i$, k_i denotes the small scale fading and shadowing, and $h_i = \frac{k_i}{R_i^b}$. Let Y denotes the accumulated interference at the receiver node generated by interferer nodes scattered in a finite plane, hence

$$Y = \sum_{i=1}^{n} \frac{Q_i}{R_i^b} \tag{1.7}$$

where b is channel amplitude attenuation coefficient, while corresponding power loss exponent is γ. The characteristic function is the characteristic function of the α-stable distribution [5]:

$$\phi_Y(w) = exp(-\sigma|w|^\alpha) \tag{1.8}$$

where

$$\alpha = 2/b \tag{1.9}$$

and

$$\sigma = \lambda\pi C_{2/b}^{-1} E\{|Q_{i,n}|^{2/b}\} \tag{1.10}$$

α is index of stability and it sets the degree of the impulsiveness of α-stable distribution [5] [28]. σ is scale/dispersion parameter and it measures spread of noise samples following α-stable distribution [28]. C_α is some constant defined as:

$$C_\alpha \triangleq \begin{cases} \frac{1-\alpha}{\Gamma(2-\alpha)\cos\frac{\pi\alpha}{2}}, & \alpha \neq 1 \\ \frac{2}{\pi}, & \alpha = 1 \end{cases} \tag{1.11}$$

$\Gamma(\cdot)$ denotes gamma function. If any positive valued, independent and identical variable a is Rayleigh distributed then its probability distribution function will be:

$$f(a) = \frac{a}{\varepsilon^2} e^{-a^2/2\varepsilon^2} \quad \text{for} \quad a \geq 0 \tag{1.12}$$

where ε is scale parameter of Rayleigh distribution. Its value determines statistical dispersion of Rayleigh distribution.

For $|Q_{i,n}|^{2/b}$ where let $|Q_{i,n}|$ is Raleigh distributed. Then expected value $E\{|Q_{i,n}|\}^{2/b}$ can be obtained as:

$E[|Q_{i,n}|^p] = \int_0^\infty a^p \; f(a) \; da \qquad \because p = \frac{2}{b} \;\text{ and }\; a = |Q_{i,n}|$

$E[|Q_{i,n}|^p] = \int_0^\infty \frac{a^{p+1}}{\varepsilon^2} \; e^{-a^2/2\varepsilon^2} \; da$

We will get:

$E[|Q_{i,n}|^p] = 2^{p/2} \; \varepsilon^p \; \Gamma(p/2+1)$ and $p = 2/b$, so

$$E\big[|Q_{i,n}|^{2/b}\big] = 2^{1/b} \; \varepsilon^{2/b} \; \Gamma\Big(\frac{1}{b}+1\Big) \tag{1.13}$$

Channel power loss exponent γ is twice the channel amplitude attenuation coefficient b, we get

$$E\big[|Q_{i,n}|^{4/\gamma}\big] = 2^{2/\gamma} \; \varepsilon^{4/\gamma} \; \Gamma\Big(\frac{2}{\gamma}+1\Big) \tag{1.14}$$

(1.8), (1.9) and (1.10) can be expressed as

$Y \sim S\Big(\alpha_Y = 2/b, \beta_Y = 0, \delta_Y = 0 \text{ and } \sigma_Y = \lambda\pi C_{2/b}^{-1} E\{|Q_{i,n}|^{2/b}\}\Big)$
or
$Y \sim S\Big(\alpha_Y = 2/b, \beta_Y = 0, \delta_Y = 0 \text{ and } \sigma_Y = \lambda\pi C_{2/b}^{-1} 2^{1/b} \; \varepsilon^{2/b} \; \Gamma\big(\frac{1}{b}+1\big)\Big)$

7 WSN Simulator

As mentioned before, deployment of sensor networks in an efficient and optimal way has become an important topic these days. Interference from the neighboring nodes makes it difficult for a

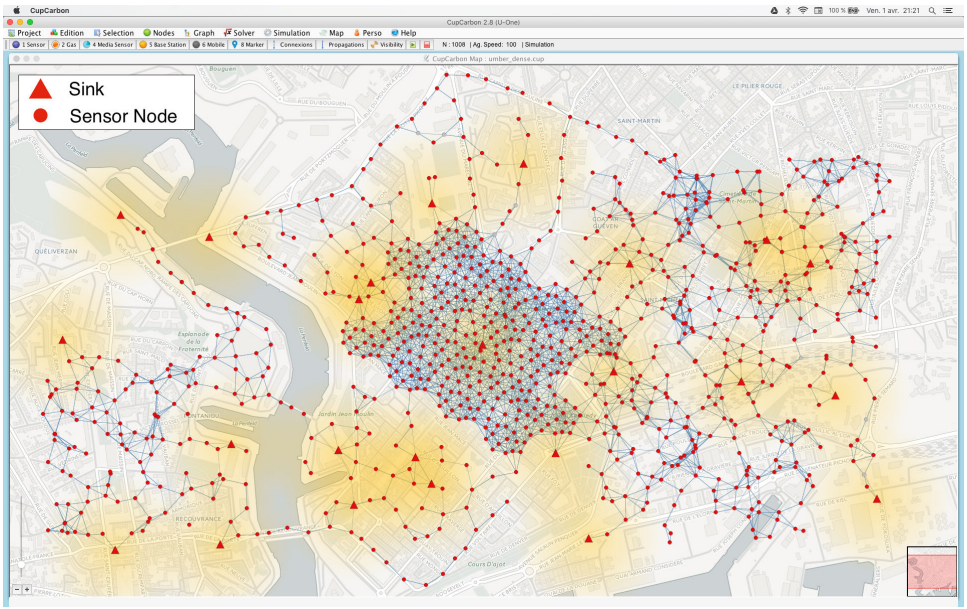


Figure 1.13: GUI of CupCarbon: a WSN Simulator.

communication system to achieve the required throughput and also makes it difficult to predict the network behavior. Finding a suitable simulation environment for WSN is a difficult task. Simulation of the wireless sensor network in real time environment is an essential part in deployment of a new network because running the real experiments on WSN testbeds is not cost effective.

7.1 Cupcarbon

CupCarbon is an open source WSN simulator which is recently developed for the better analysis and understanding of WSN protocols. This simulator runs under the Java environment and can be downloaded from the Internet (http://www.cupcarbon.com). Simulations in CupCarbon are based on physical layer of Open System Interconnection (OSI) model. The main objective of this simulator is to design and simulate WSN dedicated to smart cities and IoT applications. It is the only simulator (in our knowledge) that supports wireless communication interference models and signal propagation models like Gaussian and α-stable models. It allows to validate distributed algorithms in both 2D and 3D environments. CupCarbon also allows sensor nodes to set their radio visibility according to the urban environment. It provides graphical user interface (GUI) to facilitate the comprehension and visualization of WSN as shown in Figure 1.13. We have randomly placed nodes (sink and sensor nodes) on a city map.

The already existing simulators, e.g. NS-2 [29], WSNet [30], Castalia [31], TOSSIM [32], etc., mainly focused on routing protocols. Howeveer when it comes to radio channel modeling for smart cities or IoT applications, there interference models are simple and unrealistic. The CupCarbon platform aims to contribute to the following:

- Give detailed study on WSN deployment with consideration of mobility and the availability of radio spectrum,
- Simulate performances and services of WSN in $2D$ as well as in $3D$ for more realistic environment,
- Give detailed analysis of the feasibility of communication under given environment, reliability of the network and network cost,
- Detects potential interference zones, to enhance the communication quality,
- Gives simulation results of the radio propagation in a real urban environment, more fast and accurate and quickly,
- Provides the visualization of simulation results, to debug and validate any developed algorithm.

Open Source Code

The opening of the source code is not only necessary for embedding or modifying the simulation engine, but also gives significant help in debugging simulation models. Commercial and some non-commercial software do not give the source code of the simulation kernel, CupCarbon is fully open-sourced (OMNET++ and NS2). CupCarbon is not paid for the license, and users can learn its logical architecture.

8 Thesis Contribution

Most of WSN use unlicensed Industrial, Scientific and Medical (ISM) frequency band that makes interference probable phenomenon on a given channel. System throughput gets influenced by the interference as it can congest wireless medium, cause packet drops, re-transmissions, link instability, and inconsistent protocol behavior. Most of the available literature on WSN have not considered interference effect from inside or outside of the network unless nodes are in radio range of each other and system have unlimited radio spectrum. In contrast, real-time WSN are band limited and nodes can interfere with other nodes even if they are beyond their communication range.

Moreover, The rapid growth in the field of WSNs entails the need of creating new simulators that have more specific capabilities to tackle interference and multipath propagation effects that are present in the wireless environment. Finding a suitable simulation environment that allows researchers to verify new ideas and compare proposed future solutions in a virtual environment is a difficult task.

Keeping the above arguments in mind, we have contributed to the following:

- Brief description about technical features of IEEE 802.11 and IEEE 802.15.4/ZigBee based physical layer structures.
- As LoRa PHY is not openly available, we also provide a comprehensive analysis of technical features of the PHY layer associated with LoRa.
- Study of system behavior with Gaussian noise and wireless channel impulsive interference modeling with α-stable interference

  - What parameters of α-stable distribution can be estimated for more accurate representation of wireless channel?

- Include the Gaussian and α-stable interference model in CupCarbon simulator for evaluation of wireless channel behavior.
- Study the performance of above mentioned PHY systems under both kinds of background noise (i.e., Gaussian and α-stable). From these simulation results, we conclude that if an impulsive background is mistakenly assumed as Gaussian in the evaluation, the system performance is largely overestimated.
- Capacity limitations of CSS based PHY layer are explored by calculating the packet reception rate (PRR) with varied network sizes.
- Capture effect and successive interference cancellation are incorporated in the LoRa based system and simulation results show significant improvement in the probability of successful reception rate.
- Find an optimal way to integrate these methods in the WSN simulation platform CupCarbon.

9 Thesis Structure

Rest of the thesis is organized as follow:

- Chapter 2 describes:
  - Compares most of the available simulators for the WSN.
  - Explains CupCarbon plate form and architecture.
  - Explains where this thesis work is used in the CupCarbon.
- Chapter 3 defines:
  - Some technical features associated with IEEE standard 802.11 and 802.15.4,
  - Considered system model and working of IEEE standard 802.11 and 802.15.4 based transceivers,
  - System behavior with Gaussian noise and α-stable interference.
  - Results with the ray tracing model and statistical model.
  - How above mentioned systems can be added in Cupcarbon.
- Chapter 4 focuses on:
  - Some key features of LoRaWAN and LoRa technology,
  - Capture phenomenon along with its different scenarios and its implementation in LoRa,
  - Successive interference cancellation and its impact on system performance when combined with LoRa.
  - How above mentioned systems can be added in Cupcarbon.
- Finally, Chapter 5 gives conclusive remarks about whole work in this thesis and shares some perspectives for the future work.

2

Wireless Sensor Network Simulator

1 Introduction

The rapid growth in the field of WSNs entails the need of creating new simulators that have more specific capabilities to tackle interference and multipath propagation effects that are present in the wireless environment. Finding a suitable simulation environment that allows researchers to verify new ideas and compare proposed solutions in a virtual environment is a difficult task. CupCarbon is an open source WSN simulator. It provides better analysis and understanding of WSN protocols. It allows users to verify new ideas and evaluate proposed solutions in a more realistic virtual environment for wireless sensor network applications. This simulator runs under the Java environment and can be downloaded from the Internet (http://www.cupcarbon.com). It supports wireless communication interference models like Gaussian and α-stable models [33] [34] [35] and include a $3D$ ray-tracing channel model.

2 Importants of simulators for WSN

WSN consists of hundreds or thousands of sensor nodes so, performing real-world experiments can be very expensive [36]. However, it is important to be able to test the envisioned network configrations before its deployment in order to avoid the long and expensive task of correcting and dysfunction. WSN is a unique wireless network in many aspects, e.g., limited resources (power, memory, and processing ability), a significant quantity of nodes, multitasking, decentralized collaboration, heterogeneity and so on. These constraints make the development of any simulation tools more challenging. By taking into account these unique characteristics of WSN and the demands in different WSN design fields (e.g., application design, node system design and communication protocol design), we can summarize the key elements that are critical to any WSN simulation framework, as follows:

- **Fidelity:** A simulator should be able to predict and model the real-world environment. For WSN, it requires precise radio channel models and node system models. Fallacious models may lead to erroneous conclusions.
- **Scalability:** Scalability and performance are the primary concerns for any WSN simulator. The high number of concurrent transmissions would require to simulate many simulataneous active links. Even, a single radio channel is so complex to simulate, when you have hundred transmitters and receivers simultaneously active, it results in thousand (100×100) active links.

- **Energy-aware:** Power consumption is particularly critical for WSN operating on limited power stocks, such as batteries or solar cells. WSN designers must obtain fair power consumption and timing figures to harmonize their applications before the deployment in real-time environments. Hence, a simulator shall be able to precisely capture the power consumption and timing information of the embedded software and radio communication at the network level.

- **Heterogeneity Support:** Many deployed WSN systems are heterogeneous systems, which incorporate a blend of elements with widely diverging capabilities [14]. So, modeling different sort of nodes and managing the interconnections among those nodes are essential in WSN simulations.

- **Graphical user interface (GUI):** A good GUI can facilitate and speed up the process of the establishment of the WSN topology and the composition of primary modules. It also allows the immediate visualization of the simulation results. Moreover, it assists to track and debug the simulation in real time. It provides easier control of the simulation environment.

Currently, there are tens of WSN simulators available. Selection of suitable simulators depends on the given application. In this chapter, we survey several well-known simulators.

3 Design of WSN Simulator

The design, development, and deployment of sensor networks are application specific, mainly because of the distinctiveness of the considered deployment environment. Reliable generic predictive models for radio propagation or data correlation are seldom available. An absolute preliminary test phase is therefore essential, either utilizing particularly crafted testbeds or via sound simulations. Different WSN applications must be examined on a large scale, and under varying complex conditions to apprehend an adequately wide range of communications, both among sensor nodes, and with the environment. WSN simulator consists of multiple modules namely events, medium, node, environment, protocols, transceiver, and applications [37]. Every category is described by an interface that defines its methods and events consumed and generated.

- **Event:** is an abstract base class which provides basic functionality for all other events. It includes the execution time of an event, and implements methods to: compare and distinguish events based on their execution times, print themselves to a string, determine whether events are equal, and provide an abstract method to the event.

- **Medium:** models the wireless medium. It allows nodes to send and receive data and information and is also capable for informing nodes of signals that affect it.

- **Environment:** is similar to the medium module but differs in the implementation. The implementation of the environment module has properties that are more related to the physical phenomenon modeled. Physical aspects of concern in WSNs include humidity, temperature, light, magnetic field, sound, chemical process, optical, etc.

- **Node:** which is the single sensor node of the network. It contains both hardware and software components. Essential hardware components are sensors/actuators, processor, transceiver, an energy source (battery). The software components could be network protocols, GPS, and applications.

- **Transceiver:** models the hardware transceiver for the sensor node. It models different transceiver states (i.e., awake, sleep, transmit, receive or standby), and their associated behavior and power consumption.
- **Physical Protocol:** represents the lowest layer protocols in the network stack. It is usually implemented in the transceiver hardware. The main purpose of the physical layer is to send and receive sensed data. Besides that it provides services for carrier sensing, changing the state of the transceiver, received energy detection on received packets, etc.
- **MAC Protocol:** is represented by the second layer of the network stack. It is often implemented on the node's processor. This layer provides services like setting and getting protocol parameters, state changing of the MAC layer (i.e., low power mode), etc. A good WSN simulator usually implements several MAC protocols.
- **Routing Protocol:** resides above the MAC protocol. It provides packet routing services for multi-hop communications between nodes.
- **Application Layer:** resides at the top of the network stack. It provides the interface between the lower layers and between the sensors and actuators as well.

WSN simulators should include all the above mentioned points, but usually, they include the small parts of them. Besides, a WSN simulator should also include the following capabilities:

(i) Reusability and availability: Simulations are used to test novel methods and techniques as realistic as possible and controlled scenarios [38].

(ii) Scalability is a primary concern when designing a WSN simulator. In the design of a simulator, performance is usually bounded the effectiveness of the simulator and the capability of the processor define the scalability of a simulator, limitted by teh logs storage size requirements, and memory [39].

(iii) Aid for rich-semantics scripting languages to interpret experiments and processing results is also required. A large number of variables involved in a WSN experiment involves the use of explicit input scripting languages. Moreover, a large number of output data is also produced through experiments. Hence, a proper output scripting language, which assists in obtaining the results from the experiments precisely and quickly, is also desired [40].

(iv) Graphical support for simulations is interesting in three aspects:

  (a) As debugging support: The more practical way to immediately detect a bad behavior during simulation is to "watch" and follow the execution. A graphical interface should support capability of inspection of modules, together with "step-by-step," variables and event queues in real time, and "run-until" execution possibilities. These characteristics make graphical interfaces a powerful debugging tool.

  (b) Visual modeling and composition tool: These usually facilitate and speed up the design of short experiment schedules. However, for large-scale simulations, they are not very helpful.

  (c) Finally, it should also allow quick visualization of results without the use to develope post-processing application [41].

All these points underline the complexity of WSN simulations and related challenges. Clearly, it is highly complex to gather all features impacting the performanceof WSN, in a realistic real time and large scale simulations. Some simulators lack in the availability if protocol models, which increases by developing time and some simulators have limited scalability, etc. Moreover, modeling problems arise when a new environment and the energy components are considered. They also compromise on scalability and accuracy [42]. So, an in-depth detailed study of these issues is mandatory for the better understanding of sensor networks and their simulators.

4 WSN simulators

There are many different possible platforms for simulations and testing of routing protocols for WSN. In the following we present some of the most popular simulators for WSN. Various aspects like energy efficiency, resources, decentralized collaboration, fault tolerance, simulation scenarios, global behavior, etc. have been compared.

4.1 NS-2

NS-2 is one of the most known discrete event network simulator. In 1989, NS-2 was first developed as the variant of the REAL network simulator and had evolved over the past few years. It has a modular approach and is effectively extensible. It provides support for the simulation of routing, TCP, and multicast protocols over both wired and wireless (local and satellite) networks. Its support for the wireless networks was added later for wireless LAN protocols, mobile ad-hoc networks, and WSNs [29].

The protocols behavior and simulator itself are C++ based, while interpreted scripts performing the simulation itself are to be OTcl based. It supports an object-oriented design which provides straightforward creation and adoption of new protocols. The key features for WSN simulation include wireless channel models, battery models, hybrid simulation support, lightweight protocol stacks, hybrid simulation support, and scenario generation tools. It gives a visualization tool NAM (Network AniMator). All the simulations run at the packet level that results in a more detailed form. In spite the fact that NS-2 is considered to be too general and inappropriate for WSN simulations by some researchers, an example of NS-2 usage is a simulation of Low Energy Adaptive Clustering Hierarchy (LEACH) protocol used for routing and clustering of sensor nodes in a network. The NS-2 has a rich set of IP network focused protocols [43].

Because NS-2 has never been intended to design for WSN, that's why it does not scale very well and has some troubles if the number of nodes count in a network exceeds 100. It also cannot simulate problems of the bandwidth or the power consumption constraint in WSN. Sensing hardware models, packet formats, MAC protocols, and energy models all differ from those found in most sensors. It also lacks in the application model. However, in many network environments, this is not a problem, but sensor networks usually contain interactions between the application level and the network protocol level.

4.2 TOSSIM

TOSSIM (http://docs.tinyos.net/index.php/TOSSIM) is a discrete event simulator based on TinyOS operating system and was developed at UC Berkeley. It enables simulation of an application written in C language (network embedded system C) for real hardware. The simulation scenario can be

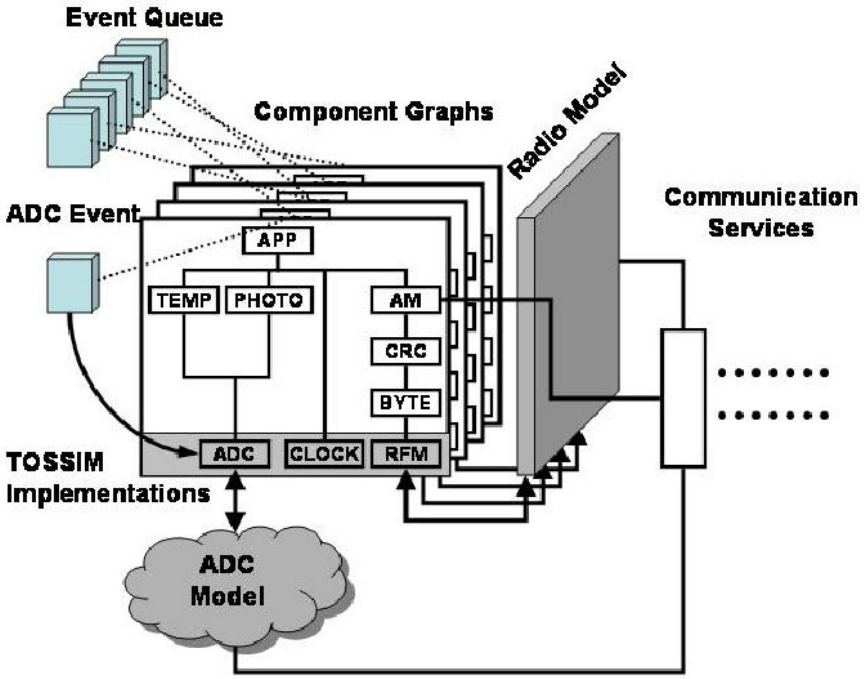


Figure 2.1: TOSSIM architecture: frames, events, models, components and services

written either in Python or C++ [32]. TOSSIM was designed mainly for the TinyOS applications to be run on the MICA Motes.

The TOSSIM architecture is comprised of following parts:

- TinyOS Component Graphs (Frames),
- Hardware abstraction components,
- Models (Radio and ADC Models),
- Execution Model (Events), and
- Communication Services .

Figure 2.1 shows five main components that make up TOSSIM. TOSSIM can simulate entire TinyOS applications. It can replace a low-level radio chip component for more precise simulation of the code execution, or replace a packet-level communication component for packet-level simulation. It facilitates TinyOS developers to choose the complexity and accuracy of the selected radio model. It simulates TinyOS network stack at the bit level, allowing experimentation with low-level protocols in addition to top-level application systems [44].

TOSSIM's "run-instantly" execution model does not capture CPU time. Also as interrupts are considered as discrete events, TOSSIM does not model the resulting possible TinyOS data races and preemption. Compilation steps lose the fine-grained timing and interrupt properties of the code. Which can be crucial when the application operates on the hardware and communicates with

other nodes. This simulator is not very flexible as it runs with the assumption that each node is programmed in the same way. It does not model energy consumption, but the PowerTOSSIMz can tackle this problem.

4.3 GLoMoSim

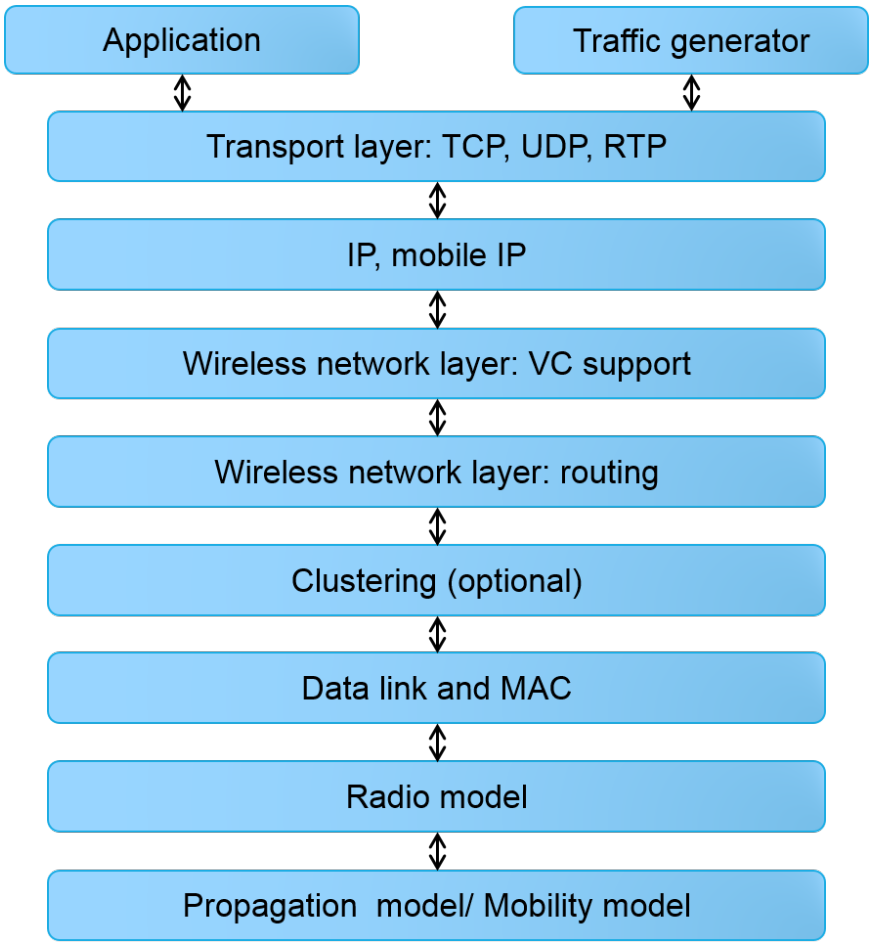


Figure 2.2: GloMoSim architecture

Global Mobile Information System Simulator (GloMoSim) is a C-based library for sequential and parallel simulations of wireless networks [45]. It is composed as a set of library modules, which simulates a particular wireless communication protocol, in the protocol stack. Its ability to offer a parallel environment, makes it distinguish from the other simulators. With all its protocols implemented as modules in the GloMoSim library, GloMoSim was also intended to be extensible like NS-2. Node aggregation technique offered by GloMoSim gives significant advantages in the simulation performance. GloMoSim offers various choices for CSMA MAC protocols (including 802.11), radio propagation, implementations of UDP and TCP, and mobile wireless routing protocols. GloMoSim can be executed using many synchronization protocols and was successfully implemented on

both distributed memory computers and shared memory [46].

GloMoSim has some limitations as well. GloMoSim currently supports protocols for a completely wireless network. In the future, the developers anticipate adding functionality to simulate a wired as well as a hybrid system with both wired and wireless capabilities. GloMoSim is effectively limited to IP networks because of low-level design assumptions. Therefore, it experiences the same problems as NS-2, the energy models; packet formats, and MAC protocols are not representative of those used in WSN.

4.4 Avrora

Avrora was a research project by the UCLA Compilers Group and is an open-source cycle-accurate Java-based simulator for embedded sensing programs. It can emulate two common platforms, Mica2 and MicaZ, and run AVR elf-binary or assembly codes for both platforms. Unlike traditional simulators, which only simulate specific platforms, Avrora is independent to the language and operating system. It gives a framework for program analysis, allows static checking of embedded software and also an infrastructure for future program analysis research. Avrora emulates a network of motes, runs the actual microcontroller programs (rather than models of the software), and runs accurate simulations of the devices and the radio communication [47].

Avrora is an accurate, reliable and scalable simulator and provides hardware platform to run sensor programs. It has an almost complete implementation of the mica2 hardware platform. It is also capable of running an entire sensor network simulation with full timing accuracy, allowing programs to communicate via the radio using the software stack provided in TinyOS. It also has an extension point that will enable users to create a new simulation type and choose the kind of simulation to perform, depending on the number and orientation of the nodes. Developers claim that Avrora scales to networks of up to $10,000$ nodes and delivers as much as 20 times faster than the many other simulators with equivalent accuracy [48].

One significant limitation of Avrora is that it does not model clock drift, a phenomenon where sensor nodes may run at slightly different clock frequencies over time due to either temperature, manufacturing tolerances, or battery performance. Recently, developers have modeled distance attenuation for multi-hop scenarios, but have not yet been able to model mobility.

4.5 SENS: A Sensor Environment and Network Simulator

SENS is a customizable sensor network simulator written in C++, for WSN applications [49]. There exists a thin compatibility layer to enable direct portability between SENS and real sensor nodes. Multiple component implementations in SENS allow varying degrees of realism. Users can assemble application-specific environments; such environments are modeled in SENS by their various signal propagation characteristics. The corresponding source code that is executed on simulated sensor nodes in SENS can also be deployed on actual sensor nodes. This enables application portability. SENS is a wireless sensor network simulator with modular, layered architecture with customizable components which models network communication and the physical environment. It allows realistic simulations, by using values from real sensors to represent the behavior of component implementations. Such action includes sound and radio signal strength characteristics and power usage [50].

The ability to develop portable applications is an essential feature of SENS, considering that WSN platforms continuously emerge as new sensor node implementations emerge. Another notable characteristic of SENS is its novel mechanism for modeling physical environments. WSN applications feature tight integration of communication computation, and interaction with the physical environment. To give users with the flexibility of modeling the environment and its interaction with applications at different levels of detail, SENS defines an environment as a grid of interchangeable tiles [51].

SENS is less customizable than many other simulators, providing no opportunity to change the MAC protocol, along with other low-level network protocols. It does appear to use one of the most sophisticated environmental models and implements the use of sensors well. However, the only phenomenon detectable is sound. It can be argued as to whether or not it is better for a simulator to support several aspects well, or to support one phenomenon exceptionally well.

4.6 COOJA (COntiki Os JAva)

COOJA is a Java application, all interactions with compiled Contiki code are made through Java Native Interface (JNI). It is a simulator for the Contiki node operating system. MSPSim can be integrated into COOJA, forming COOJA/MSPSim. It provides simultaneous cross-level simulation at application, operating system and machine code instruction set level [52]. COOJA links low-level simulation of sensor node hardware and simulation of high-level behavior in a single simulation. COOJA is adaptable and scalable to replace all levels of the system: operating system software, sensor node platforms, radio transceivers, and radio propagation models. As network communication is central to a WSN, the COOJA Simulator supports adding and using different radio mediums [53].

Initially, COOJA was a code level simulator for networks consisting of sensor nodes running on Contiki OS. Sensor nodes with different simulated hardware and different onboard software may co-exist in the same simulation. Code level simulation is achieved by compiling Contiki core, user processes, and individual simulation glue drivers into object code native to the simulator platform, and then executing this object code from COOJA. It can run the Contiki programs in two different ways: either by compiling the program code directly on the host CPU, or compiling it for the MSP430 hardware. It can simulate sensor networks simultaneously at different levels, including the operating system level and the network (application) level [54].

Nevertheless, due to its scalability, the simulator has relatively low efficiency. Simulating several nodes with many interfaces which requires a lot of calculations, especially when plugins are started and registered as observers to those interfaces. It supports a limited number of same node types; the simulator has to be restarted once and a while if the number of nodes exceeds permissible limit. A test interface GUI is also missing, thus making extensive and time-dependent simulations difficult [55].

4.7 Castalia

Castalia was developed in C++ at the National ICT Australia [31]. It is an application-level simulator for WSN based on OMNeT++. It can be used to evaluate various platform characteristics for particular applications, since it is very parametric, and can simulate a wide range of platforms.

In Castalia, sensor nodes are implemented as compound modules, consisting of sub-modules that represent, for instance, network stack layers, application, and sensor. Node modules are connected to the wireless channel and physical process modules [56]. It is a generic simulator with a more realistic wireless channel and radio model based on measured data. Since it is based on the OMNeT++ platform, it can be used by researchers and developers who want to test their distributed algorithms and/or protocols in a more realistic wireless channel and radio models, with a practical node behavior mainly relating to access of the radio [57].

Features of Castalia include: sensing device bias, physical process modeling, node clock drift, and several MAC and routing protocols implemented. Castalia has a highly tunable MAC protocol and a scalable parametric physical process model. Different physical process modules in Castalia represent various sensing devices (e.g., temperature, light, acceleration, pressure, etc.,). Castalia can consider sensing device noise, bias and node clock drift [57].

As Castalia is not sensor-platform specific, it is intended to give a generic framework for the first order validation of any algorithm before moving to its implementation in a particular sensor platform. It is not useful if one would like to test code compiled for a specific sensor node platform.

4.8 Shawn

Shawn is a Java-based, open-source customizable sensor network simulator, designed to support simulation of large-scale networks. Rather than simulating a phenomenon, Shawn is designed to simulate the impact of the phenomenon. It claims to provide the highest abstract level and supports a more extensive network as compared to other conventional simulators such as NS-2, OmNeT++, SENSE, TOSSIM, and GloMoSim. However, more details on simulating WSN with Shawn can rarely be found in the literature. Its key features are persistence and decoupling of the simulation environment by introducing the concept of Tags. They attach both persistent and volatile data to individual nodes [58]. However, Shawn does not consider detailed simulations on the issues such as radio propagation properties or low-layer matters.

4.9 EmStar

Emstar is a Linux-based framework which allows a range of runtime environments from classic simulation to actual deployment. EmStar was intended to be compatible with two different types of nodes. Like the other emulators, it can be used to develop software for Mica2 motes. It also allows support for developing software for iPAQ based microservers. EmStar gives a flexible environment for transitioning between simulation and deployment for iPAQ-class sensor nodes running Linux. Users have three options: running many virtual nodes on a single host with a simulated network, running many virtual nodes on a single host with each virtual node bridged to a real-world one for networking, or running a single real node on a host with a network interface [59].

EmStar gives a choice to interface with actual hardware while running simulation. The simulation model is component-based and uses a discrete event model. It has a half simulation/half-emulation approach similar to SensorSim's, where software is running on a host machine and interfacing with the actual sensor. This allows for the use of the real communication channel and sensors. EmStar code may be run on a various set of execution platforms, each run the identical code and use the

same configuration files, which makes it simple for developers to iterate among all modes seamlessly. It combines a number of the stronger features of other simulators and emulators. EmStar's use of the component-based model allows for fair scalability [60].

EmStar uses a straightforward network medium and environmental model. As the goal is to transfer the code to a real sensor environment, simple environment and network medium abstractions served for the developers. Secondly, the simulator will only run code for the types of nodes that it is designed to work with. Also, if a user is attempting to model a system by progressing through the development cycle, it must either ensure that the hardware configuration being used matches the configuration file, or it must bear in mind that there will be differences and compensate appropriately, if necessary. Emstar does not support parallel simulations and lacks algorithms that are reactive to real dynamics, seen in physically situated sensors.

4.10 J-Sim

J-Sim is a general purpose simulator modeled after NS-2, developed at the university of Washington by the National Simulation Resource [61]. Just like NS-2, J-Sim is also written using two programming languages; however, for J-Sim, they are Java and Jacl, a Java version of Tcl. Energy modeling, except for radio power consumption, is also appropriately provided for sensor networks. Unlike NS-2, however, J-Sim uses the concept of components, replacing the notion that each node should be represented as an object. J-Sim uses three crucial components: the target node (which produces stimuli), the sensor node (that reacts to the stimuli), and the sink node (the ultimate destination for stimuli reporting). Each component is broken into different parts and modeled differently within the simulator. The breakdown of each component makes it easy to use different protocols in different simulation runs [62].

J-Sim features many improvements on NS-2 and other similar simulators. It has component-based architecture, which scales better as compare to the object-oriented model used by NS-2 and others. Moreover, J-Sim has an improved power model and also the ability to simulate the use of sensors for event detection [63].

J-Sim is, however, comparatively complicated to use. J-Sim, yet, more scalable than several other simulators, also suffers its share of inefficiencies. The only MAC protocol that can be used in J-Sim is from 802.11.

4.11 VisualSense

VisualSense provides modeling and simulation framework for WSN that build on Ptolemy II [64]. The extension to Ptolemy consists of a few new Java classes and some XML files. VisualSense gives an explicit and scalable radio model, which is based on a general energy propagation model. The energy propagation model can be reused for physical phenomena. Wireless networks modeling requires advanced representation and interpretation of communication channels, ad-hoc networking protocols, media access control protocols, energy consumption in sensor nodes, localization strategies, etc. VisualSense supports the component-based construction of such models [65].

It supports actor-oriented network nodes, physical media such as acoustic channels, wireless communication channels, and wired subsystems. Its software architecture is based on the set of

base classes for defining channels and sensor nodes. It is mainly designed to facilitate the research community to share models of disjoint features of the sensor nets problem and to build models that incorporate sophisticated elements from several aspects. However, VisualSense does not implement any protocols above the wireless medium.

4.12 (J)Prowler

Prowler is Matlab based simulator, thus it provides a fast and easy way to prototype applications, and has better visualization capabilities but (J)Prowler is written using Java [66]. (J)Prowler is mainly for Berkeley MICA Mote hardware platform running an application built on TinyOS, though it can also be modified to simulate more general systems. Prowler and (J)Prowler are probabilistic WSN simulators. Prowler is an event-driven simulator that can be set to operate in either deterministic mode or probabilistic mode . The deterministic mode is used to generate replicable results during testing of the application. While the probabilistic mode is used to simulate the non-deterministic nature of the communication channel and the low-level communication protocol of the motes. It can incorporate an arbitrary number of motes, on arbitrary (possibly dynamic) topology, and it was designed to easily embed into optimization algorithms. (J)Prowler provides only the default MAC protocol of TinyOS.

5 Analysis

Building a comprehensive WSN simulator is and will likely remain a challenging task. For WSN simulators, two features are challenging: reproducible experimentation and dynamic environment modeling. The key features and limitations of each of simulators mentioned above are highlighted in Table 2.1.

Simulator	Programming language/ (Platform)	Key features	Limitations
NS-2	C++	Availability of a visualization tool. Easy to add new protocols. A large number of protocols available publicly.	Supports only two wireless MAC protocols, 802.11, and a single-hop TDMA protocol.
TOSSIM	nesC	High degree of accuracy Availability of a visualization tool.	Compilation steps lose the fine-grained timing and interrupt properties of the code.
GloMoSim	Parsec	Parallel simulation capability. It is tailored specifically for wireless networks. Availability of a visualization tool.	Effectively limited to IP networks because of low level design assumptions. Unavailability of new protocols.

Avrora	Java	Can handle networks having up to $10,000$ nodes. Enables validation of time-dependent properties of large-scale networks	Fails to model clock drift. Cannot model mobility.
SENS	C++	Platform-independent. Users can assemble application-specific environments.. Defines an environment as a grid of interchangeable tiles.	Not accurately simulates MAC protocols. Provides support for sensors, actuators, and physical phenomena only for sound.
COOJA	Java (Simulations in C)	Concerning both simulated hardware and software. Larger-scale behavior protocols and algorithms can be observed.	Supports a limited number of simultaneous node types. Makes extensive and time-dependent simulations difficult.
Castalia	C++	Physical process modeling, sensing device bias and noise, node clock drift, and several MAC and routing protocols implemented. Highly tunable MAC protocol and a flexible parametric physical process model.	Not a sensor specific platform. Not useful if one would like to test code compiled for a specific sensor node platform.
Shawn	Java	Not limited to the implementation of distributed protocols. Can simulate vast networks.	Radio propagation properties or low-layer issues are not well considered.
EmStar	Linux	May be run on a diverse set of execution platforms. Combination of simulator and emulator. EmStar's use of the component-based model allows for fair scalability.	Code only runs for specific type of nodes. Does not support parallel simulations.
J-Sim	Java	Provides support for energy modeling, with the exception of radio energy consumption. Support mobile wireless networks and sensor networks. Component-oriented architecture.	Low efficiency of simulations. The only MAC protocol provided for wireless networks is 802.11. Unnecessary run-time overhead.
VisualSense	Ptolemy II	Provides an accurate and extensible radio model as well as a sound model that is accurate enough to use for localization.	Does not provide any protocols above the wireless medium, or any sensor or physical phenomena other than sound.

(J)Prowler	Matlab/Java	Probabilistic WSN simulators. (J)Prowler provides an accurate radio model.	Provides only one MAC protocol, the default MAC protocol of TinyOS.

Tableau 2.1: Key features and limitations of some popular WSN simulators

6 CupCarbon:Wireless Sensor netwrok Simulator (SCI-WSN)

Most of the existing simulators (as mentioned in the previous section) are mainly used to develop routing protocols. However, They do not offer real-time interference and radio propagation modeling for smart cities and IoT applications. The main idea behind proposing CupCarbon is to keep simulation time short while taking into account a realistic evaluation of the wireless interference in a $3D$ environment with an accurately simulated radio channel. The CupCarbon platform is developed with the following primary objectives:

- To study the deployment of WSN by considering the mobility and the availability of the radio spectrum.
- To simulate and analyze the performance of a proposed WSN in a 2D/3D environment.
- To study the feasibility and reliability of communication in the network.
- To detect high radio interference zones in the network.
- To simulate the radio propagation accurately in a real-time urban environment.
- To better visualize the simulation results to debug and validate a developed algorithm.

Cupcarbon is a WSN simulator specially designed for Smart Cities and Internet of Things applications. Its objective is to design, visualize, and validate the proposed algorithms for environmental monitoring, data collection, etc. It offers following two simulation environments [67]:

- The first simulation environment allows the design of mobility scenarios and the generation of natural events such as fires as well as the simulation of mobiles like vehicles and flying objects (e.g. UAVs, insects, etc.).
- The second environment describes a discrete event simulation of WSN which considers the scenario designed on the basis of the first environment.

Any WSN can be prototyped with its intuitive graphical interface using OpenStreetMap (OSM) framework. The sensor nodes can be directly placed on the map. In addition, each sensor node can be individually configured by its command-line with the script called SenScript. From SenScript, it is also possible to generate codes for hardware platforms such as Arduino/XBee. In CupCarbon, the energy consumption of the nodes can be calculated and graphically displayed as a function of the

simulation time.

Moreover, it enables the visibility and impact of wireless interference and signal propagation. CupCarbon can simulate the PHY layers based on ZigBee, LoRa and Wi-Fi protocols (Which is also discussed in this document). CupCarbon offers the simulation of scenarios and algorithms on several levels. For example, on the first level, it will determine the nodes of interest in the given area on the map, followed by a level to analyze the nature of the communication between these nodes to accomplish a given task (e.g., detection of an event). On the final level, describing the nature of the data routing to the base station if an event is detected. CupCarbon also allows configuring the nodes dynamically so that the nodes can be split into separate networks or merge in different networks.

CupCarbon represents the main kernel of the ANR project PERSEPTEUR that aims to develop algorithms for an accurate simulation of the propagation and interference of signals in a $3D$ urban environment.

6.1 CupCarbon Plateform

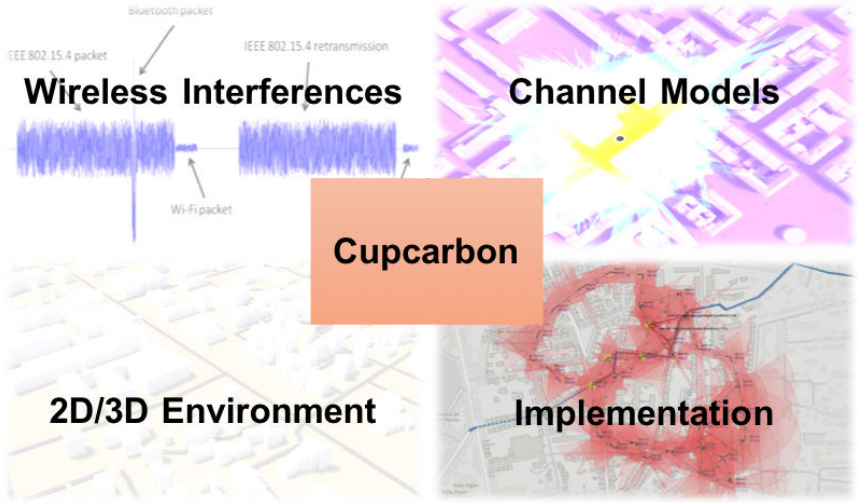


Figure 2.3: Main components of the CupCarbon platform.

Figure 5.1 shows the CupCarbon plateform, which is composed of four main parts [68].

Interference block

This block is the main contribution of this thesis. It can be further divide further into two catagories:

1. PHY Layer: a significant originality of CupCarbon is to be able to implement the true physical layers of some of the LP-WAN communication protocols like ZigBee, Wi-Fi, and LoRa. The evaluation of a link quality can now be based on accurate transmission conditions that consider the radio channel and the data encoding. This approach, however, can be very time consuming

if we need to generate all the signals in the city. That is why we have also proposed and included some statistical approaches to model interference.

2. Interference: interference is a significantly limiting factor in dense networks. Figure 2.4 shows the overlap between the Wi-Fi, Bluetooth, and Zigbee spectrum. This shows the overlap of the channels for these three protocols which operate in the $2.4\ GHz$ band. If a ZigBee device is in the same area as several Wi-Fi networks on channels 1, 6, and 11, it would experience bad channel conditions due to interference. There are a few ways to get around this issue, by either increasing the allocated energy, or moving to another frequency spectrum, or by setting some rules for the interfrence avoidence [69].

   In Figure 2.5, a ZigBee packet is transmitted in an environment with Bluetooth and WiFi sources of interference. The short Bluetooth signal corrupts the ZigBee packets which need to be re-transmitted. An exact evaluation of interference with these protocols, is feasible and CupCarbon allows a generation of all the received signals. However, such an approach can become very time consuming due to a possibly high number of interfering sources: signals have to be generated, channels to be simulated and the addition of all the received contributions creates the interference. To avoid too heavy calculations, it is possible to randomly generate the interference. However, as shown in [70][28], a Gaussian model is not accurate. One solution is to accurately evaluate the main contributions of the interference and to randomly generate the global contribution of the less impacting interferers. Another possibility is to generate the global interference with a single distribution. Our proposal is to use α-stable distributions with parameters depending on the radio channel statistics and the interferer density.

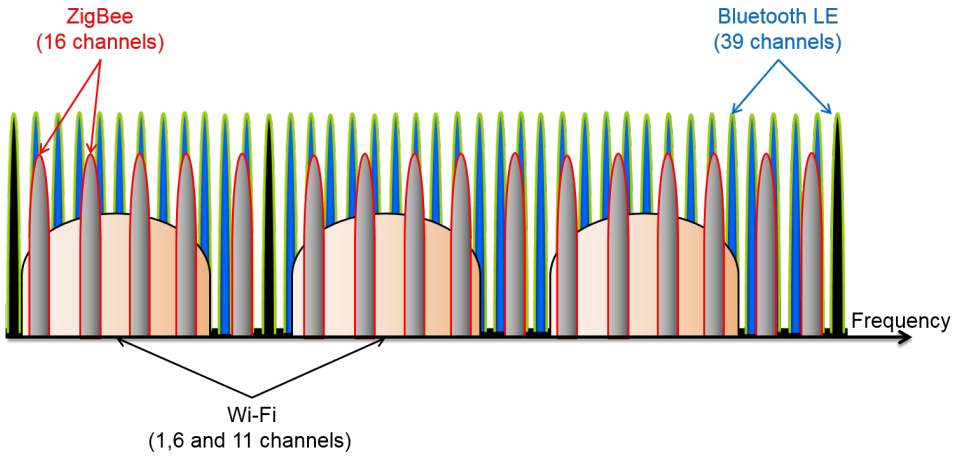


Figure 2.4: Zigbee, Wi-Fi and Bluetooth interference in 2.4 GHz frequency band.

2D/3D Environment block

The $2D/3D$ visualization of the city or the network area is essential, before the deployment of the nodes. The $3D$ environment helps to obtain an accurate deployment in which the elevation can be taken into account. Figure 2.6 shows an example of a city displayed in the 3D environment of

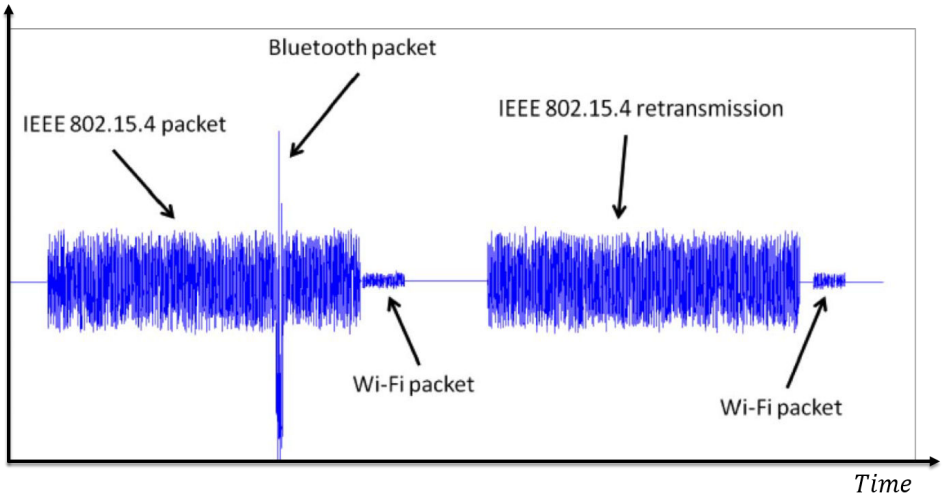


Figure 2.5: Zigbee, Wi-Fi and Bluetooth packet interference.

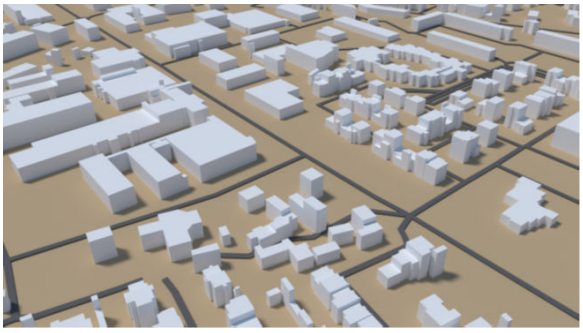


Figure 2.6: Example of a 3D display of a city in CupCarbon.

the CupCarbon simulator. The 3D environment of CupCarbon is composed of ground elevation, buildings and various objects like sensor nodes.

Radio channel block

Two radio propagation models are integrated into CupCarbon. The first one is a $2.5D$ based on a point to zone acceleration structure called visibility tree [71] [72]. It provides the estimation of channel attenuation and the channel Impulse Response (IR) according to a large number of the receivers. The second model is a full 3D ray-tracing associated with a Monte-Carlo algorithm.

Implementation block

The platform was created with a modular structure to simplify the replacement and customization of specific parts of the simulator. The modular structure has two properties. The first is to simplify the target architecture. The second is to promote multiple implementations of a given module, which enables users to switch rapidly and efficiently between different versions.

6.2 Architecture of CupCarbon

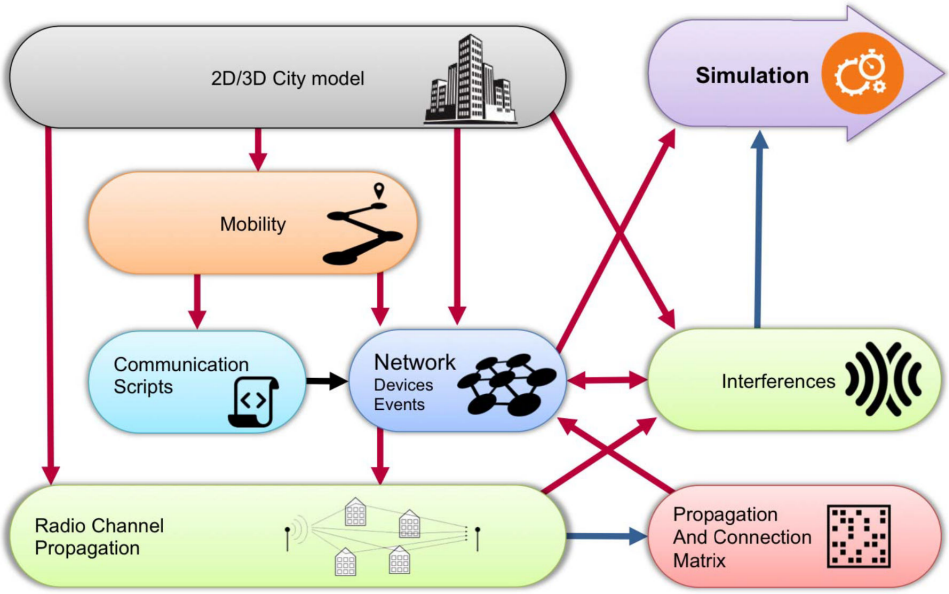


Figure 2.7: Main components of the CupCarbon architecture.

Figure 2.7 shows main modules of the CupCarbon simulator. Each can be defined as follows:

- **2D/3D City model module:** represents a digital format of a city (or the area of interest for the network deployment). It contains the information about the buildings, roads, green spaces, other obstacles, etc. It is the first part of the simulator that is presented to the user. It allows the deployment of the nodes directly on the map and also calculates the wireless interference and the signal propagations.

- **Mobility module:** allows creating the routes for the mobiles. A mobile can be a device with or without any communicating system and a sensor or a sensor node. The mobility of any device/node can be prefixed, where the mobile follows a given trajectory. The mobility route can also be defined in the script based on a given situation (e.g., detection of a target, an abnormal sensed value, etc.), which allows performing intelligent mobility.
- **Network module:** allows designing the WSN.
- **Communication Script module:** acts as the interpreter of the SenScript, which is used to program each node in the network. The simulator will execute the instructions of the script of each node.
- **Radio channel propagation module:** used to calculate the channel attenuation and its impulse response for a group of nodes in the network. Resulting data helps to determine the quality of the wireless channel between groups of the nodes and is used to decide whether a communication link can be established or not.
- **Interference module:** used to allow the user to allocate the different communication protocols to the nodes in the network. This is where the PHY layer is implemented. Then it determines for each sent packet whether it is received or not by the receiver. Interference between the nodes can be seen, tackled and modeled at this level though either Gaussian or α-stable models. Radio congested areas on the map can choose the impulsive interference model(α-stable model), and for the less radio congested areas, Gaussian interference model can be choosen.
- **Simulation module:** is the kernel of the proposed architecture. It is based on discrete event simulation. The events are generated either by executing each instruction of the script of each node or by a real event (e.g., mobility) or a natural event (e.g., temperature, gas, etc.).

3

LAN Protocols for CupCarbon

The widespread use of mobile computing devices, including personal digital assistants (PDAs), laptops, wearable computers has increased the demand of creating new either wireless personal area networks (WPAN) or local area networks (LAN). The proximity of these wirelessly connected devices with each other often results in interference due to shared $2.4\ GHz$ frequency band. $2.4\ GHz$ band is license free radio spectrum, widely known as industrial, scientific, and medical (ISM) radio frequency band. This part of the radio spectrum is intended to reserve for the operations of scientific, medical and industrial use rather than for communications. Despite the original purpose of ISM band, rapid growth can be seen in its use in short range, low power communications like WPAN, LAN, etc.. When several technologies share increasingly crowded ISM band, simultaneous transmission in the same sub-band may become problematic. Especially, for the low power wireless networks, neighbouring devices transmitting at comparatively high power may cause the decrease in the packet delivery rates and link quality. Interference has been accepted as one of the major problems for performance degradation to the wireless networks and wireless sensor networks [18].

The primary upshot of interference is increased packet loss rate, which often results in increased network traffic flow due to packet retransmissions. Interference can also cause high latencies and unpredictable medium access contention times. This cannot be ignored in WSN for some applications like industrial automation and control [73], high-confidence transportation systems [74], and health care [75], etc., where unreliable connections are intolerable and high packet delivery rates with limited delay bounds are required. Mainly, interference can be divided into three types:

- Inter-modulation,
- electrical, and
- radio frequency interference.

There is minimal literature available on inter-modulation interference. It can be defined as a direct result of nonlinearities of passive components [76]. Usually, this type of interference caused by some strong signals, whose frequency not near to the frequency of that wireless system. These signals overload circuit in the receiver. The circuit can produce the generation of signal harmonics. These harmonics get mixed or combined in the receiver to generate a new frequency which was not there initially, at the receiver input. This newly generated frequency creates interference with the wireless system just like any other interference and called intermodulation interference. Electrical interference is caused by digital equipment like computers, lighting systems, faulty electrical devices, heavy electrical equipment, etc. The impact of electrical interference on the reception of the signal relies on the proximity of the electrical device (which generates electrical interference) to the wireless

access point. Advances in wireless technologies have significantly reduced this impact on wireless transmissions. Also, electrical interferences are comparatively rare and account for the small share of all wireless interference problems. Electrical noise has further three forms:

- noise generated by electronic devices, e.g. computers,
- noise from natural sources, e.g. lightenning, and
- noise from electrical resources, e.g. poor quality wiring.

Wireless medium is a shared medium, and the increase in wirelessly communicating devices introduce new research topics on interference. This exponential increase is driven by users desire to access information from handheld devices. On the other hand, it is also stressed due to the recent incorporation of wireless communications in some applications concerning home networks, public safety, healthcare, and vehicular networks, etc. However, the efficiency of wireless systems is limited by the phenomenon of radio frequency interference. Because of the shared medium, if nearby transmitters transmit at the same time, using same or adjacent frequency band, their signals will interfere with each other, causing a collision. Collision can result in loss of all colliding packets and loss of resources. To be troublesome, the interfering signal is not necessarily to be on the exact frequency as the wanted signal. Strong RF signals with the nearby frequency band can also affect the reception of signals in the wireless systems.

To avoid the radio interference, wireless systems are designed with keeping in mind that interference is intrinsically harmful to wireless communications and must be avoided. However, complete mitigation of interference is not possible, especially when the wireless systems use unlicensed frequency band (e.g., 2.4 GHz) [77][78]. From the beginning of radio transmissions, many methods have been used to tackle interference, and they can be grouped into following three categories:

- radio spectrum licensing,
- efficient modulation, power control methods, and
- signal processing techniques at the receiver.

Since in this work, we have only considered unlicensed bands so, the first interference avoidance category is out of the question here. However, in this chapter and the next chapter, we shall discuss the other two categories. Depending on the necessary coverage and location, a number of different networks can be utilized for the smart cities. In next sub-sections, we shall talk about IEEE standards 802.11 and 802.15.4.

1 Single-carrier vs Multi-carrier Systems

In a typical Single-carrier system, the transmitter uses the entire bandwidth to send the data stream. The transmitter shapes the data symbols by a pulse, modulate it to RF and move through an analog filter. The receiver uses matched filtering, equalizing techniques to remove frequency selectivity, and a demodulator. With all its simplicity at the transmitter, the single-carrier systems are more complex at the receiver and give a limited performance. Frequency selectivity is one of the biggest limiting factors in single-carrier systems, due to the multipath nature of the channel. At the receiver, equalizers are used to correct frequency selectivity, but as the data rates increase, the complexity of the used adaptive algorithms also increase. Moreover, due to the multipath nature of the channel,

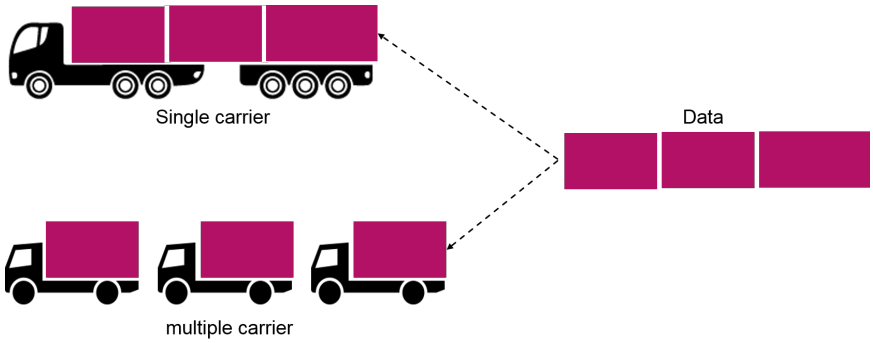


Figure 3.1: Basic concept of single carriers and multi-carrier system

the associated delay spread can cause intersymbol interference (ISI). The ISI occurs if maximum delay τ_{max} is less than symbol time T_s, $(\tau_{max} < T_s)$.

A single carrier system like FDM modulates data on one carrier by adjusting phase or amplitude of the carrier. Signals from multiple users are transmitted simultaneously using multiple frequencies. Each user is assigned a different frequency and each frequency range is called a sub-carrier, and it is modulated separately by different data stream. A guard band is placed between sub-carriers to avoid signal overlap. Bandwidth used in such case is higher as the duration of a single bit becomes very small. So, in such systems, there may be greater loss of information due to impulse noise, signal reflections, and other noise factors. These additions of noise in the system slow down the ability of the system to recover the information. In addition, as the bandwidth of the carrier increases, the interference from the other sources, which are sending information continuously also increases [79].

On the other hand, the basic idea of multi-carrier modulation (MCM) is to divide the bit stream into a number of different bit streams and also divide the total bandwidth into N number of sub-carriers, to send the divided bit stream over them. This can relate to a transport company using several smaller transportation trucks (multi-carrier), instead of one big truck for the same amount of data, as shown in Figure 3.1. MCM has been used in many wired and wireless applications, including digital broadcasting of audio and video in Europe [80], Digital Subscriber Lines (DSL) and the most recent Wireless LANs. Other emerging applications include fixed wireless broadband services [81], mobile wireless broadband known as FLASH-OFDM [82], and even for ultra-wideband radios, where multiband OFDM is one of the two competing proposals for the IEEE 802.15 ultrawideband standard. MCM can be implemented as Vector Coding and OFDM. Both of these techniques have slight differences, but the basic idea of these techniques is the same. MCM has many advantages over single-carrier as it has the ability to cope with severe channel conditions such as attenuation, narrowband interference and frequency-selective fading [80]. Multi-carrier modulation is the most commonly used communication technique due to its capability to attain relatively good reception quality notwithstanding of frequency selective fading. The first MCM systems introduced were military HF radio that came in the market in late 1950 [83].

At higher data rates, the channel distortion of the data is much important, and it is somewhat impossible to recover the transmitted data with a simple receiver. A very complex receiver structure is needed for such a system, which equalizes and estimates channel with very high accuracy. OFDM can simplify the problem of equalization by converting frequency selective channel to several flat

faded subchannels. Multiplexing generally refers to dividing data among different sub-carriers that are overlapping and orthogonal to each other. The basic concept of OFDM is to divide the available bandwidth into a number of independent sub-channels that are orthogonal to each other, in this way each narrow sub-channel experiences almost flat fading [79].

1.1 IEEE 802.11 Standard and Formats

IEEE standard 802.11 provides set of specifications for media access control (MAC) and physical layer (PHY) for implementing wireless local area network (WLAN) in a frequency band of $900MHz$, $2.4GHz$, $3.6GHz$, $5GHz$ and $60GHz$ [84]. The family of IEEE 802 standards deal with metropolitan area networks (MAN) and local area networks (LAN). It is maintained by an individual working group called IEEE 802 LAN/MAN Standards Committee (LMSC). The IEEE 802.11 specifically deals with LAN. IEEE 802.11 provides a set of MAC and PHY specifications for communication in wireless local area networks that have been introduced over many years. The IEEE 802.11 is a series of modulation techniques with the same basic protocol (Table 3.1). The recent versions of the 802.11 standards have more capacity and faster than its previous versions. These different versions can differ in transmission ranges, speed, frequency used but are similar in implementation. All standards can use either adhoc network design or an infrastructure based. These standards give the basis to Wi-Fi brand for wireless network products. The radio frequency spectrum used by 802.11 differs between countries. 802.11 mainly uses two frequency bands: $2.4\ GHz$ and $5\ GHz$. It divides both frequency bands in different ways. For example, the $2.4\ GHz$ band is divided into 14 sub-channels (also shown in Figure3.2), channel number $1,\ 6$ and 11 are non over lapping channels. Channels are spaced $5\ MHz$ apart. The $5\ GHz$ band is much bigger than the $2.4\ GHz$ band, and also supports more channels. Availability and number of channels are dependent on the country and the region [85].

Two ways to handle the wideband disadvantages are proposed in IEEE 802.11. The initial versions (including IEEE 802.11b) were based on code division multiple access (CDMA). CDMA is a multiple access scheme where several users share the same physical medium. Each user is assigned a distinguished code for transmission. CDMA is explained in more detail in the Section 3. To increase the data rate IEEE 802.11 introduces use of OFDM. In next section we will disscuss the latest proposed modulation technique OFDM.

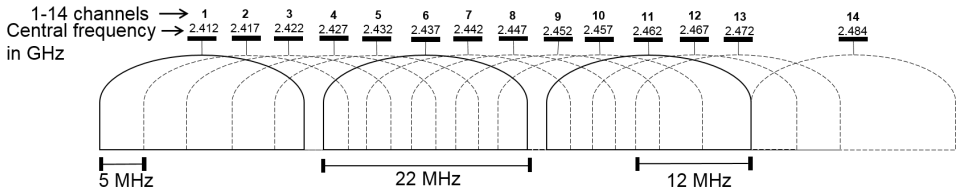


Figure 3.2: Graphical representation of channels for 2.4 GHz frequency band

Tableau 3.1: 802.11 standards and their specifications

Release Date	Standard	Frequency Band (GHz)	Bandwidth (MHz)	Modulation	Maximum Data Rate
1997	802.11	2.4	20	DSSS, FHSS	Mbits/s
1999	802.11b	2.4	20	DSSS	11 Mbits/s
1999	802.11a	5	20	OFDM	54 Mbits/s
2003	802.11g	2.4	20	DSSS, OFDM	542 Mbits/s
2009	802.11n	2.4 , 5	20 , 40	OFDM	600 Mbits/s
2013	802.11ac	5	40 , 80 , 160	OFDM	6.93 Gbits/s

2 Orthogonal Frequency Division Multiplexing (OFDM)

Most of the 802.11 standards (a, g, n, and ac) use orthogonal frequency division multiplexing (OFDM) as their modulation technique. OFDM is a way to encode data on multiple sub-carriers. OFDM is a multi-carrier modulation technique, where data symbols modulate parallel collection of equally spaced sub-carriers [80]. In OFDM available bandwidth is divided along each sub-carrier equally and each sub-carrier operates at a different frequency and these sub-carriers are orthogonal [86]. OFDM is a combination of modulation and multiplexing technique. In OFDM, multiplexing is applied to the signals, which are independent and are subset of the main signal. Initially, the signal is split into independent symbols, modulated and then re-multiplexed to create the OFDM symbol [86]. In OFDM, inverse discrete fourier transform (IDFT) is used to convert the spectra in the time domain. After the modulation, all carriers are transmitted to fullly occupy the bandwidth. The spectral efficiency of OFDM system as compared to FDMA and depicted in the Figure 3.3. The overlapping multicarrier technique can achieve superior bandwidth utilization. There is a huge difference between the conventional non-overlapping multicarrier techniques such as FDMA and the overlapping multicarrier technique such as OFDM. Because of long symbol periods, OFDM provides resilience to inter-symbol interference (ISI) and multipath propagation effects. Cyclic extention is also used with OFDM symbols to completely eleminate the adverse effects of ISI [87].

2.1 Principle of OFDM

OFDM is an advanced form of FDM where the frequencies multiplexed are orthogonal to each other and their spectra overlap with the neighbouring carriers. As shown in the Figure 3.4, the subcarriers never overlap for FDM. In contrast to FDM, OFDM is based on the principle of overlapping orthogonal sub-carriers. The comparison between the spectral efficiency of OFDM system and FDMA

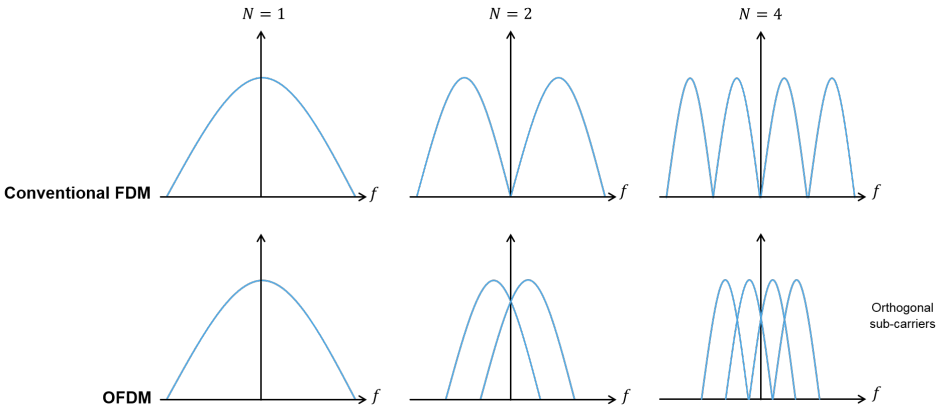


Figure 3.3: Spectrum efficiency of OFDM

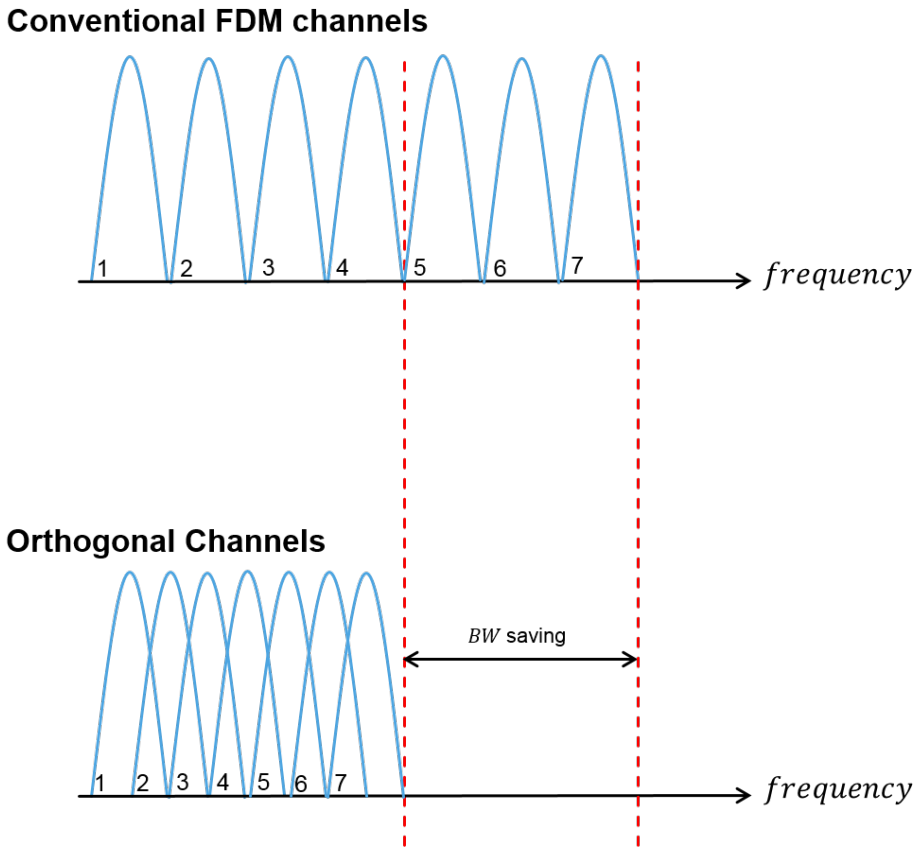


Figure 3.4: orthogonal signals concept

system is depicted in Figure 3.3. OFDM overcomes some of the disadvantages offered by FDMA and TDMA. As in OFDM available bandwidth is divided into a number of narrowband sub-channels (typically 100 to 8000) and these sub-channels are very closely spaced, there is no need of overhead due to cyclic prefixing as is used in case of FDMA and sub-channels need not to be multiplexed as in the case of TDMA [79]. The orthogonality of the carriers means that each carrier has an integer number of cycles over a symbol period. Due to this, the spectrum of each subcarrier has a null at the center frequency of each of the other subcarriers. This results in almost zero interference between the carriers, allowing to be spaced as close as theoretically possible. This overcomes the problem of overhead carrier spacing required in FDMA [79].

In the OFDM system, the number of sub-carrier not only depends on available bandwidth, but also on the IFFT size, $(No\_subCarriers \leq \frac{size\_IFFT}{2} - 2)$. The higher the FFT size is, the higher the number of sub-carriers are, the more complex is the system, and higher the transmission rates can achived [88]. Some of the advantages gained by applications of OFDM are listed below [89]

- OFDM is simple to implement. One thing that makes OFDM a very low cost technology is its ability to allocate bits to unique carrier through IFFT.
- Frequency selective fading in OFDM is very small as compare to single carrier system.
- Much simpler structure of receiver is needed in OFDM for equalization due to channel effect is simplified at the transmitter, using a single tap equalizer.

OFDM also have some shortcommings too, like:

- Peak-to-Average Power Ratio (PAPR): Peak to average power ratio makes the digital to analog converter and analog to digital conversion very difficult and is directly proportion to the number of sub-carriers. So as the PARP increases, designing becomes complicated [79].
- Strict Synchronization Requirement: OFDM is highly affected by errors because of time and frequency synchronization, specially the errors when frequency is not synchronized and de-modulation of OFDM signal frequency errors gives high bit error rate due to Inter Carrier Interference [79]. The sources of synchronization error are two:

    1. the difference between oscillator frequencies in transmitter and receiver, and
    2. Doppler spread due to relative movement of transmitter and receiver.

Local oscillator frequencies at transmitting and receiving point must match each other as closely as possible. Matching should be more accurate for a higher number of sub-channels. Movement of transmitter and receiver also cause the frequency errors. So the performance of OFDM degrades for an object moving at high speed. Therefore to improve the performance of OFDM link, synchronization between transmitter and receiver must be very accurate. Synchronization is done in three stages; symbol, carrier frequency and sampling frequency [79]. $2.4\ GHz$ frequency operation band is most common in many extensions of IEEE standard 802.11, with 14 distinct channels. Figure 3.2 shows these 14 channels from 1 to 14 spaced $5\ MHz$ apart.

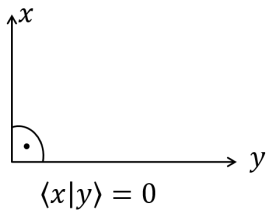


Figure 3.5: Orthogonality of two vectors

Orthogonality

The term orthogonal signifies that there is a particular mathematical connection between the sub-carrier frequencies. For orthogonality, sub-carriers must be organized as, so that the side lobes of the individual sub-carriers overlap and the signals are received with no neighboring carrier interference. For this, the sub-carriers must be mathematically orthogonal. Because of orthogonality, transmitter can transmit a lot more sub-carriers simultaneously with a relatively smaller gap, and inter-channel interference can be avoided [90]. The basic idea of orthogonality is similar to the CDMA, in which we use codes to make data sequences independent to each other and allow many different users to transmit data using same channel [91]. Mathematically, two signals are considered to be orthogonal to each other, if the dot product between them is zero over a time period (as depicted in Figure 3.5). Two sinusoid signals with frequencies that are integer multiples of a common frequency can satisfy this orthogonal condition. Therefore, orthogonality can be defined as:

$$\int_0^T cos(2\pi n f_0 t) cos(2\pi m f_0 t) dt \;=\; 0 \quad ; \qquad (m \neq n) \tag{3.1}$$

where T is the time period or the symbol period, m and n are two different integers, and f_o is the central frequency.

Orthogonality allows the signals to be transmitted over a common channel and detected without interference. Loss of orthogonality results in blurring between the transmitted signals and loss of information. For OFDM signals, the peak of one sub-carrier coincides with the nulls of the other sub-carriers. This is shown in Figure 3.6. Thus there is no interference from other sub-carriers at the peak of a desired sub-carrier even though the sub-carrier spectrums overlap. OFDM system avoids the loss in bandwidth efficiency prevalent in system using non orthogonal carrier set.

2.2 Packet Format

Figure 3.7, shows PHY frame format for IEEE 802.11. The PHY frame contains preamble, header, and MAC data. This frame format is compatible with previous 802.11 sub-standards working in $2.4\ GHz$ band. The preamble is used for the synchronization purposes and the header is used for the control/management purposes. PHY frame consists of:

- **SYNC:** Contains 128-bits of alternative $0's\ and\ 1's$, to alert the receiver for a potential frame. After detecting the SYNC, the receiver begins the synchronization process with the incoming packet.
- **SFD:** Start of the frame delimiter contains value $[1111\ 0011\ 1010\ 0000]$ and mark the beginning of the frame.

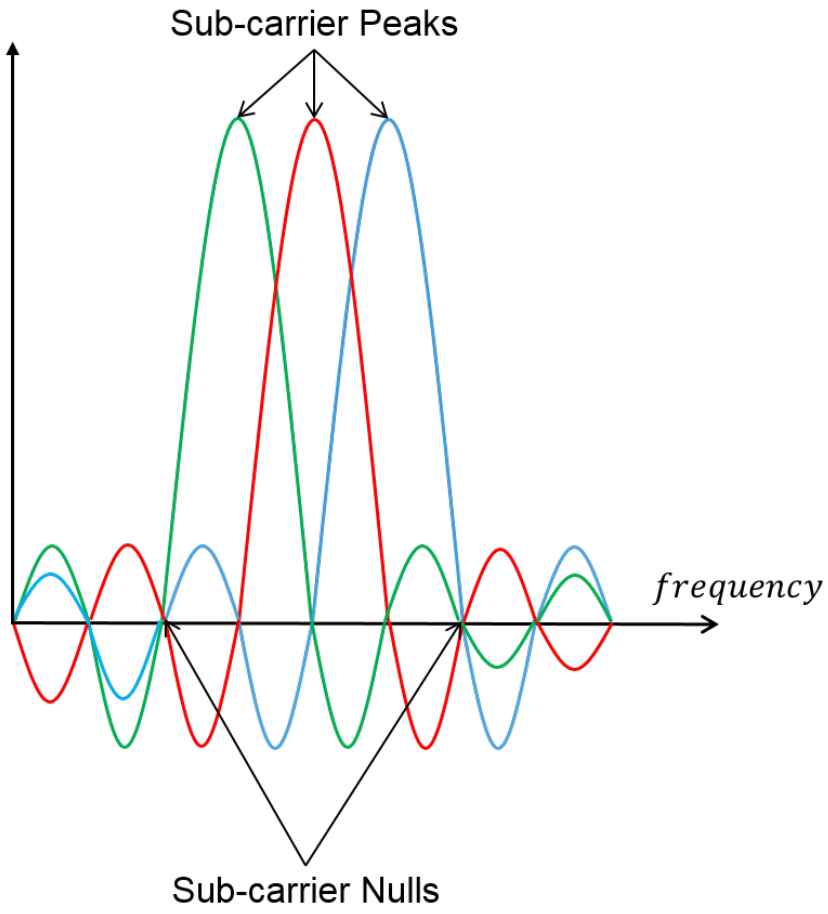


Figure 3.6: Orthogonal subcarriers in multicarrier systems (OFDM)

16 bytes	2 bytes	1 byte	1 byte	2 bytes	2 bytes	Max. 2304 bytes
PPDU						
PLCP Preamble		PLCP Header				MAC Frame
SYNC	Start of Frame Delimiter	Signal	Service	Frame Length	CRC	MPDU
D-BPSK Modulation		**D-BPSK Modulation**				**OFDM Modulation**

Figure 3.7: PHY encapsulation for IEEE 802.11

- **Signal:** It specifies the data rate used for MPDU. The header is always encoded at the fixed data rate to avoid the signal detection complexity. Only the MPDU field is encoded at the variable data rate.
- **Service:** This field specifies the modulation used to transmit the PSDU.
- **Frame length:** In case of OFDM, this field specifies the data length of the MAC frame. This information in combination with the service field is used by the receiver to estimate the time required for MAC data transmission. Service field identifies the modulation technique used for the payload. This way, the receiver can determine how long the channel/medium will be busy.
- **CRC:** Cyclic redundancy check of 2-bytes is used for error detection in the PHY header. The MAC layer does contain its own CRC.
- **MPDU:** MPDU is the MAC frame and contains the actual data.

2.3 Transceiver Scheme

Figure 3.8 shows the block diagram of the OFDM communication system. First, the data symbols $(d_{n,k})$ are assembeled and modulated over complex exponential waveform $\phi_k(t)$. $\phi_k(t)$ represents each baseband sub-carrier:

$$\phi_k(t) = \begin{cases} e^{j2\pi f_k t} & t \in [0, T_d] \\ 0 & otherwise \end{cases} \tag{3.2}$$

where $d_{n,k}$ is the symbol transmitted using k^{th} sub-carrier during n^{th} timing interval. T_d is the symbol duration and f_k is the frequency of k^{th} sub-carrier. The frequency for each sub-carrier is calculated as $f_k = f_0 + \frac{k}{T_d};\ k = 0, 1, 2,\ ... N-1$. f_0 is the lowest frequency and N is the maximum number of sub-carriers. After the modulation, OFDM symbols are transmitted simultaneously. The continuous time signal consisting of OFDM block will be:

$$x(t) = \sum_{n=-\inf}^{\inf} \left[\sum_{k=0}^{N-1} d_{n,k} \phi_k(t - nT_d) \right] \tag{3.3}$$

IFFT/FFT operation in OFDM helps to convert a frequency selective wireless channel into N parallel flat fading channels by dividing the large bandwidth that causes the ISI. At the receiver, two consecutive OFDM samples can interfere with each other which is called Inter Block Interference (IBI). A Cyclic prefix of length L is added to the serial data at this point to avoid IBI at the receiver. At the receiver, this cyclic prefix is discarded. This signal is transmitted over the wireless channel [92].

Symbol generation and Constellation Mapping

Typically, the input data is transmitted and received in form of a serial data stream. Serial-to-parallel conversion block is used to convert input bit stream to OFDM symbols. The number of bits in each symbol depends on the number of sub-carriers and modulation scheme. In this work, we are using $16-QAM$, where each sub-carrier carries 4-bits. The modulator maps the parallel bits to complex values that are also called constellation points and represent a modulated carrier, Figure 3.9. The amplitude and phase of the sinusoids depend on the input data. Data transitions are synchronized at each carriers, and may process together.

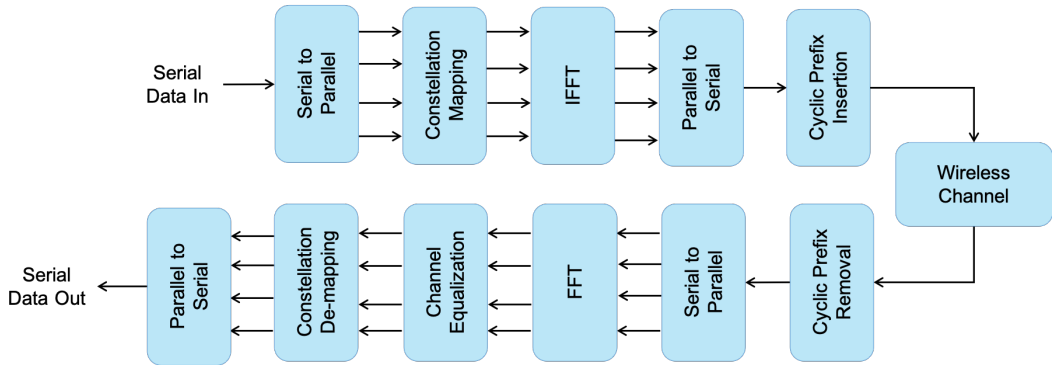


Figure 3.8: Bock diagram of an OFDM based communication system

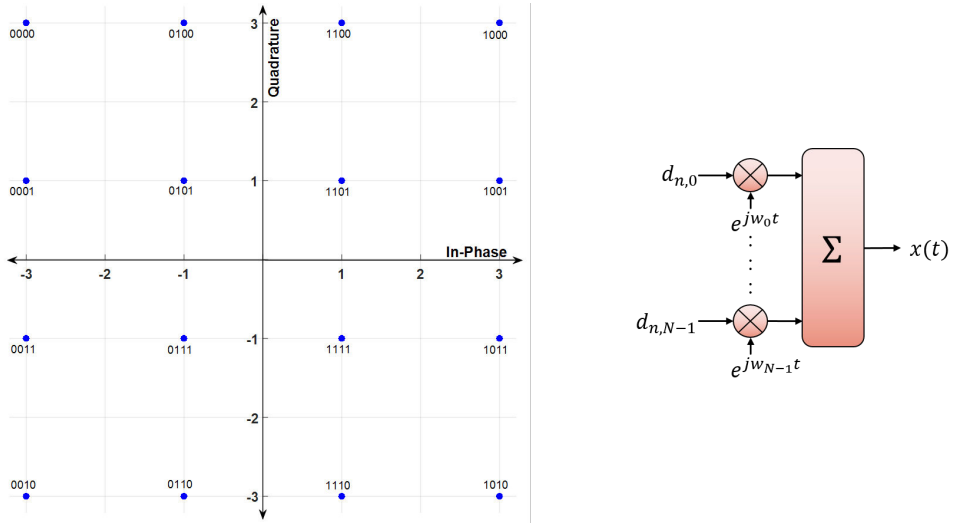


Figure 3.9: Scatter plot of $16 - QAM$

IFFT

N-point IFFT operation is performed on the OFDM symbols to generate OFDM signal. IFFT is used to transform the spectrum into time domain signal by converting complex data points into N time domain samples. It allows the sub-carriers to be transmitted simultaneously, without interfering with each other. The IFFT places each symbol on the respective sub-carrier:

$$X_k = \sum_{k=-N/2}^{N/2} x_n e^{(-j2\pi nk)/N} \quad ; \quad k = 0, 1, ..., N-1 \tag{3.4}$$

Resulting signals are then combined for transmission. The frequency separation makes the sub-carriers orthogonal and due to the orthogonality, they do not interfere with each other at receiver.

Cyclic Prefixing

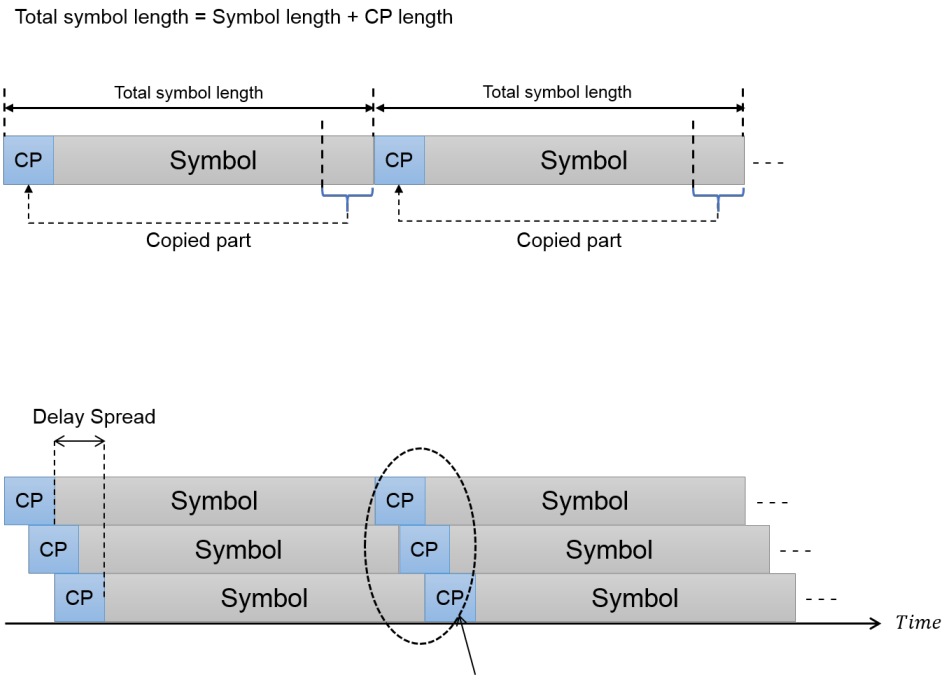


Figure 3.10: Cyclic prefixing concept in OFDM.

Difference between the time of arrival of first multipath signal and time of arrival of the last multipath signal is called delay spread of the system. The channel tends to spread the transmited signal in time domain, as the tail of one symbol is merged into the head of the subsequent symbol. This gives rise to interference between consecutive symbols, which is known as inter symbol interference (ISI). Channel dispersion also destroys the orthogonality between the sub-carriers and cause inter-carrier interference (ICI) for the signal. For this, guard intervals were proposed as the solution. A guard interval is an empty space between successive OFDM symbols and it serves as a buffer for the multipath reflection. Use of guard bands avoids ISI but cannot cope with ICI (the loss of the subcarrier orthogonality). This problem can be addressed by introducing cyclic prefix

(CP) instead of guard interval between successive OFDM symbols. A cyclic prefix is basically a repetition of the last few microseconds of the OFDM symbol that is added at the beginning of every symbol as shown in Figure 3.10. Cyclic prefix plays an important role in mitigating ISI and is used with discrete fourier transform to provide computationally efficient method of equalization [93].

As shown in the Figure 3.10, CP ensures that the delayed replicas of the OFDM symbols always have a complete symbol within the FFT interval. So that the transmitted signal is still periodic and this periodicity helps maintaining the orthogonality. In a fourier transform, all the resultant components of the original signal are orthogonal to each other.

At the receiver, cyclic prefix is removed before any processing. As long as the length of CP interval is larger than maximum expected delay spread, all reflections of previous symbols are removed and orthogonality is restored. The orthogonality can be lost if the delay spread is larger than length of CP interval.

2.4 Receiver

The receiver performs the almost exact and opposite operations of transmitter. After removing the cyclic prefix and converting from serial-to-parallel, the combined time domain signals are transformed into frequency domain by taking the N-point FFT. Here we assume that OFDM symbols are well-timed. In FFT operation, the spectrum bins corresponds to the individual sub-carrier. After the FFT, channel equalization is applied at the receiver.

Channel Equalization

For the estimation of the channel, OFDM employs the insertion of the known pilot symbols. They are used to aid the receiver for estimation of the wireless channel for each sub-carrier position. These pilot symbols are extracted at the receiver from the data signal and channel estimation is executed. The possible distortion inserted by the channel is reversed based on the estimated complex value per active sub-carrier.

3 IEEE Standard 802.15.4

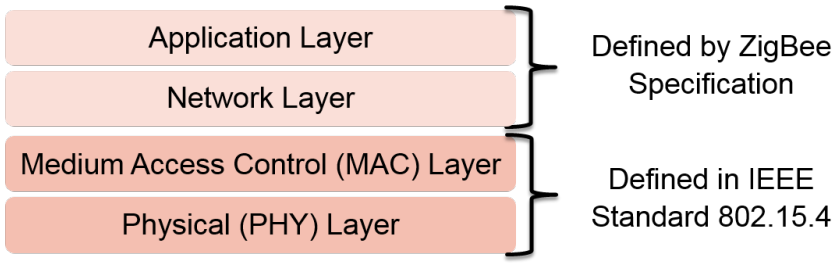


Figure 3.11: IEEE 802.15.4 and ZigBee protocol stack architecture

The IEEE standard 802.15.4 is for Low Rate Wireless Personal Area Networks (LR-WPAN). LR-WPAN includes home automation, healthcare monitoring, environmental surveillance, military

Tableau 3.2: IEEE 802.15.4 defined frequency bands and associated specifications

Frequency Band (MHz)	Chip rate (kchip/s)	Modulation	Bit rate (kb/s)	Symbol rate (ksymbol/s)	Symbols
868-868.6	300	BPSK	20	20	Binary
902-928	600	BPSK	40	40	Binary
2400-2483.5	2000	O-QPSK	250	62.5	16-ary Orthogonal

applications, smart cities and so forth. IEEE standard 802.15.4 defines the characteristics of physical (PHY) layer and medium access control (MAC) layer for the wireless communication systems that does not require high data rates. MAC layer manages access to the wireless physical medium, frame validation, guaranteed services, wireless node associations and security services of wireless networks [94]. IEEE 802.15.4 LR-WPAN assumes centralized and decentralized networks that allow wireless devices to communicate with other wireless devices within their radio transmission range [95]. For a peer-to-peer or decentralized wireless network, it defines the un-slotted carrier sense multiple access with collision avoidance CSMA/CA wireless medium access mechanism by which sensor nodes compete with each other to occupy the shared wireless medium for transmission [96]. In this work, we have considered that scattered sensor nodes follow peer-to-peer topology. The specifications for physical layer and medium access control layer of wireless networking are defined by IEEE standard 802.15.4. This standard does not have any requirements for higher networking layers in Open System Interconnection (OSI) model. IEEE standard 802.15.4 was developed by IEEE standards 802 committee and was initially released in 2003 [97]. The ZigBee standard defines only the networking, applications and security layers of the protocol and adopts IEEE standard 802.15.4 PHY and MAC layers as a part of the ZigBee networking protocol (see Figure 3.11). Therefore, ZigBee based devices are also adaptive to IEEE 802.15.4 as well.

IEEE 802.15.4 offers $2.4\ GHz$, $915\ MHz$ and $868\ MHz$ operational frequency bands. A total of 27 channels, numbered from 0 to 26, are specified by IEEE 802.15.4 across three unlicensed operational frequency bands. Sixteen channels are available in the $2.4\ GHz$ frequency band, ten in the $915\ MHz$ frequency band, and $868\ MHz$ frequency band based system occupies one channel (see Figure 3.12). Modulation techniques, chip rate and data rate information associated with each frequency band is given in Table 3.2 [96] [98]. In this work we have utilized ISM Frequency band of $2.4\ GHz$ with 16-channels and data rate of $250\ kbps$.

Network Topology

Based on the application requirements, IEEE standard 802.15.4 based wireless devices can adapt star topology and peer-to-peer topology for networking. Both are demonstrated in Figure 3.13. In star topology, sensor nodes follow more structured star pattern and a coordinator of the network will essentially be the central node to initiate, terminate or route flows. Star topology provides

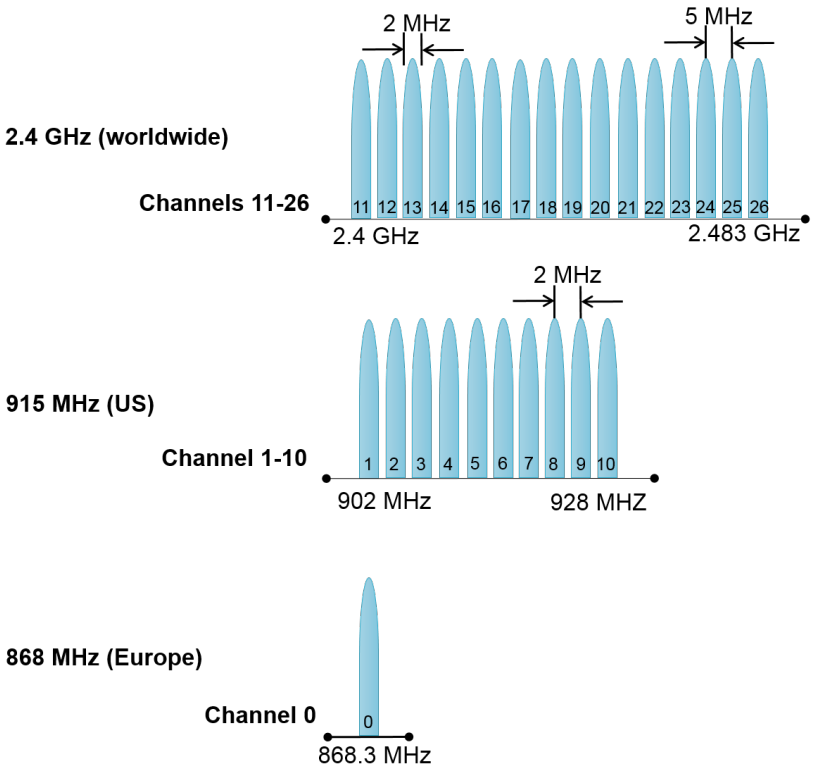


Figure 3.12: IEEE 802.15.4 operational frequency bands.

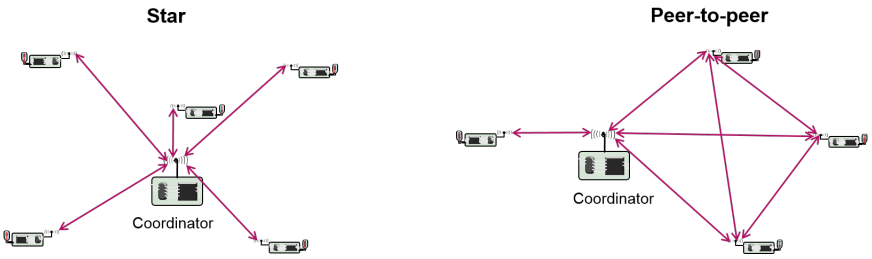


Figure 3.13: Star and peer-to-peer topologies.

<table>
<tr><td>4 bytes</td><td>1 byte</td><td>1 byte</td><td>2 bytes</td><td>1 byte</td><td>4 or 10 bytes</td><td>Variable</td><td>2 bytes</td></tr>
<tr><td colspan="8">PHY Frame</td></tr>
<tr><td rowspan="2">Preamble Sequence</td><td rowspan="2">Start of Frame Delimiter</td><td rowspan="2">Frame Length</td><td colspan="5">MAC Frame</td></tr>
<tr><td>Frame Control</td><td>Sequence Number</td><td>Addressing Fields</td><td>Frame Payload</td><td>Frame Check Sequence</td></tr>
<tr><td colspan="2">SHR</td><td>PHR</td><td colspan="3">MHR</td><td>MSDU</td><td>MFR</td></tr>
<tr><td colspan="8">PLCP Protocol Data Unit (PPDU)</td></tr>
</table>

Figure 3.14: Schematic view of the general frame format.

contention-free and contention-based wireless medium access to member nodes. The prominant short commings of star network topology are; it limits the overall distance that can be covered and it is limited to single hop.

Peer-to-peer topology allows each sensor node to communicate with other nodes within its radio range, in the network. Nodes can communicate with the network coordinator as well as to its peer nodes. It supports only contention-based (unslotted CSMA/CA) wireless medium access [99]. Peer-to-peer topology helps to form more complex network formations, like mesh networking topology. Some WSN applications such as, intelligent agriculture, industrial control and monitoring, etc., can benefit from such topology. Peer-to-peer network also allows multi-hop routing of messages from one node to the other in the network. In this work, we have considered that scattered sensor nodes follow peer-to-peer topology.

3.1 MAC Layer

The functions of the MAC layer are to access the network by using carrier sense multiple access with collision avoidance (CSMA/CA), to transmit beacon frames for synchronization, and to provide reliable transmission.

IEEE 802.15.4 Frame Format

In IEEE 802.15.4, frame structures have been designed to lower the complexity while at the same time assuring their robustness for transmission on wireless channel. Each successive layer of OSI model adds their layer-specific headers and footers. IEEE 802.15.4 only specifies PHY and MAC layers, other upper OSI model layers specifications are defined by ZigBee Alliance. IEEE 802.15.4 defines four frame structures [99]:

- **Beacon Frame:** originates by MAC layer and is used by the network coordinator to send beacons in a beacon-enabled network .
- **Data Frame:** originates from upper layers and used for the data representation.
- **Acknowledgment Frame:** originates from MAC and used for sending confirmation of successful frame reception at receiver.
- **MAC Command Frame:** originates from MAC and used for handling all MAC peer entity control transfers.

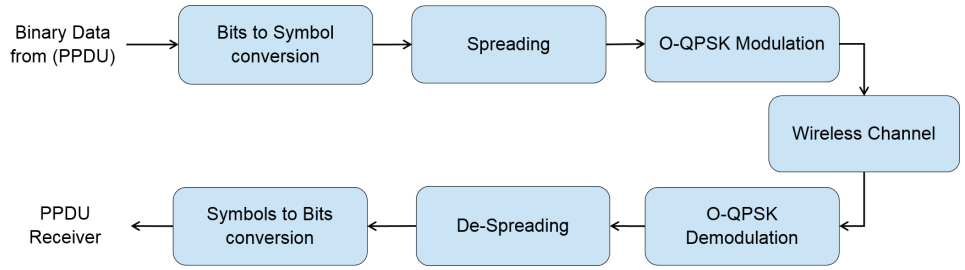


Figure 3.15: Functional block diagram of IEEE 802.15.4 transceiver.

Fig. 3.14 shows schematic representation of general frame at MAC and PHY layer. MAC layer frames contain MAC footer (MFR) of 2-bytes, MAC service data unit (MSDU) of variable length and MAC header (MHR) of 7-bytes to 13-bytes. MAC footer (MFR) contains frame check sequence of 16-bits. MAC frames are then passed to PHY layer and prefixed with 1-byte of PHY header (PHR) and 5-bytes of synchronization header (SHR) [99]. In this work our 802.15.4 data frame contains:

- **Preamble:** is used for synchronization purposes and contains 4 bytes of zeros.
- **SFD:** start of the frame delimiter along with preable is used for symbol synchronization and its value is 01111011.
- **Frame length:** this field mentions the length of the MAC frame. Mac frame length in this case is 22-bytes.
- **Data:** this field contains the actual data. For simplification, we have eliminated the MAC header.
- **FCS:** frame check sequence of 2-bytes is used for error detection and is calculated over MAC frame.

3.2 IEEE 802.15.4 PHY Layer

Figure 3.15 shows functional block diagram of IEEE 802.15.4 transceiver. It uses the Direct Sequence Spread Spectrum (DSSS) as spreading technique. Spreading techniques are utilized to increase the bandwidth of transmitted signal. DSSS phase shifts a sine wave pseudo-randomly with continuous string of pseudo-noise (PN) code symbols called "chips". This phenomenon is called symbol-to-chip mapping and helps to increase the transmitted power of the signal and decrease the interference influence on received signal. DSSS uses a signal structure in which transmitter produce PN-sequence and shares with receiver for reconstruction of the symbols. IEEE 802.15.4 standard has predefined these spreading codes (shown in Table 3.3). DSSS technique in IEEE 802.15.4 transceiver provides resistance to transmitted signal against intended or unintended jamming and allows sharing of a single channel among multiple users. System with frequency band $2.4\ GHz$ utilizes offset quadrature phase shift keying (O-QPSK) technique for chip modulation. Each 4-bits symbol is mapped to 32-bit pseudo random code, shown in Table 3.3. Digital systems operating on $868\ MHz$ and $915\ MHz$ frequency bands use 15-bit PN-sequence to map one symbol and Binary phase shift keying (BPSK) modulation technique [97].

Tableau 3.3: PN-Sequence for Symbol-to-Chip mapping for 2.4 GHz band system

Symbols (dec)	Symbols (bin)	PN-Sequence $(C_0, C_1, ..., C_{31})$
0	0000	11011001110000110101001000101110
1	0001	11101101100111000011010100100010
2	0010	00101110110110011100001101010010
3	0011	00100010111011011001110000110101
4	0100	01010010000101110110110011100011
5	0101	00110101001000101110110110011100
6	0110	11000011010100100010111011011001
7	0111	10011100001101010010001011101101
8	1000	10001100100101100000011101111011
9	1001	10111000110010010110000001110111
10	1010	01111011100011001001011000000111
11	1011	01110111101110001100100101100000
12	1100	00000111011110111000110010010110
13	1101	01100000011101111011100011001001
14	1110	10010110000001110111101110001100
15	1111	11001001011000000111011110111000

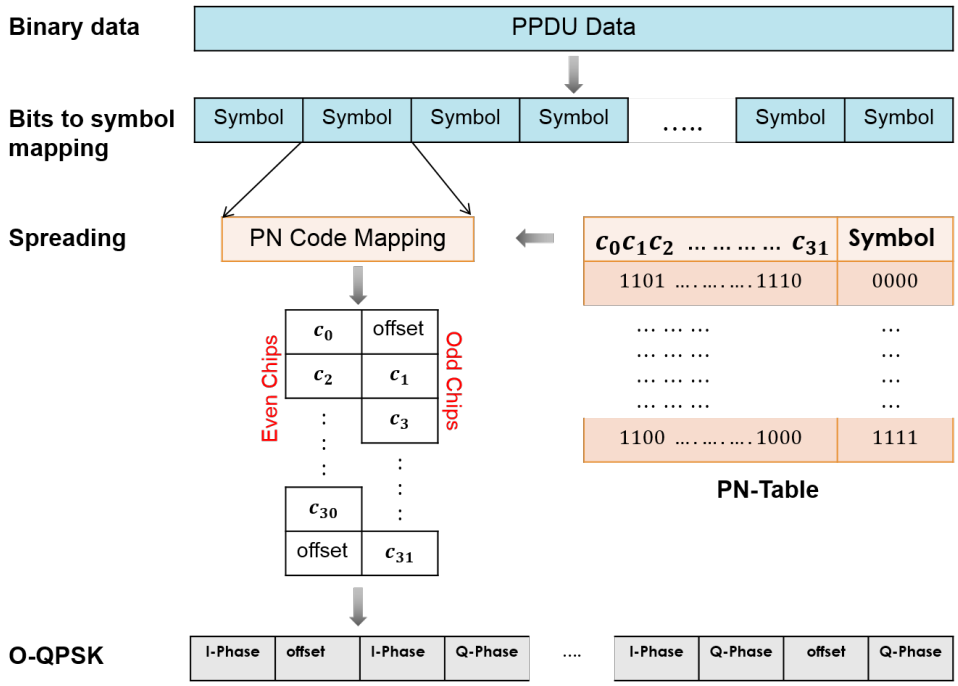


Figure 3.16: ZigBee transmitter operations.

Figure 3.16 shows working of IEEE 802.15.4 transmitter. At the transmitter the data from PPDU is divided into symbols. Each 4-bit symbol is spread using PN-sequence (shown in Table 3.3). These chips are then encoded using orthogonal quadrature phase shift keying (O-QPSK) modulation technique. O-QPSK is used in case of $24\ GHz$ system. For 1-byte binary data $(b_0b_1b_2b_3b_4b_5b_6b_7)$, first grouped to make 4-bit symbols like, $(b_0b_1b_2b_3)$ and $(b_4b_5b_6b_7)$. Then these symbols are replaced with the 32-bits PN-sequence or spreading codes, accordingly. Each even chip $(C_0C_2C_4\dots)$ is modulated as In-phase component of the carrier and each odd chip $(C_1C_3C_5\dots)$ is modulated as quadrature-phase component of the carrier. The chip $'1'$ is shaped using positive half-sine wave, and chip $'0'$ is shaped using negative half-sine wave. The time duration for each chip is $1\ \mu s$, so, I-phase and q-phase have timing offset of $0.5\ \mu s$, which results in continuous phase change (also shown in 3.17). The radio front-end, then, up-converts the baseband signal to $2.4\ GHz$ carrier for on air transmission.

At the receiver, for demodulation, received half-sine pulses are translated into a chips. Despreading is done by comparing received 32-bit chip sequence with each given 32-bit chip sequence from the PN-sequence table, which is already known at receiver. We have performed this comparison by applying XOR operation. The chip-to-symbol despreading block outputs the matching symbol which has smallest hamming distance (see Figure 3.18). Hamming distance provides the number of differ bits between received code and the each PN-sequence. In case of errors in the received sequence, the receiver mapps the received code on to the corresponding symbol of PN-sequence with minimum hamming distance. To lower the complexity of the system, setting a threshold of the Hamming distance can be valuable, as it can reduce the number of 32-bit XOR operations for each received sequence [100].

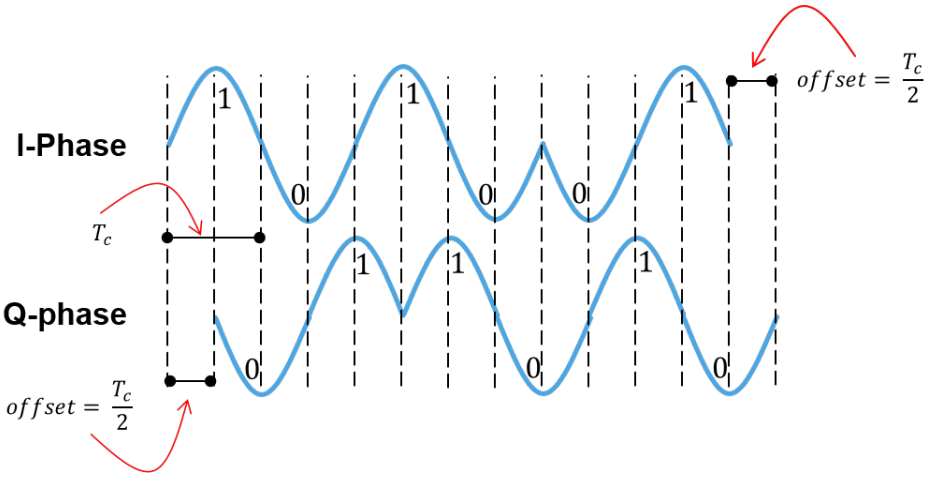


Figure 3.17: O-QPSK pulse shaping with half sinuside waveform.

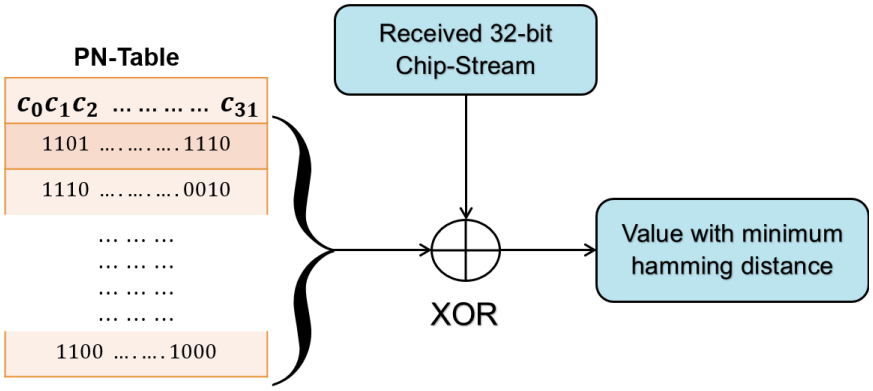


Figure 3.18: De-spreading.

4 Simulation Results

We have assumed that unwanted wireless interferer signals are following α-stable distribution. This distribution considers more realistic wireless channel environment by taking into account the events that are rare and have significant mass. Bit error rate (BER) can be calculated by comparing known transmitted bits with received bits. It is considered as performance analyzer of any digital communication system. We have analyzed the performance of proposed system in terms of its BER. Instead of assuming the parameters of α-stable distribution we have estimated two out of four

parameters based on real time wireless environment. The system parameters used here are based on IEEE standard 802.11 and 802.15.4. We have also embed this system into a wireless sensor network simulating tool called CupCarbon [35]. Simulations were repeated several times to obtain consistent results.

4.1 IEEE 802.11 Results

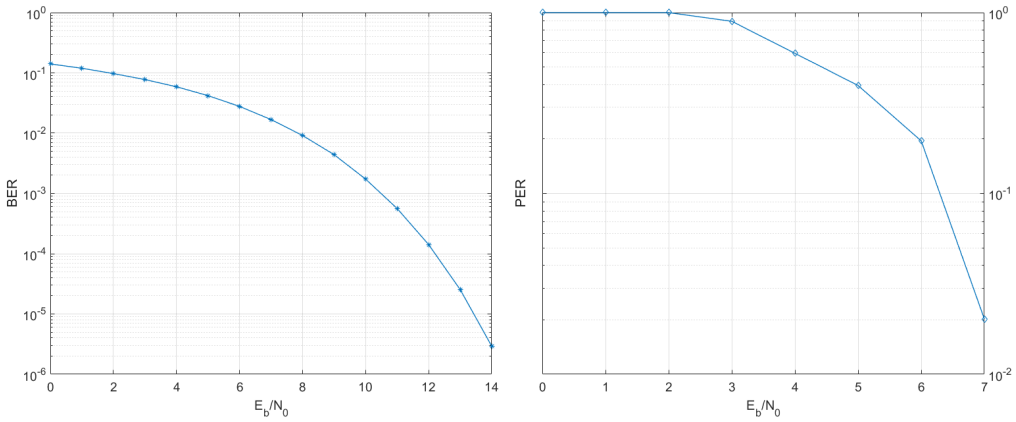


Figure 3.19: BER and PER performance of OFDM based PHY

Figure 3.19 shows the BER and PER of an OFDM system with Gaussian noise. The horizontal axis shows the E_b/N_0 (normalized SNR) in dBs. The vertical axis shows the bit error rate (BER) and packet error rate (PER). Monte Carlo simulations were performed for BER and PER. From Figure 3.19 it can be concluded that as the E_b/N_0 (energy per bit to noise ratio) increases in the system, the bit error rate decreases. Also, the AWGN based system performs much better in terms of BER as compared to α-stable based system.

Figure 3.20 shows the BER and PER performance of OFDM based PHY vs he scale/dispersion parameter σ. The α is 0.5, the skewness parameter β and location parameters δ are assumed to be $'0'$.

4.2 IEEE 802.15.4 Results

Fig. 3.21 shows the BER and PER performance of IEEE standard 802.15.4 based wireless communication system with Gaussian noise. PHY frames used in transceiver simulations, follow structure as given in Fig. 3.14.

Figure 3.22 and Figure 3.23 show the BER and PER performance of IEEE standard 802.15.4 based PHY with α-stable interference, consecutively. The α is used between 0.5 to 1.8. α is the index of stability which sets the degree of the impulsiveness of the distribution. The skewness parameter β and location parameters δ are assumed to be $'0'$. The scale/dispersion parameter σ which measures the spread of the noise samples around the mean, is calculated based on the λ. Vertical axis shows different bit error rate and packet error rate values. Horizontal axis shows different values for λ which is the spatial density of the network within the specified area around

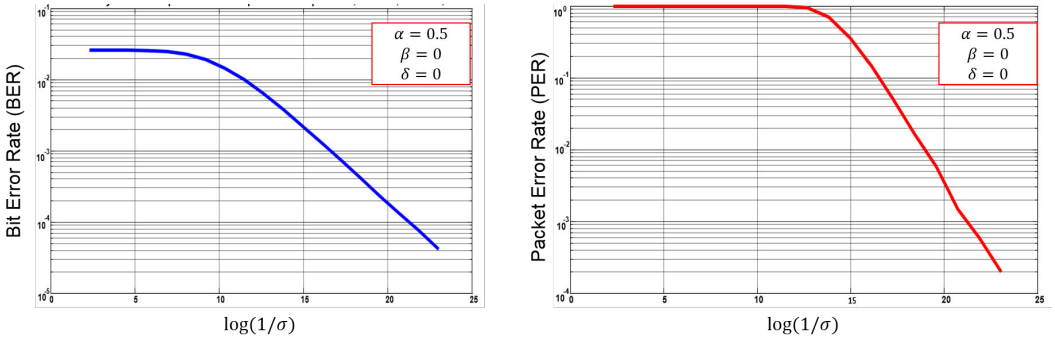


Figure 3.20: BER and PER performance of OFDM based PHY.

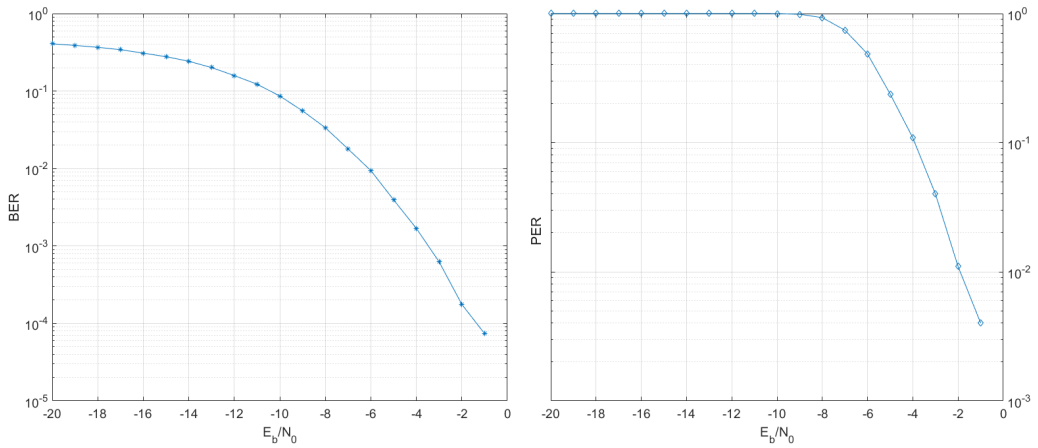


Figure 3.21: BER and PER performance of 802.15.4 based PHY with AWGN.

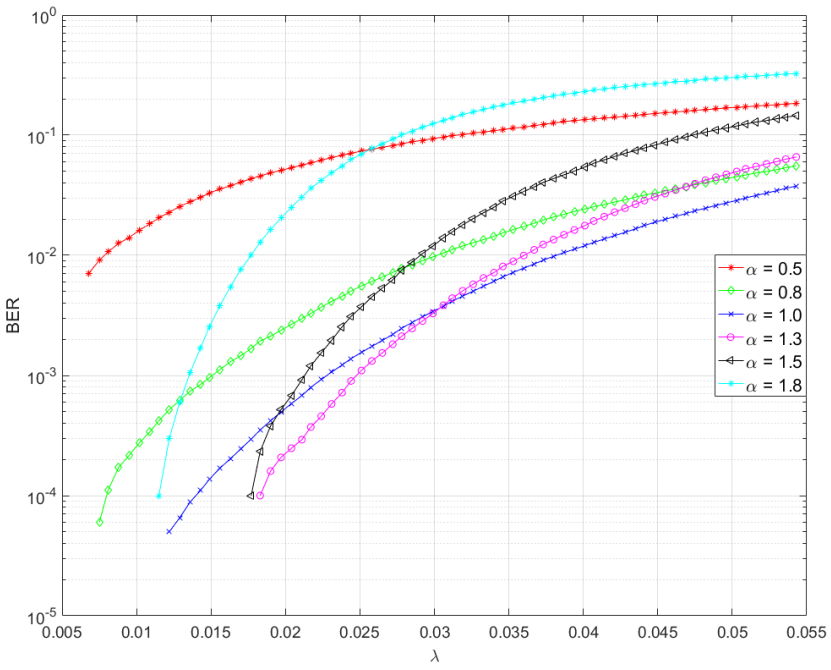


Figure 3.22: BER performance of 802.15.4 based PHY with α-stable.

a sensor node. As the spatial density of network increases, the BER and PER also increases. To calculate λ, we have varied the total number of wireless devices N in the network. We have consider that 12% wireless devices are active at any time interval. λ can be calculated:

$$\lambda = \frac{N}{\pi \times radio\_range^2} \times \frac{12}{100} \tag{3.5}$$

In CupCarbon, there is also propagation model integrated in it, which is based on the ray tracing model. The purpose of the propagation modelling is to obtain a good estimation of the field strength when some parameters are known, such as the frequency, antenna heights, the propagation environment, and so on [101]. In our context, due to the specific nature of the urban environments of the smart cities, we need site-specific or geometry-based propagation models such as ray-tracing (RT) models. This leads us to exclude the empirical/statistical models in this context, despite the fact that they are extremely fast and can be easily implemented.

Here, we have compared a statical model with the ray tracing model embedded in the CupCarbon. Figure 3.24 shows a map of small part of the city, the lines show the roads and also the path of the ten mobile nodes. One the node is the tranmitter and covers the path shown in dotted line. The other nine are the receivers and colored lines show the path they move. Figure 3.25 shows the average distance of eache receiver node from the transmitter. Based on this, we calculated the shadowing and attenuation (as shown in the Figure 3.26), as the function of distance.

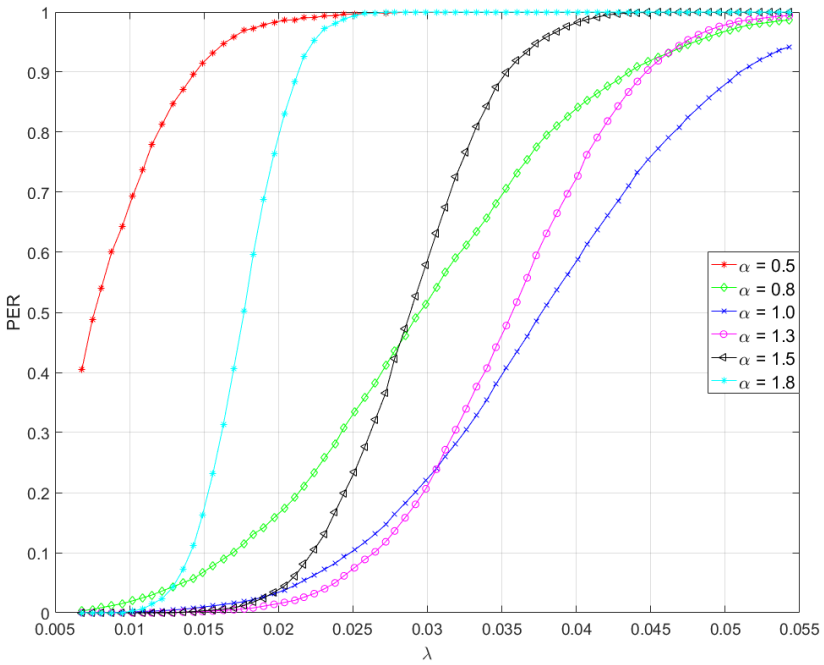


Figure 3.23: PER performance of 802.15.4 based PHY with α-stable..

Figure 3.24: Placement of nodes on the map.

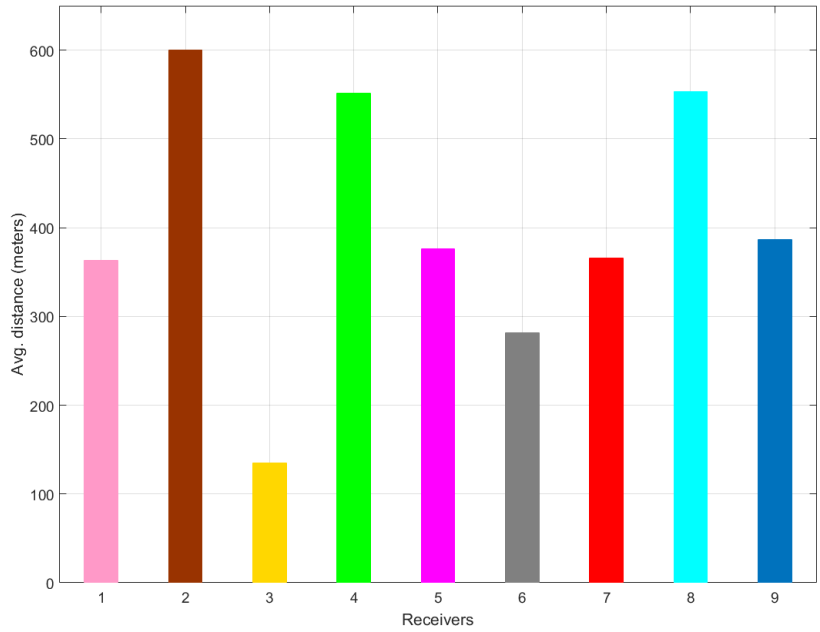


Figure 3.25: Average distance of each receiver node from the transmitter.

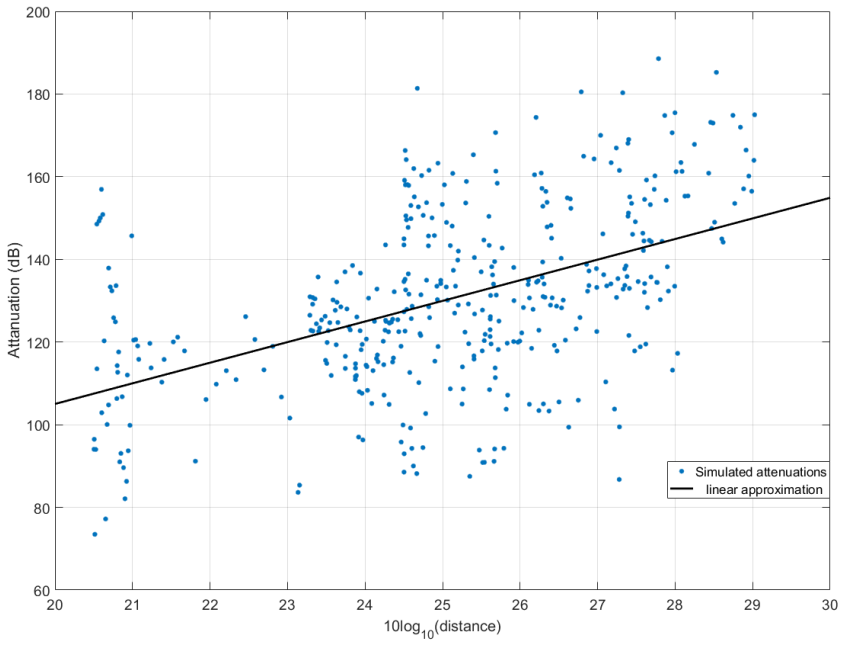


Figure 3.26: Attenuation as function of distance.

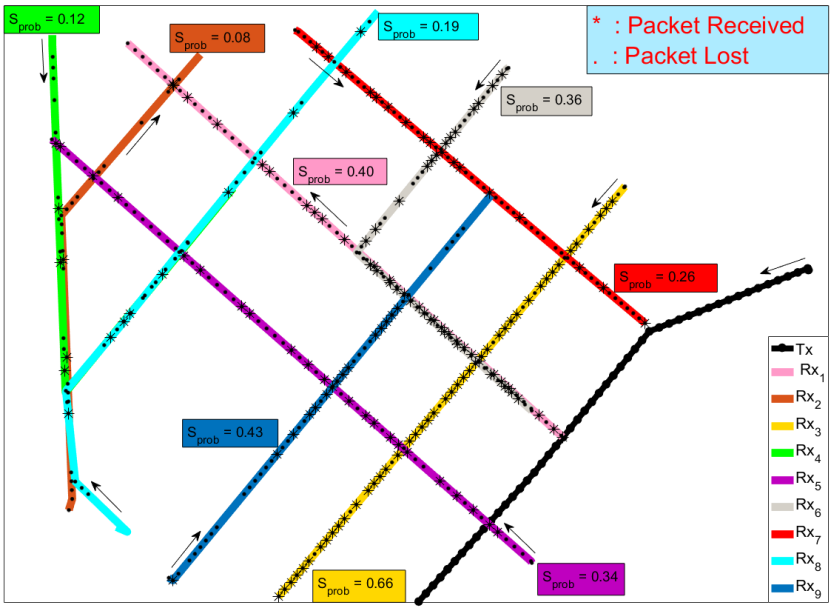


Figure 3.27: Packet received or lost with ZigBee with statistical model.

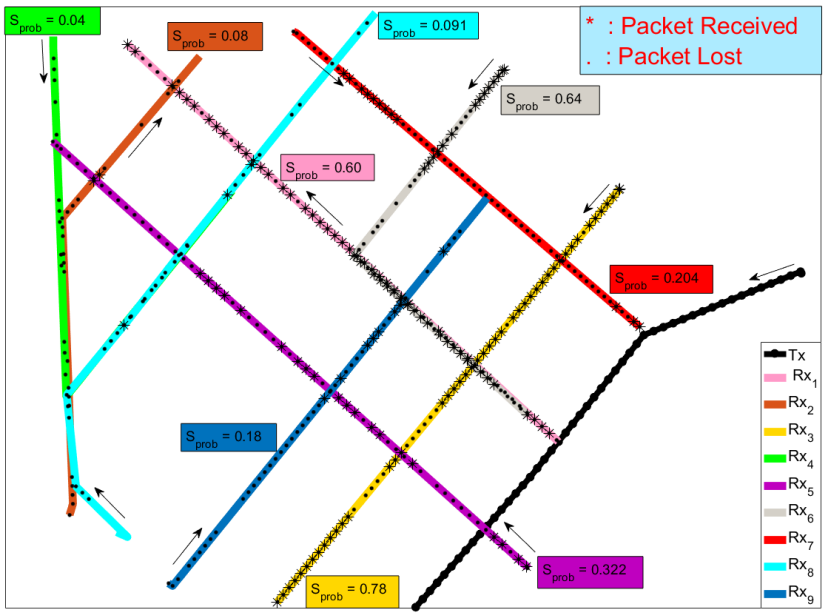


Figure 3.28: Packet received or lost with ZigBee with ray tracing model.

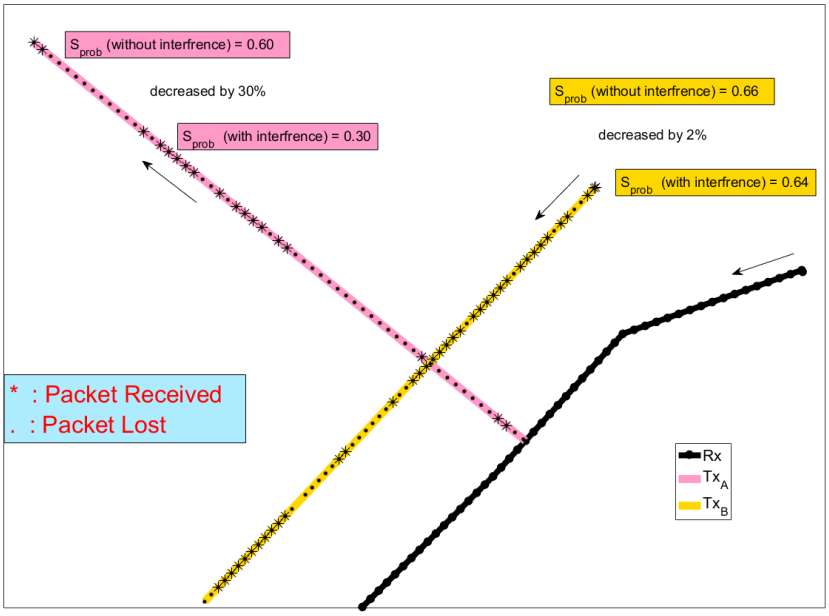


Figure 3.29: Packet received or lost with ZigBee with one interferer.

And if we use these statistical model of attenuation with ZigBee, the results are shown in the Figure 3.27. The results are also generated with the attenuation values which were found in the CupCarbon through ray tracing model (results are shown in Figure 3.28). The success probability is almost same in both cases (0.387% with statistical model and $0.0.3985\%$ with ray tracing model). But with the ray tracing model, it gives more accurate results of packet success at each point. S_{prob} is the probability of success of each node. Each transmission is done using the MAC which is defined in the next section. If a transmission is failed in first attempt the transmitter would retransmit the packet and keep trying for next three times. After three transmissions, if the packet is still not received then the packet will be considered lost. $'*'$ mark shows the packet received at that point, and $'.'$ mark shows the lost packet at that point on the map. Arrows on the both graphs show the the moving direction of the nodes.

Figure 3.29 and 3.30 show the results with the interference between two transmitters at the receiver. In Figure 3.29, both the transmitters are relatively strong but due the interfrence, their probability of of succes has decreased by 30% and 2% (this tranmitter is more close to receiver). Both transmitters and the receiver is mobile, and transmitters are also crossing each other. In these simulations, we have considered receivers as the transmitters and the tranmitter from the Figure 3.28 is considered as receiver. We have assumed the distance and attenuation in both direction is same. Same case is shown in Figure 3.30, here both transmitters are moving on the same path for a while. In this case, the success probability is droped by 20% and 17%.

Figure 3.31 and 3.32 show the results with the interference between three transmitters at the receiver. In these cases, the success probability is further decreased as the number of interferers are increased. In Figure 3.31, one node is crossed by two interferers, and its success probability is decreased by 21%. Same is the case in Figure 3.32, where the success probability is decreased by the 27%, 33% and 19%.

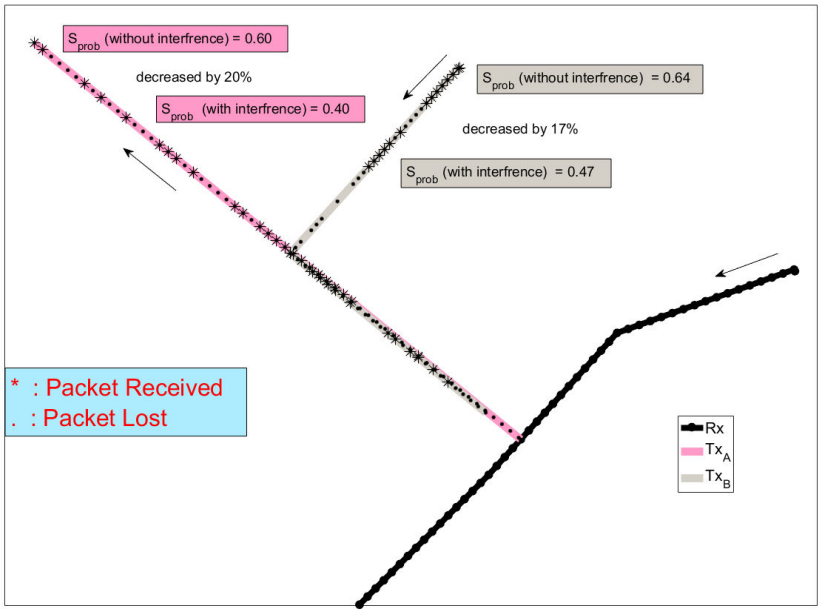


Figure 3.30: Packet received or lost with ZigBee with one interferer.

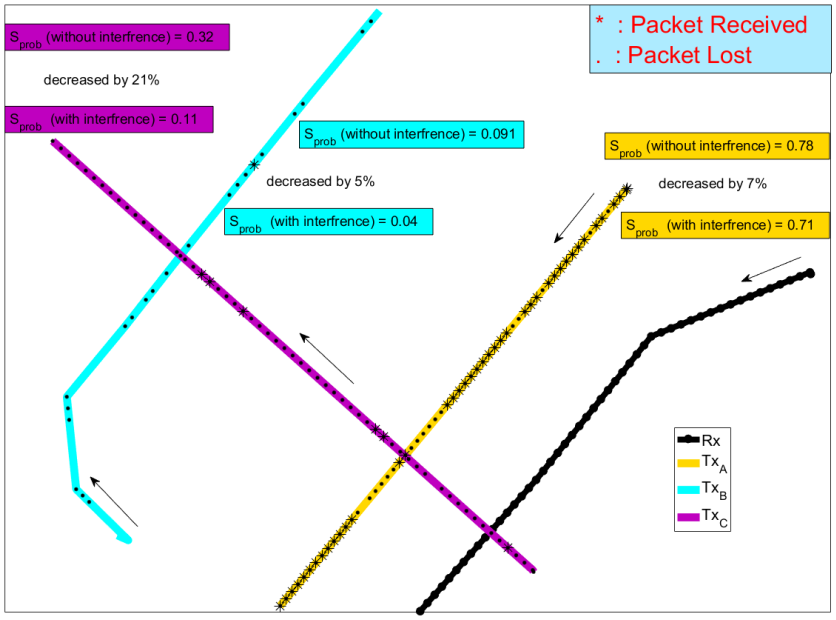


Figure 3.31: Packet received or lost with ZigBee with two interferers.

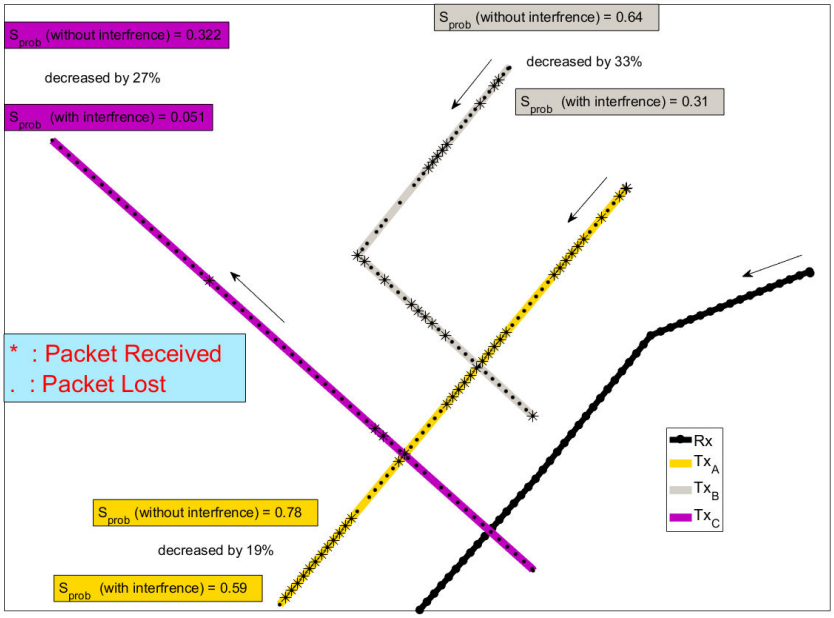


Figure 3.32: Packet received or lost with ZigBee with two interferers.

5 Integration of IEEE 802.11 and 802.15.4 in CupCarbon

To see the performances of the above interference models or the propagation models, the first step was to integrate PHY layers of the protocols that can be used in sensor network communications. An optimal way is required to integrate and implement these PHYs in a simulator. Simulation time and the run time complexity of the simulation are two of the main points to be conssider before the integration. After observing their results in MATLAB, PHY of 802.11 and 802.15.4 are added in simulation platform CupCarbon.

5.1 Integration of simplified MAC in CupCarbon

Figure 3.33 shows the flow chart to explain the mechanism that is used in CupCarbon. Where N shows the number of times a transmitter can resend the data packet to receiver node. S is the waiting time for the acknowledgement from the receiver node and i is a loop variable. A node will transmits data to another node in the network and will wait for $S\ sec$ for ACK response. If transmitter node receives the ACK message from the receiver node in $< Ssec$ then it will either continue the communication by sending next data packets or exit the communication. Otherwise transmitter node will resend its previous data packet assuming that it was lost in first place. Transmitter node can re-transmit the data packet to the receiver for maximum of $N = 3$ times and will wait for the ACK response for $S\ sec$ after each transmission. If ACK message from the receiver will not receive at transmitter node then the receiver node will considered as a dead node.

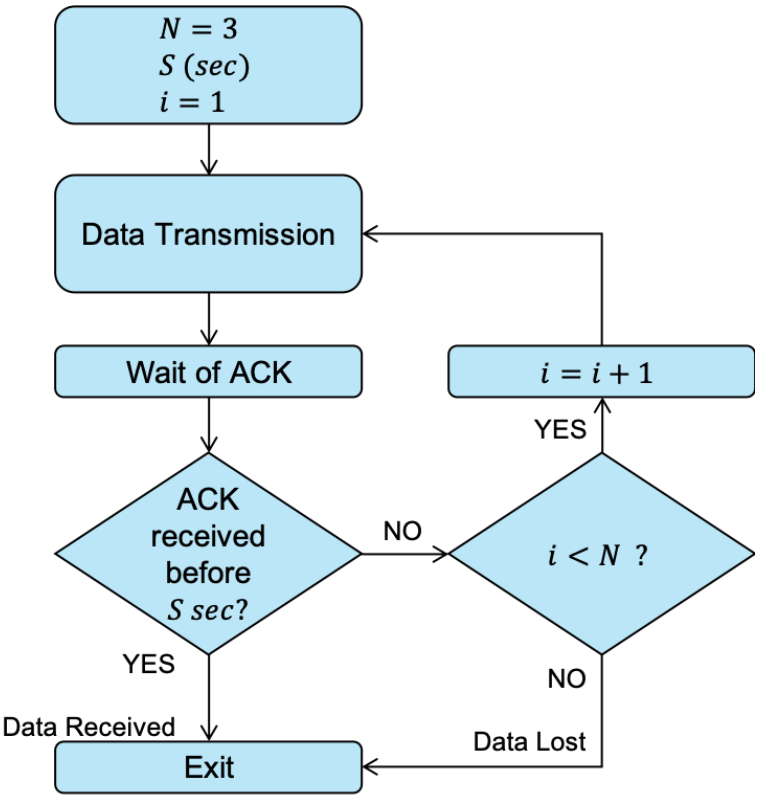


Figure 3.33: Flow chart for node to node data transmission in CupCarbon.

Acknowledgments in Transmission

Figure 3.34 depicts the packet delivery realization from CupCarbon. In this case, we have considered two sensor nodes $S3$ and $S4$ in the urban environment and both nodes are within the radio range of each other. Up right part of the Figure 3.34 shows that node $S3$ have initialized the communication and transmits data to node $S4$ over the wireless medium. Transmission of data is marked by red arrow in direction from node $S3$ to node $S4$. bottom right part of the Figure 3.34 shows that node $S4$ has successfully received data from node $S3$ and send him an ACK message which is marked by black arrow in the opposite direction. According to the re-transmission algorithm that is also shown in Figure 3.33. This transmission over the wireless channel can be effected by impulsive noise and can face four different cases as shown in left part of the Figure 3.34, which are explained below:

- Case 1: node $S4$ will successfully receives data in first transmission by node $S3$ and will send an ACK message back to transmitter node.
- Case 2: node $S4$ will receive the data not at first but in second re-transmission by node $S3$ and will send an ACK message back to transmitter node.
- Case 3: node $S4$ will not receive the data in first two tranmissions but in third re-transmission by node $S3$ and will send an ACK message back to transmitter node.
- Case 4: if $S3$ will not receive ACK message from node $S4$ within $S\ sec$, then after three

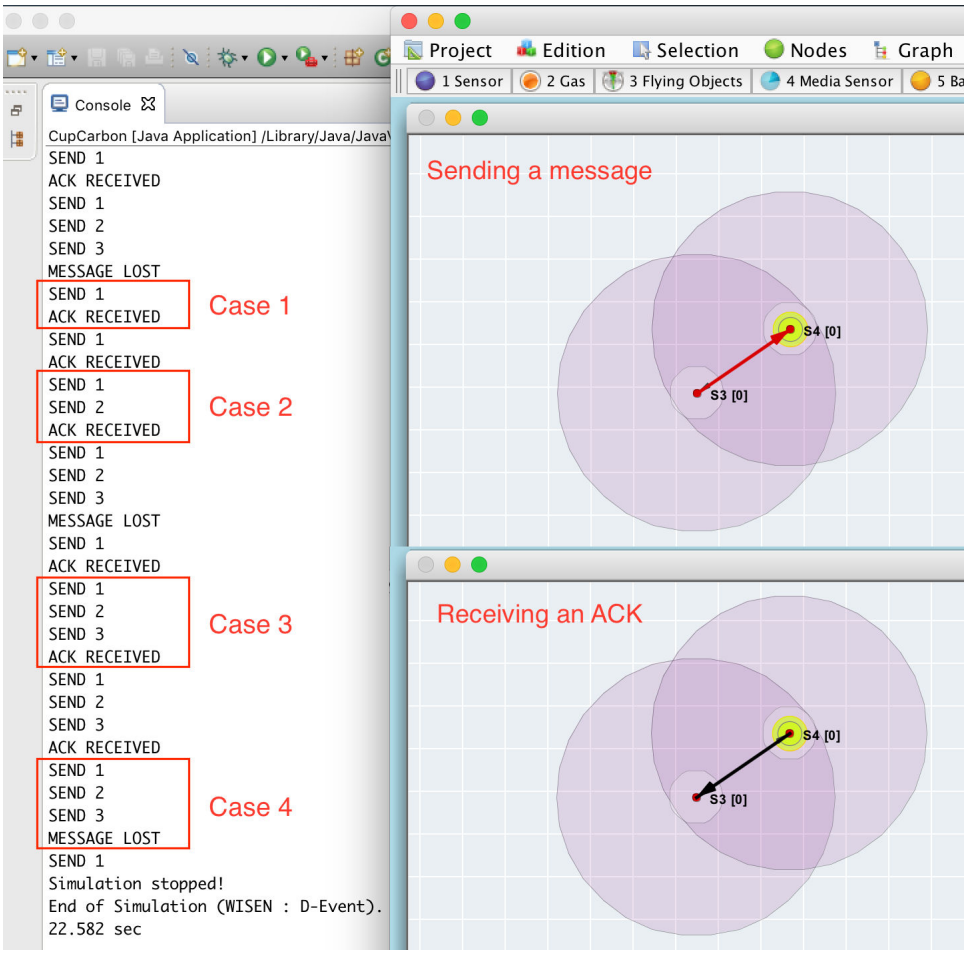


Figure 3.34: Data transmission and Acknowledgment message visualization in CupCarbon.

re-transmissions of data it will declare node $S4$ as a dead node and will not make any communication link in future.

SenScript

SenScript is the script used to program sensor nodes of the CupCarbon simulator. It is a script where variables are not declared but they can be initialized (set command). For string variables, it is not necessary to use the quotes. A variable is used by its name and its value is determined by $. It is possible to use the instruction function to add complex and additional functions programmed in Java (in a source code mode only). An example of "send command" is explained below:

- send hello 2: Sends hello to the sensor having an $id = 2$
- send $p 2: Sends the value of p to the sensor having an $id = 2$
- send $p: Sends the value of p in a broadcast mode

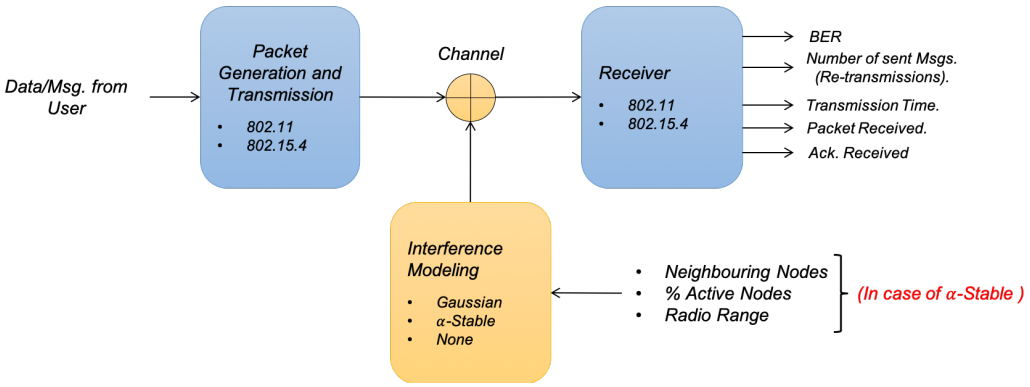


Figure 3.35: Schematic view of PHY's and interfrence model in CupCarbon.

- send $p * : Sends the value of p in a broadcast mode(the same as sens $p)
- send $p * 3: Sends the value of p in broadcast except the sensor having an $id = 3$
- send $p 0 4: Sends the value of p to sensors having a MY address equal to 4

5.2 Integration of PHY in CupCarbon

After adding the MAC the next step is to integrate PHY layer. PHY layers of the IEEE 802.11 and 802.15.4 are added as explained in the Section 2.3 and 3.2. Data from the MAC layer is provided from the user (from SenScript), then a communication protocol is assigned (whether 802.11 or 802.15.4). After applying the transmission functions, the system calculates the the parameters of the interference model (as explained in the Chapter 1). As mentioned in the second chapter, most of the WSNs simulators either does not use interference modeling or only use simplified Gaussian model. This blind use of the simple models, gives false estimations with an unacceptable margin of error for the received power levels. Thus, the idea here is to use the more realistic interference model. Two interference models are used in the CupCarbon: the Gaussian model and the α-stable model. As Gaussian is the special case of the α-stable class $(\alpha = 2)$, so the Gaussian model is not mentioned separately. Figure 3.35 shows the block diagram of the whole procedure. After calculating the α-stable parameters, the interference model is applied to the transmitted signal. Then, at the receiver the received signal is demodulated.

Another point which is worth mentioning here is that to avoid the complexity and to reduce the simulation time, the signal processing is done in the baseband.

Figure 3.36 shows the CupCarbon environment for the assignment of the communication protocol. It shows four sensor nodes placed arbitrarily in the city of Brest - France. The map is taken from OpenStreetMap. Each node is assigned the ZigBee protocol. The circle around each node shows its radio range. Nodes should be in the radio range of each other to communicate. Also on the main window shows the total simulation time, number of sent packets, number of received packets, number of acknowledgment packets, and number of lost packets in the network (or in the simulation of the selected nodes). Other parametric description is given below:

- **Standard:** The standard of the radio module (802.15.4, Wi-Fi, or LoRa) to be assigned to the node.

- **Radio name:** The name of the radio module to add to the sensor node. By default, one radio module ZigBee (802.15.4) is added automatically for each sensor node with the name 'radio1'.
- **Add:** To add new radio module.
- **Remove:** To remove the selected radio module.
- **Current:** To determine the current radio module. This part is used by the SenScript during simulation.
- **Radio Module List:** List of the added radio modules.
- **Network ID:** Network ID of the selected radio module.
- **MY:** Address of the selected radio module.
- **CH:** Channel of the selected radio module.
- **Radius:** Radius of the radio range of the selected radio module.
- E_Tx : Transmitted power.
- E_Rx : Received Power.
- **Sleeping Energy:** Energy consumption during sleep mode (not available in the current version).
- **Listening Energy:** Energy consupmtion during listening (not available in the current version).
- **Data Rate:** Data rates (default: 250 kbits/s).
- **Spreading Factor:** For LoRa radio modules only (explained in next Chapter).

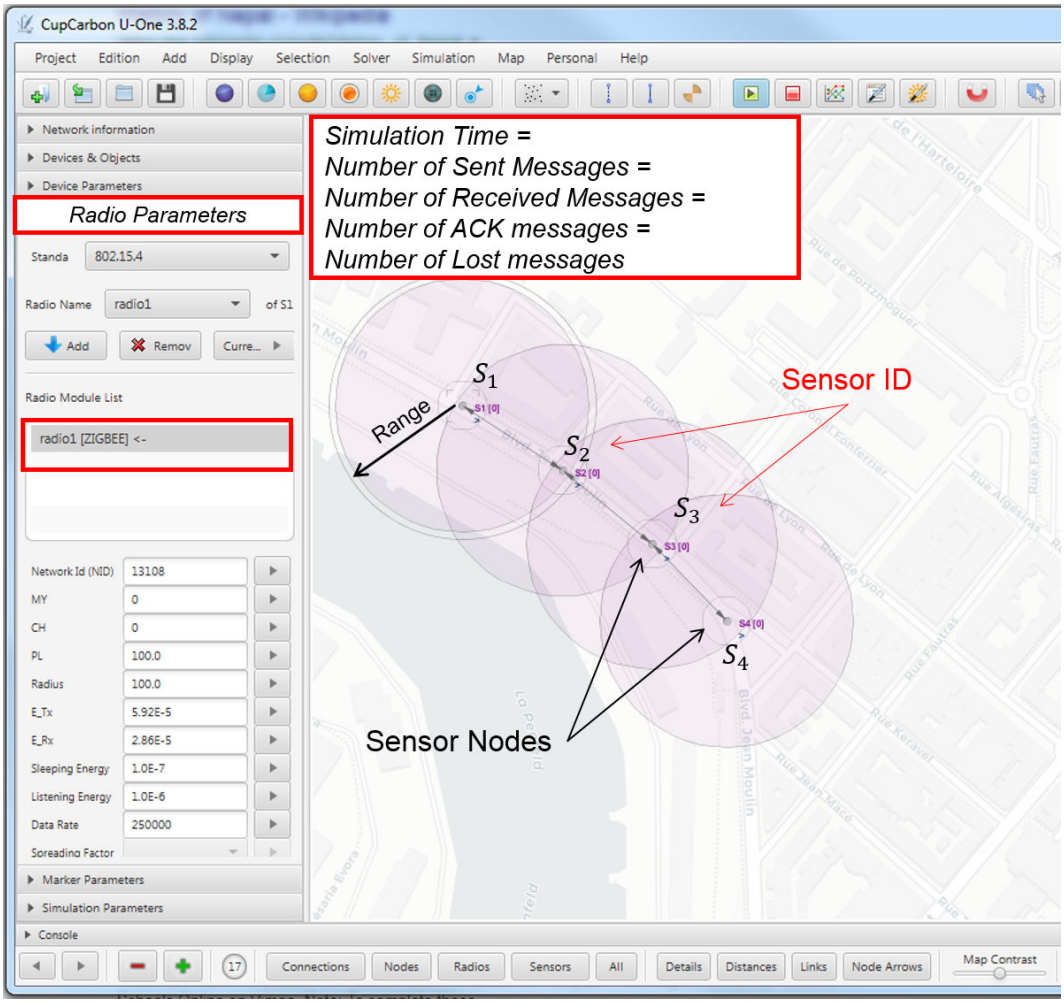


Figure 3.36: Radio parameter settings in CupCarbon.

4

LPWAN Protocol for CupCarbon

In order to address diverse requirements of IoT's applications, LPWAN offers novel communication paradigm. LPWAN technologies (SigFox, LoRa, Weightless-W etc.) successfully propose wide area connectivity from a few to tens of kilometers for low data rates, low-power and low throughput applications [3]. LPWAN are low-power version of cellular networks. LPWAN are supposed to ensure communication with low cost and long lifetime (typically up to ten years) nodes, and the distance between a node and base station may exceed $10\ km$. Amongst the recently proposed and prominent LPWAN technologies, the semiconductor manufacturer Semtech has introduced extensive utilization of advanced spread spectrum technology with their long range $LoRa^{TM}$ product line. $LoRa^{TM}$ is a spread spectrum modulation technique densely deployed for low-power wide area networks (LPWAN). The fact that in a wide area network, all the end-devices will communicate to the fewer gateway nodes using pure ALOHA medium access protocol, can cause interference. For this, LoRa provides different options in the usage of spreading factors (SF), Code rate (CR) and bandwidth (BW). Yet, there are some limitations regarding the network size that needs to be explained.

In a LoRa system, it is considered that when two or more end-devices transmit their data simultaneously using the same spreading factor (SF), and over the same channel, a collision will occur at the receiver. This collision will result in loss of all colliding frames. However, by utilizing capture effect, the receiver can successfully decode the frame even in the presence of a collision. Capture effect is the receiver's ability to correctly receive the frame from a collision. LoRa systems can also benefit from the capture effect, which would enhance the system's throughput performance. And this throughput can further be enhanced by introducing successive interference cancellation (SIC). SIC is a PHY ability to allow the receiver to decode simultaneous transmissions. This technique is well studied in communication literature but is not yet applied in LoRa. This motivates us to explore more options for LoRa to enhance its throughput.

This chapter explores the performance and the capacity limitations of LoRaWAN and LoRa networks. First, we measure the network throughput by calculating the probability of successful transmission with varied network sizes. Then we incorporate the capture effect in LoRa PHY. Capture effect enables a receiver to correctly receive a packet even if it collides with another transmission. Our results show that capture effect has a significant impact on the probability of successful reception. That proves that collision does not always lead to loss of both frames. In some cases, even both of colliding packets can be decoded successfully with SIC. Moreover, as LoRa PHY is proprietary and not clearly available, we provide a comprehensive analysis on technical features of PHY layer associated with LoRa. This chapter contributes the following:

- Throughput and nodes density limitations of LoRa network are explored by calculating the packet reception rate (PRR) with varied network sizes.

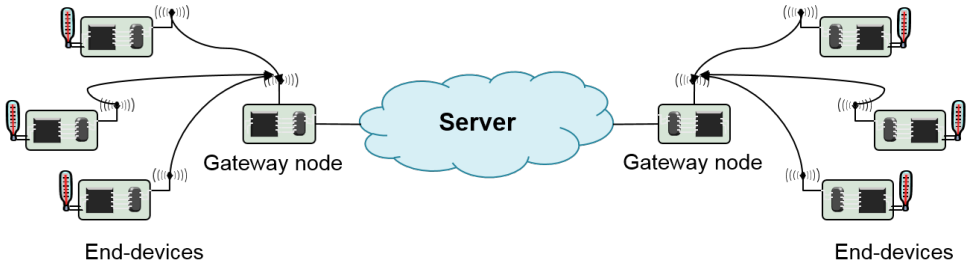


Figure 4.1: $LoRaWAN^{TM}$ network structure.

- Moreover, as LoRa PHY working is not much openly available, this chapter provides comprehensive analysis on technical features of PHY layer associated with LoRa.
- The benefits of capture effects are studied and further possible impact on the network throughput are proposed.
- SIC technique has studied and applied in LoRa to further improves its throughput.
- Integration of LoRa and LoRaWAN in the CupCarbon simulator.

1 LoRaWAN

Long range wide area network $LoRaWAN^{TM}$ is an open standard which defines MAC protocol envisioned for LoRa. LoRaWAN specifications provide the ability to control spreading factor SF and bandwidth BW utilization, (SF, CR and BW will be explained in the next section with LoRa PHY). This also allows nodes to transmit with the possible highest rate in bi-directional transmission.

1.1 Network Structure

LoRa network typically based on end-devices (sensors, actuators or both), gateways and a network server. End-devices in a LoRaWAN network follow the star of stars topology and connect to one or several gateways through LoRa wireless link. The gateways connect to the network server through an IP-based network and provide the bridging from end-devices to the IP world [102], as represented in Figure 4.1. Orthogonal nature of spreading factors enables simultaneous and collision free communication in the network. LoRaWAN defines three categories of end-devices, which are explained below [102].

- Class A: in this class, the communication is initialized by the end user, and they are allowed for bi-directional communications. End-devices can schedule an uplink transmission slot based on ALOHA protocol. Each uplink transmission is followed by two downlink receive windows. This class is considered more appropriate for energy constrained devices.
- Class B: in addition to random receive time windows, Class B devices also use scheduling methods. They receive a beacon from the gateway for the time synchronization.
- Class C: unlike class A and B, Class C end-devices have almost continuously open receive time windows. These windows only close while transmitting. This class is more energy consuming as compared with other classes.

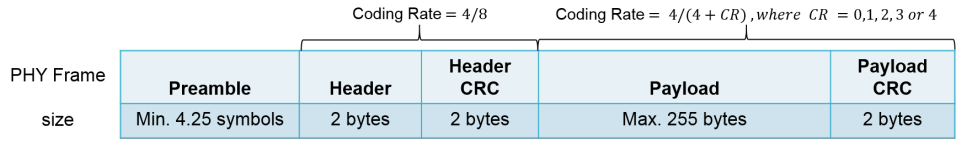


Figure 4.2: $LoRa^{TM}$ PHY frame format

1.2 Packet Structure

Figure 4.2 shows packet structure used by LoRa. LoRa offers maximum packet size of 256 bytes. This frame structure is also used in our simulations. The LoRa packet is composed as follows [103]:

- Preamble field : is used for the synchronization purposes. The receiver synchronized with the incoming data flow. The preamble field contains a sequence of constant up-chirps, two down-chirps and a quarter of up-chirp for synchronization purposes.
- Header field : depends on the choice of two available operation modes. In default explicit operational mode, the number of bytes in the header field specifies payload length, forward error correction (FEC) code rate (CR) value and presence of CRC at the end of the frame. The header also includes its cyclic redundancy check (CRC) field to protect the integrity of header. The second implicit operational mode specifies that coding rate and payload in a packet are fixed. In this mode, packet doesn't contain header field, which gives reduction in transmission time. If present, the header itself uses $CR = 4$.
- Payload field : contains actual data of 2 to 255 bytes.
- CRC : comprises cyclic redundancy check (CRC) for error protection of payload (2 bytes).

1.3 Time on Air

For LoRa packet, the actual time on the air can be defined as:

$$T_{packet} = T_{preamble} + T_{payload} \tag{4.1}$$

where $T_{preamble}$ is preamble duration and $T_{payload}$ is payload duration. For LoRa $T_{preamble}$ is defined as:

$$T_{preamble} = (n_{preamble} + 4.25)T_s \tag{4.2}$$

where constant value 4.25 represents the minimum preamble length of LoRa frame. $n_{preamble}$ is the preamble length and T_s denotes time of one symbol which is inversely proportional to the symbol rate:

$$T_s = 1/R_s \tag{4.3}$$

Let SF be the spreading factor and BW be the transmission bandwidth. Symbol rate R_s can be defined as:

$$R_s = BW/2^{SF} \quad SF \in \{7, 8, ..., 12\} \text{ and } \quad BW \in \{150, 250, 500\}kHz \tag{4.4}$$

The payload time is:

$$T_{payload} = PLSymb \times T_s \tag{4.5}$$

where $PLSymb = 8 + max(ceil(\frac{8PL+16CRC+(28-4SF)-20H}{SF-2DE})\frac{(CR+4)}{4}, 0)$

with the following explanations:

- PL : number of payload bytes.
- CRC : 0 when the CRC is excluded and 1 when 16-bit CRC is included.
- H : 0 when the header is enabled and 1 when there is no header.
- DE : 1 for enabled low data rate optimization and 0 for vice versa.
- CR : code rate.

So total time on the air taken by LoRa devices can be defined by using (4.1), (4.2) and (4.5).

$$T_{packet} = T_s(n_{preamble} + PLSymb + 4.25) \tag{4.6}$$

From the above calculations, one can say that SF, CR and BW have direct influence on the time on air of the LoRa packet. Time on air increases with the increase in payload size and SF values, also time on air varies with the change in CR value. Although, higher bandwidth means lower the time on air. This is shown in Figure 4.3.

1.4 Network Throughput

ALOHA based system allows the transmitters to transmit whenever there is a frame to send without carrier sensing. LoRaWAN only employs pure ALOHA based MAC protocol for medium access. The vulnerable time in pure ALOHA based network is twice the frame time, e.g $(2 \times T_{air})$ in LoRa. That means, a packet will be destroyed by any overlapping transmission starting in the time window that starts one packet time before the transmission of the packet and closes at the end of the transmission of the packet. Hence the throughput of such network is [104]:

$$\eta = Ge^{-2G} \tag{4.7}$$

$'2'$ in the superscript of exponential is because the vulnerable time is twice the frame time T_{air}. G represents average number of transmission attempts during frame time.

If we consider LoRa network consists of class A devices, correspond to pure ALOHA and transmission attempts occur according to the poisson process with average rate G attempts per slot. If we assume equal length packets and only nodes with same SF will collide, then G can be defined as the average number of attempts per time frame:

$$G = N \times p_i \times \lambda_i \times T_{air} \tag{4.8}$$

where T_{air_i} is the time on air of one LoRa frame (stated in (4.1)). λ_i is the packet generation rate of all N end-devices with SF_i in a network. As LoRa employs six different SF values, and claims that concurrent transmissions with different SF values shall not collide so, p_i is the probability that an end-device is using SF_i. Due to regional restrictions on operation in licensed free ISM bands,

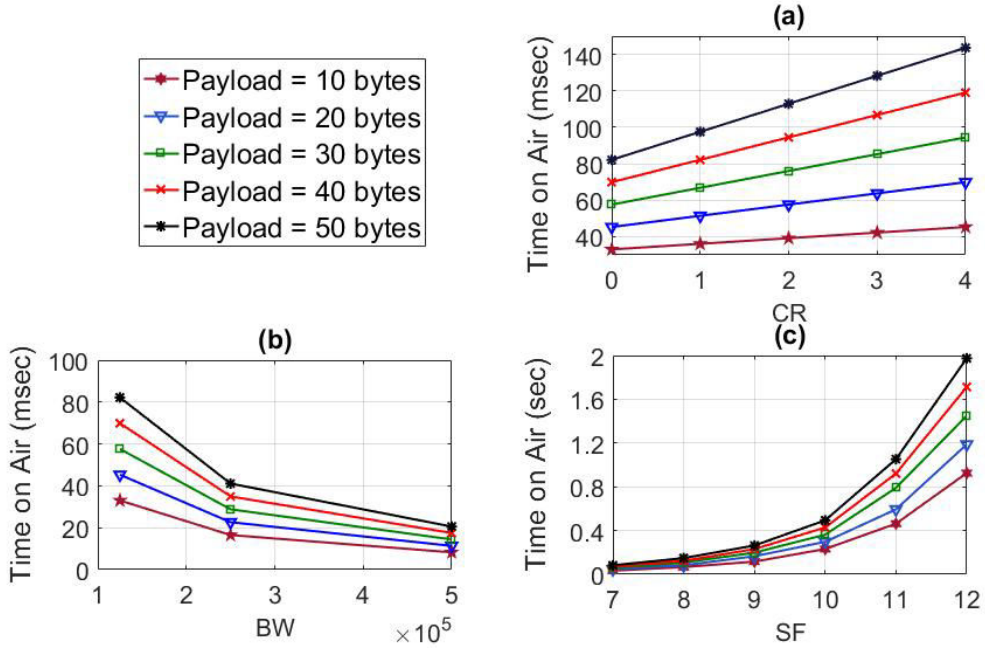


Figure 4.3: $LoRa^{TM}$ packet time on air (a) $SF = 7\ and\ BW = 125\ kHz$ (b) $SF = 7\ and\ CR = 0$ (c) $CR = 0\ and\ BW = 125\ kHz$

frame generation rate λ_i depends on the duty cycle D and time on air. If we assume that deployed network locates in Europe, then according to European Telecommunications Standards Institute (ETSI) 1% duty cycle applied on the usage of each sub-band [105].

$$\lambda_i = \frac{D}{T_{air_i}} \quad \text{or} \quad \lambda_i = \frac{1}{T_{off_i} + T_{air_i}} \tag{4.9}$$

Because $(T_{off_i} = (T_{air_i}/D) - T_{air_i})$ is the minimum period during which a device cannot access the medium due to the duty cycle restriction. Then (4.7) becomes:

$$\eta_i = p_i \lambda_i N T_{air_i} e^{(-2p_i \lambda_i N T_{air_i})}, \qquad i \in SF \tag{4.10}$$

The term e^{-2G} is called probability of successful transmission P of a frame and $0 \leq P \leq 1$. For LoRa based network it can be defined as:

$$P_i = e^{(-2p_i \lambda_i N T_{air_i})}, \qquad i \in SF \tag{4.11}$$

The outage probability will be:

$$P_{out_i} = 1 - P_i, \qquad i \in SF \tag{4.12}$$

LoRa claims that orthogonal nature of spreading factors enables simultaneous and collision free communication in the network. However, [106] shows that these spreading factors are not perfectly orthogonal. Nevertheless, we are going to consider that different spreading factors do not interfere and we are interested in the limit in successful transmissions in a LoRaWAN system. Figure 4.4

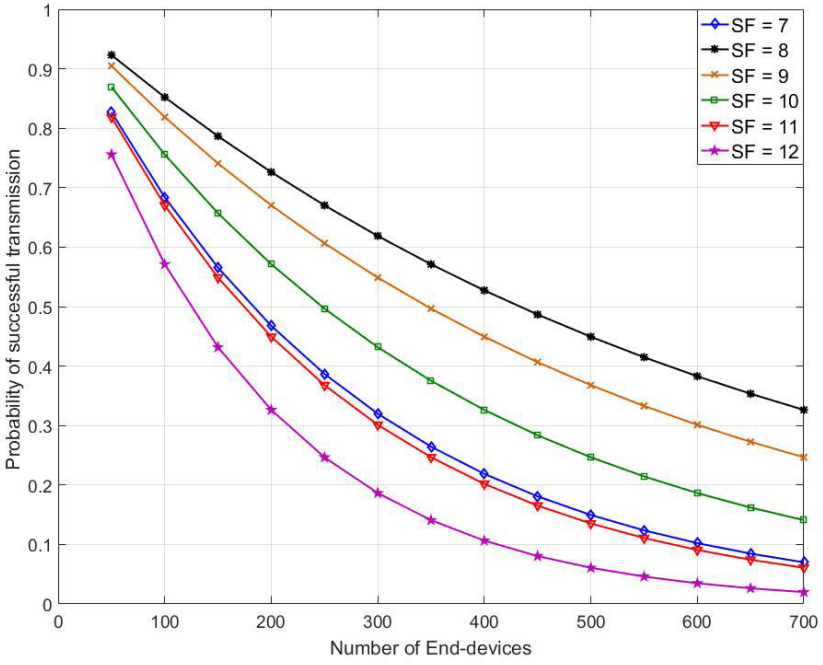


Figure 4.4: Probability of successful transmission

shows the probability of successful transmission depending on the number of users and SF. With the increase in the network size, the probability of successful reception drops due to increase in collisions in the network.

2 LoRa Technology

LoRa utilizes wide band usually of $125\ kHz$ or more to broadcast a signal. Using bands that are not too narrow, allows LoRa to exhibit some robustness against some characteristics of the channel (frequency selectivity, doppler effect). The transmitter generates chirp signals by varying their frequency over time and keeping phase between adjacent symbols constant. Receiver can decode even a severely attenuated signal 19.5 dBs below the noise level [103]. Main characteristics of LoRa modulation depends on SF, CR, and BW. Spreading factor ($SF = log_2(R_c/R_s)$) is the ratio between symbol rate (R_s) and chip rate (R_c). LoRa employs six orthogonal spreading factors (7 to 12). SF provides a trade-off between data rate and range. Along with the spreading factors, forward error correction (FEC) techniques are used in LoRa to increase the receiver sensitivity further. Code rate CR defines the amount of FEC in LoRa frame. LoRa offers $CR = 0,\ 1,\ 2,\ 3$ and 4, where $CR = 0$ means no encoding. The choice of a higher SF and CR values also influence the T_{air}, and an increase in T_{air} will increase the off period T_{off} duration as well. Although, choice of higher BW will reduce the T_{air}, but lowers the receiver sensitivity. LoRa provides three scalable BW settings of $125kHz, 250kHz$ and $500kHz$. Taking these parameters into account, the useful bit

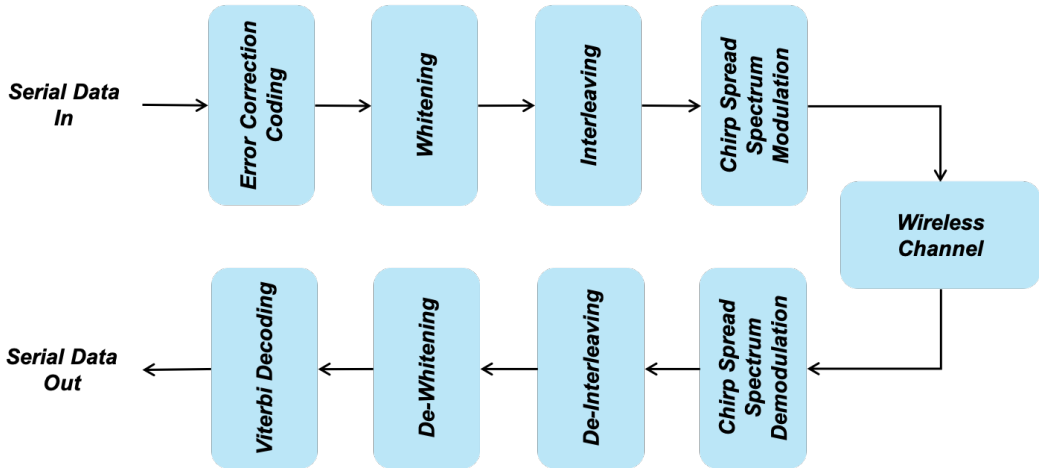


Figure 4.5: $LoRa^{TM}$ PHY layer architecture

Tableau 4.1: Error detection and correction capabilities of $LoRa^{TM}$.

Coding Rates	Error Detection (bits)	Error Correction (bits)
$4/5$	0	0
$4/6$	1	0
$4/7$	2	1
$4/8$	3	1

rate R_b equals:

$$R_b = \frac{4 \times SF \times BW}{(4+CR) \times 2^{SF}} \qquad (bits/sec) \tag{4.13}$$

2.1 LoRa PHY Structure

The LoRa is a Semtech proprietary technology and is not fully open. This section gives the analysis on the working of PHY in LoRa, according to our understanding. Figure 4.5 shows the block diagram of LoRa transceiver, which is briefly explained below.

Encoding

First, the binary source input bits passes through an encoder. The output of encoder depends on the choice of CR value. Encoding reduces the PER in the presence of short bursts of interference. LoRa uses Hamming codes for FEC. These are linear block codes and are easy to implement. LoRa uses coding rates ($coding\ rate = 4/(CR+4)$) of $4/5$, $2/3$, $4/7$ and $1/2$. Which means, if the code rate is denoted as k/n, where k represents the number of useful information bits, and encoder generates n output bits, then $(n-k)$ are the redundant bits. If we assume CR values between 1, 2, 3 and 4 for coding rates $4/5$, $4/6$, $4/7$ and $4/8$ respectively, then the error detection and correction capabilities are as shown in table 4.1.

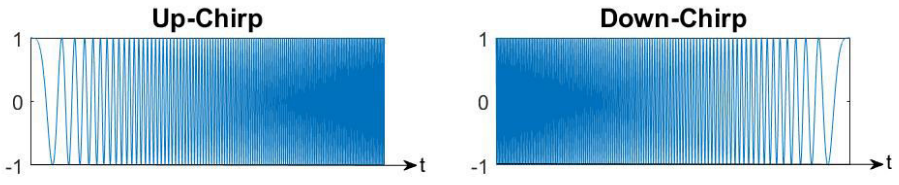


Figure 4.6: Up-chirp and down-chirp signal.

Whitening

The output of the encoder passes through the whitening block. Whitening is an optional step in LoRa, which can be implemented by Manchester encoding to induce randomness. Here, the purpose of whitening is to make sure that there are no long chains of $0's$ or $1's$ in the payload.

Interleaving

The output of whitening block passed to the interleaving block. Interleaver uses diagonal placing method to scramble each $4 + CR$ codeword, and sends it to spreading block.

Chirp Spread Spectrum Modulation

This block here spreads each symbol over an up-chirp according the SF value used. For example, for $SF = 7$ and 12, 128 and 4096 $chips/symbol$ will be used, consecutively. The relationship between the symbol rate $R_s = BW/2^{SF}$ and chip rate R_c is:

$$R_c = 2^{SF} \times R_s, \quad \text{and} \quad R_c = \frac{2^{SF} \times BW}{2^{SF}}, \quad \text{so,} \quad R_c = BW \ chips/sec$$

It takes much larger BW for transmission than required for the considered data rate. Chirp signal is a sinusoidal signal with either linearly increasing or decreasing frequency. A linear chirp waveform can be expressed as:

$$c(t) = \begin{cases} exp(2\pi j(at+b)t), & \frac{-T_s}{2} \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.14}$$

with $at + b = f_{min} + \dfrac{f_{max} - f_{min}}{T_s} t$

where f_{max} and f_{min} are the maximum ($125\ kHz$ in our case) and minimum frequency, respectively. T_s is the symbol duration. An up-chirp and a down-chirp are shown in Figure 4.6.

Each symbol of SF bits can be represented by shifting the frequency ramp based on the symbol value. So, each coded chirp is obtained by a cyclic shift in an up-chirp, as illustrated in Figure 4.7. Circular shift in raw up-chirp is expressed in (4.15).

$$c(t) = \begin{cases} exp(2\pi j(a(T_s + t - \Delta t) + b)(Ts + t - \Delta t)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t \\ exp(2\pi j(a(t - \Delta t) + b)(t - \Delta t)), & \frac{-T_s}{2} + \Delta t \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.15}$$

where Δt is the shift in time that depends on the symbol value.

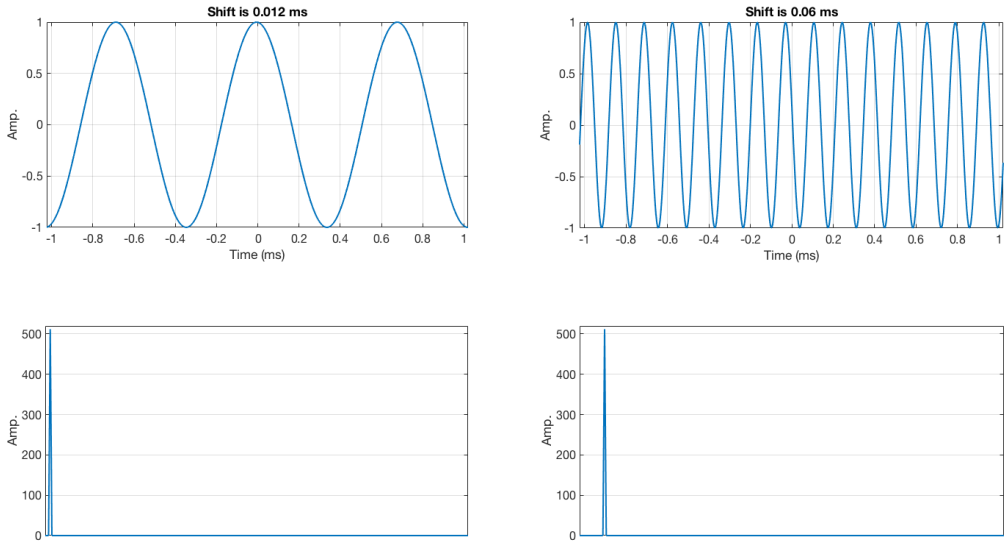


Figure 4.7: Matched Filter Output

CSS Demodulation

The matched receiver for a linear chirp is performed by multiplication with the down-chirp, as shown in Figure 4.7 and can be represent mathematically for $(\Delta t = 0)$ (note that, in case of symbol value $'0'$, the transmitter will send an up-chirp):

$$y(t) = exp(2\pi j(at + b)t) \ \times \ exp(-2\pi j(a(T_s - t) + b)(T_s - t)). \tag{4.16}$$

$$y(t) = exp(2\pi j(at^2 + bt - a{T_s}^2 + 2aT_s - at^2 - bT_s + bt)) \tag{4.17}$$

$$y(t) = exp(2\pi j(2atT_s + 2bt - a{T_s}^2 - bT_s)) \tag{4.18}$$

$$y(t) = exp(2\pi j(2t(aT_s + b) - a{T_s}^2 - bT_s)) \tag{4.19}$$

The matched receiver output for any other symbol would be (when $\Delta t \neq 0$) for $\frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t$:

$$y_i(t) = exp(2\pi j(a(t + T_s - \Delta t) + b)(t + T_s - \Delta t)) \ \times \ exp(-2\pi j(a(T_s - t) + b)(T_s - t)). \tag{4.20}$$

$$y_i(t) = exp(2\pi j(2t(b - a\Delta t - 2aT_s) + a\Delta t^2 - 2a\Delta tT_s - b\Delta t)) \tag{4.21}$$

And for $\frac{-T_s}{2} + \Delta t \leq t \leq \frac{T_s}{2}$

$$y_i(t) = exp(2\pi j(a(t - \Delta t) + b)(t - \Delta t)) \ \times \ exp(-2\pi j(a(T_s - t) + b)(T_s - t)). \tag{4.22}$$

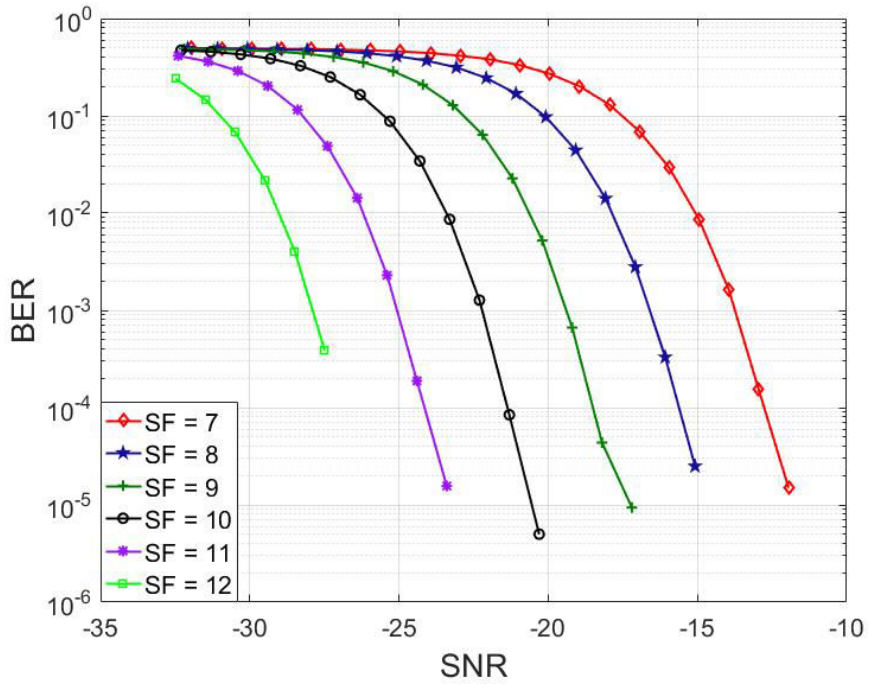


Figure 4.8: BER performance of PHY layer based on LoRa with different SF values and $CR = 0$.

$$y_i(t) = exp(2\pi j(2t(b - a\Delta t + aT_s) + a\Delta t^2 - b\Delta t - aT_s^2 - bT_s)) \tag{4.23}$$

Hence,

$$y_i(t) = \begin{cases} exp(2\pi j(2t(b - a\Delta t - 2aT_s) + a\Delta t^2 - 2a\Delta tT_s - b\Delta t)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t \\ exp(2\pi j(2t(b - a\Delta t + aT_s) + a\Delta t^2 - aT_s^2 - bT_s - b\Delta t)), & \frac{-T_s}{2} + \Delta t \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.24}$$

Then, the output signal is analyzed to identify the presence of the sharp narrow peak in frequency domain, which is at $'3'$ and $'15'$ on Figure 4.7. The sharp narrow peak occurs at the time index corresponding to the constant value of the coded chirp.

2.2 PHY Performance

Figure 4.8 shows the BER performance of LoRa based PHY, plotted over the signal-to-noise ratio (SNR). Vertical axis show different bit error rate values and horizontal axis show SNR values. Figure 4.8 shows the impact of spreading factor, by increasing the spreading factor, BER decreases noticeably. But the reduced data rate and increased time on air are the overheads. These results have been produced with $CR = 0$ and $BW = 125kHz$. Table 4.2 shows the date rates associated with the spreading factor SF, and the coding rate CR. It can be easily noticed that CR also effect the data rate. But increasing SF and CR helps combat the worst wireless conditions.

Tableau 4.2: Data Rates offered in $LoRa^{TM}$.

SF	**Data Rate** $(CR = 0)$	CR	**Data Rate** $(SF = 7)$
7	6.8 $kbps$	0	6.8 $kbps$
8	3.9 $kbps$	1	5.4 $kbps$
9	2.1 $kbps$	2	4.5 $kbps$
10	1 $kbps$	3	3.9 bps
11	671 bps	4	3.4 $kbps$
12	366 bps	–	–

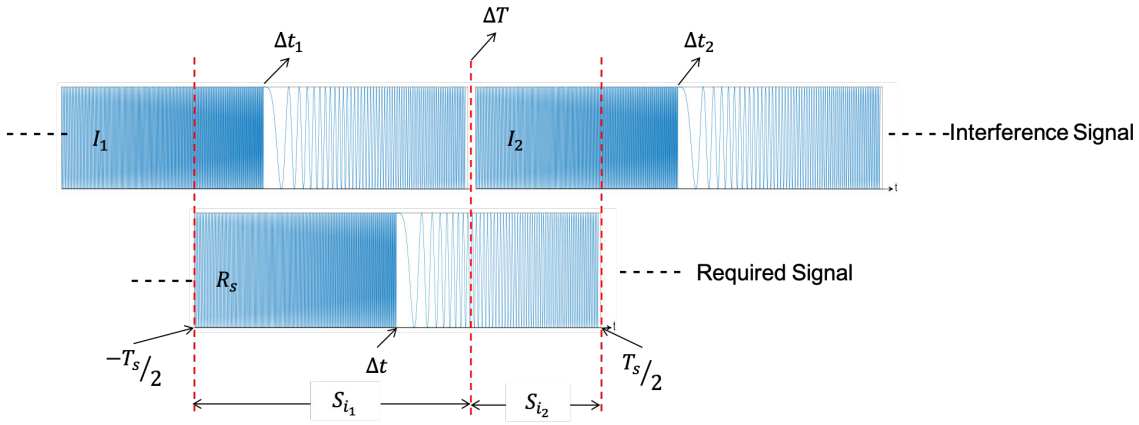


Figure 4.9: Interfering LoRa symbols at receiver.

3 Interference in LoRa

As mentioned before, LoRaWAN class A devices use pure ALOHA MAC protocol for medium access. One of the limitations of ALOHA based systems is its blind transmission strategy, which causes inefficient use of spectrum. Simultaneous transmission causes packet loss. The maximum throughput of pure ALOHA system can be defined by taking the derivative of (4.7) w.r.t average traffic G and setting it equal to zero $\frac{d}{dG}(Ge^{-2G}) = 0$, gives $G = 1/2$. Substituting this value into (4.7) gives:

$$\eta = 1/2e^{-1} = 0.1839 \tag{4.25}$$

So, pure ALOHA based network can give 18.39% of maximum efficiency. For LoRa modulation, this efficiency can be increased by improving the receiver technique. Let's assume, the collsion occurs between two consective LoRa symbols as shown in the Figure 4.9. The I_1 and I_2 are the symbols from the interference signal, consectively. The R_s is the symbol of the ongoing transmission. The

experssion for $\frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta T$ can be written as:

$$S_{i_1}(t) = S(t) + \begin{cases} exp(2\pi j(a(\Delta T - t + \Delta t_1) + b)(\Delta T - t + \Delta t_1)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t_1 \\ exp(2\pi j(a(\Delta T - t + \Delta t_1 - T_s) + b)(\Delta T - t + \Delta t_1 - T_s)), & \frac{-T_s}{2} + \Delta t_1 \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.26}$$

$$S_{i_1}(t) = S(t) + \begin{cases} exp(2\pi j(t(at - 2a\Delta T - 2a\Delta t_1 - b) + a\Delta T^2 + a\Delta t_1^2 + b\Delta T + b\Delta t_1 + 2a\Delta T\Delta t_1)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t_1 \\ exp(2\pi j(t(at - 2a\Delta T - 2a\Delta t_1 + 2aT_s - b)aT_s^2 - 2aT_s\Delta T - 2aT_s\Delta t_1 + a\Delta T^2 + a\Delta t_1^2 + 2a\Delta T\Delta t_1 + b\Delta t_1 + b\Delta T - bT_s)), & \frac{-T_s}{2} + \Delta t_1 \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.27}$$

And the experssion for $\frac{-T_s}{2} + \Delta T \leq t \leq \frac{T_s}{2}$ can be written as:

$$S_{i_2}(t) = S(t) + \begin{cases} exp(2\pi j(a(t + T_s - \Delta T + \Delta t_2) + b)(t + T_s - \Delta T + \Delta t_2)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t_2 \\ exp(2\pi j(a(t - \Delta T + \Delta t_2) + b)(t - \Delta T + \Delta t_2)), & \frac{-T_s}{2} + \Delta t_2 \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.28}$$

$$S_{i_2}(t) = S(t) + \begin{cases} exp(2\pi j(t(at - 2a\Delta T + 2a\Delta t_2 + 2aT_s + b) + 2T_s^2 - 2aT_s\Delta T + 2aT_s\Delta t_2 + a\Delta T^2 + a\Delta t_2^2 - 2a\Delta T\Delta t_2 + bT_s - b\Delta T + b\Delta t_2)), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t_2 \\ exp(2\pi j(t(at - 2a\Delta T + 2a\Delta t_2 + b) + a\Delta T^2 - 2a\Delta T\Delta t_2 + a\Delta t_2^2 - b\Delta T + b\Delta t_2)), & \frac{-T_s}{2} + \Delta t_2 \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.29}$$

At this point, multiplication with down chirp $D(t)$ is performed, and the expression would be:

$$y_{i_1} = S_{i_1}(t) \times D(t)$$
$$y_{i_2} = S_{i_2}(t) \times D(t)$$

$$y_{i_1}(t) = y_i(t) + \begin{cases} exp(2\pi j(2at(T_s - \Delta T - \Delta t_1) + a(\Delta T^2 + \Delta t_1^2 - T_s^2 + 2\Delta T\Delta t_1) + b(\Delta T + \Delta t_1 + T_s))), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t_1 \\ exp(2\pi j(2at(2T_s - \Delta T - \Delta t_1) + a(2\Delta T\Delta t_1 - 2T_s\Delta T - 2s\Delta t_1 + \Delta T^2 + \Delta t_1^2) + b(\Delta T + \Delta t_1 - 2T_s))), & \frac{-T_s}{2} + \Delta t_1 \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.30}$$

$$y_{i_2}(t) = y_i(t) + \begin{cases} exp(2\pi j(2t(2aT_s - a\Delta T + a\Delta t_2 + b) + a(2T_s\Delta t_2 - 2T_s\Delta T - 2\Delta T\Delta t_2 + \Delta T^2 + \Delta t_2^2) + b(\Delta t_2 - \Delta T))), & \frac{-T_s}{2} \leq t \leq \frac{-T_s}{2} + \Delta t_2 \\ exp(2\pi j(2t(aT_s - a\Delta T + a\Delta t_2 + b) + a(\Delta T^2 + \Delta t_2^2 - T_s^2 - 2\Delta T\Delta t_2) + b(\Delta t_2 - \Delta T - T_s))), & \frac{-T_s}{2} + \Delta t_2 \leq t \leq \frac{T_s}{2} \\ 0, & \text{otherwise} \end{cases} \tag{4.31}$$

The frequency response of the (4.30) and (4.31) combine, will give sharpe narrow peaks at time Δt, Δt_1 and Δt_2, which coressponds to the constant value of the coded chirps, as shown in Figure 4.10. The receiver can make the decision by selecting the peak with highest amplitude value.

The receiver can successfully differentiate between the interferers based on their received power values. The problem occurs when one or more interferers carry either the same value ($\Delta t = \Delta t_1 = \Delta t_2$) so that their power adds up or have an equivalent or more power than the useful one. In Figure 4.11(a), first interferer has the same received power as the useful user and in Figure 4.11(b), all interferers contain same value ($\Delta t = \Delta t_1 = \Delta t_2$) which causes an increase of peak value in the frequency domain.

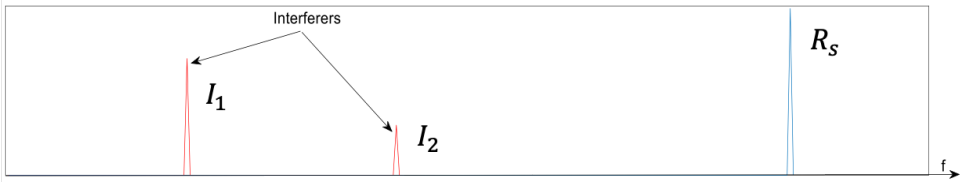


Figure 4.10: FFT of Coded chirps at the values 30, 100 and 128 consecutively, with $SF = 8$.

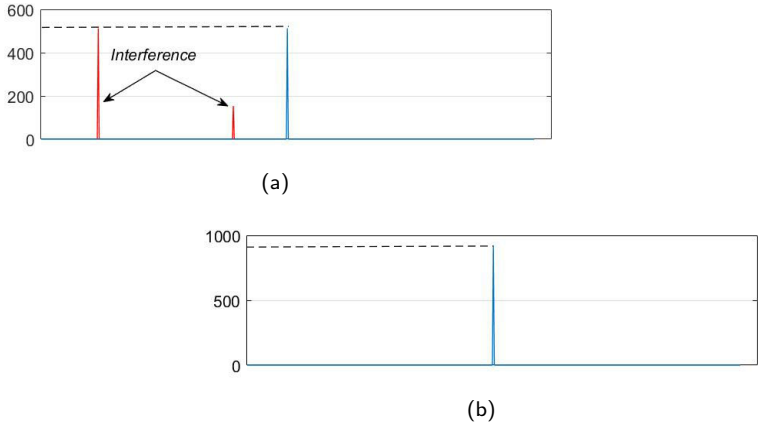


Figure 4.11: (a) FFT of Coded chirps at the values 30, 100 and 128 consecutively, with $SF = 8$. (b) FFT of 3 Coded chirps at the value 128, with $SF = 8$.

4 Capture Effect

LoRa PHY employs kind of frequency modulation that manifests capture effect. In the previous section, it is shown how the receiver might be able to perform successful reception of strong packet in the presence of collision. In the past, many theoretical studies on capture effect have been performed to increase the packet reception rate of a network, in the presence of a collision. But in context of LoRa, not much research has been done. Practical studies in [107] [108] and [109], utilized capture effect for loRa based system to decode the strongest signal. In [110], authors presented capture study on equal power collisions in the pure ALOHA based 802.15.4 system. In [111], authors state that there collision detection approach can differentiate between a packet collision and packet loss for 802.15.4 based system.

In general, the CE receiver keeps monitoring for the new potential preambles and if its SINR is above a given ratio, receiver stops ongoing reception and re-synchronizes with the new packet and demodulate the signal. We are going to characterize the capture effect in the next section (the probability for a packet to be decoded despite the presence of one interferer). Note that for the sake of simplicity, we have only considered 2-packet collision. This analysis can be extented to 3 or more packet collisions.

The capture characteristics of any radio transceiver depend on the modulation, decoding schemes and its hardware design and implementation. In a RF interference environment, a particular signal

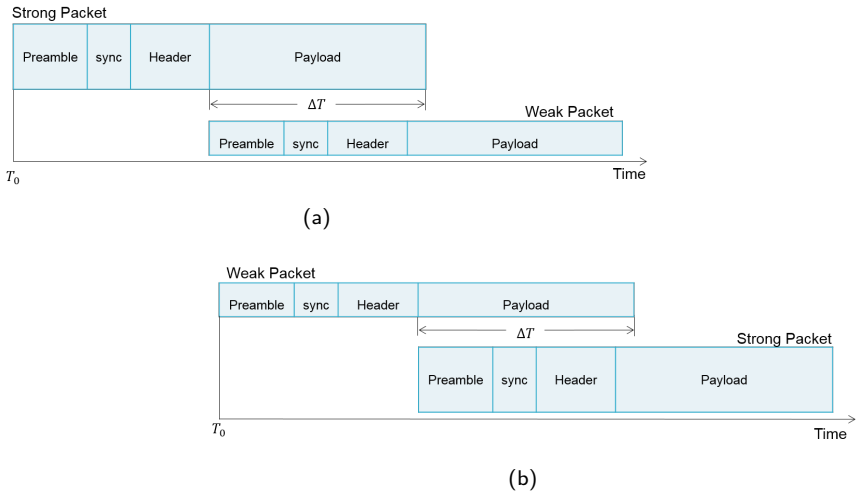


Figure 4.12: Capture Scenarios (a) Stronger first: A weak packet arrives during the reception of a strong packet. (b) Stronger last: the stronger packet arrived last.

X can be successfully decoded if:

$$SINR_X = \frac{P_X}{\sum P_I + \sigma^2} > Th \tag{4.32}$$

where P_X is the source signal strength, $\sum P_I$ is the aggregate interference strength from the other active users in the network, σ is the channel noise coefficient and Th is the minimum SINR threshold required to successfully decode signal X. When two or more packets collide, with CE it is still possible to receive one of them. CE enables the receiver to decode a packet that satisfies 4.32, even if it arrives during the reception of an ongoing packet.

For a LoRa modulation, as shown in Figure 4.11, only the strength of the strongest interferer will matter as long as the simultaneous number of interferers is not too high and the probability to have interferers with a shift that falls at the same time remains low. So (4.32) will become:

$$SIR_X = \frac{P_X}{P_I} > Th \tag{4.33}$$

4.1 Capture Effect Scenarios

In the literature [107, 108, 109, 110, 111], usually, two capture scenarios are taken into account. Both capture scenarios are shown in Figure 4.12.

- Decoding the First: During the reception of a packet, a second packet arrives and creates collision. In this case, receiver synchronizes with the first packet and tries to perform successful reception.
- Decoding the Last: Another scenario would be to decode the packet that arrives later. This necessitates to be able to detect the preamble of the second packet and then to correctly decode the packet.

Figure 4.12 illustrates these two cases. We generate random collisions at the receiver by generating two LoRa packets with time difference $(T_0 \leq \Delta T \leq T_{air})$. The first signal arrives at T_0 and

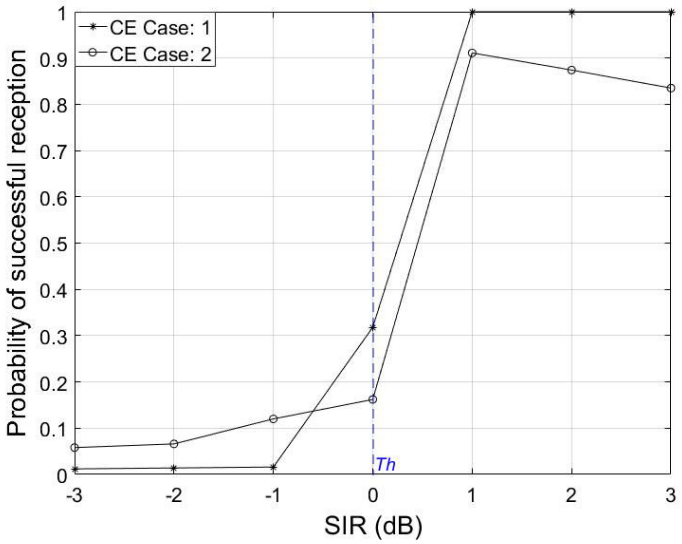


Figure 4.13: Capture Results with $SF = 8; CR = 0$ and $BW = 125kHz$

second signal arrives after a random duration, within the T_{air} of the first packet. The transmission of an interfering packet can start at any time, and overlapping length ΔT of both packets varies randomly. The goal here is to identify under which power settings the collision detection and successful reception will work. In both cases, packet reception rate (PRR) is measured at the receiver, in simple steps as follows:

- Preamble detection: if preamble detection is valid, then it passes for sync word detection.
- Sync word detection: after the receiver detects the preamble it searches for the sync word and finds the starting of the header.
- Validation of Header and Payload: if header and payload data is not corrupted then it is considered as successful frame reception.

The packet structure used in these simulations is shown in Figure 4.2. The preamble consists on four up-chirps, two down-chirps and a quarter of an up-chirp. However, increase in preamble duration can improve the detection probability. An explicit header is used with $2\ bytes$ CRC. The header is encoded with $CR = 4$. The payload is $20\ bytes$ long, with no channel encoding $(CR = 0)$ and $SF = 8$. Channel coding is used to improve the reliability of the communication system by adding redundancy in the transmit data.

Figure 4.13 presents the capture results. The probability of successful reception is calculated with 1000 packets transmissions for each power setting on random overlapping lengths ΔT. The x-axis shows signal to interference power (SIR) and y-axis shows the probability of successful reception with capture. Note that, we do not expect that the received power ordering is known a prior. From Figure 4.13 we can assume that if the received power difference between two interferers are around $1\ dB$, the receiver can successfully decode the strong packet. Thus, (4.33) can be express as,

$$SIR_X = \frac{P_X}{P_I} \geq 1\ dB \tag{4.34}$$

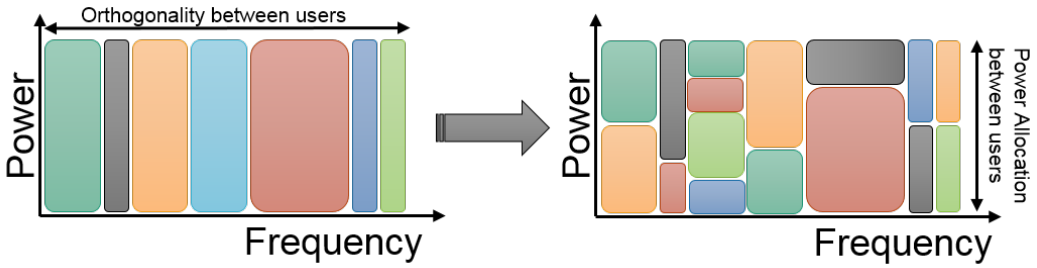


Figure 4.14: From OMA to NOMA using power domain multiplexing.

Successfully decoding one of the colliding packet can significantly increase the system throughput of any network. In the presence of CE, the total throughput of the system of N nodes can be expressed as:

$$\eta_{CE} = G \times [P(no\ collison) + P(collison) \sum_{i=1}^{G} P(SIR_i > Th)] \tag{4.35}$$

$P(no\ collison)$ and $P(collison)$ are probability of no collision and probability of collision at the receiver, respectively. Th is a threshold value set on signal to interferer i power ratio SIR_i of received signal.

In a pure ALOHA network, a node can successfully transmit a frame if no other node has a frame to transmit during two consecutive frame times (vulnerable time $2T_{air}$). The probability of a node having no frame to send is $(1-p)$. The probability that none among the rest of $N-1$ nodes have a frame to send will be $(1-p)^{N-1}$. The probability that none of the $N-1$ nodes have a frame to send during the vulnerable time is $(1-p)^{2(N-1)}$. Then the probability of being alone of a particular node will be: $P = p(1-p)^{2(N-1)}$.

5 Non Orthogonal Multiple Access (NOMA)

Non-orthogonal multiple access (NOMA) is a recently introduced multiple access technique proposed for 3GPP Long Term Evolution (LTE). The main feature of NOMA is to enable multiple users at the same time, frequency, or code, by utilizing different power levels for each user. Multiple users are multiplexed in the power domain which was not sufficiently explored in the previous mobile generations, as illustrated in Figure 4.14. This way it yields a significant spectral efficiency gain over conventional orthogonal multiple access. Take the example of conventional orthogonal frequency division multiple access (OFDMA) used by 3GPP-LTE. One of its main drawbacks is that its spectral efficiency is low when some bandwidth resources, such as sub-carriers, are allocated to users with poor channel conditions. On the contrary, NOMA allows each user to have access on all the sub-carriers. Hence the bandwidth resources allocated to the users with poor channel conditions can still be accessed by the users with strong channel conditions, which significantly improves the spectral efficiency.

In this section, we first give an introduction to NOMA, such as typical NOMA power allocation policies, and then a brief description on the use of successive interference cancellation (SIC).

5.1 Working Principle of NOMA

In this section we will provide the working details of NOMA for uplink and downlink. In general, NOMA allows the superposition of transmitted signals in a NOMA cluster. The desired signal is then detected and decoded at the receiver (nodes in the downlink and gateway in the uplink) by applying SIC.

Downlink NOMA

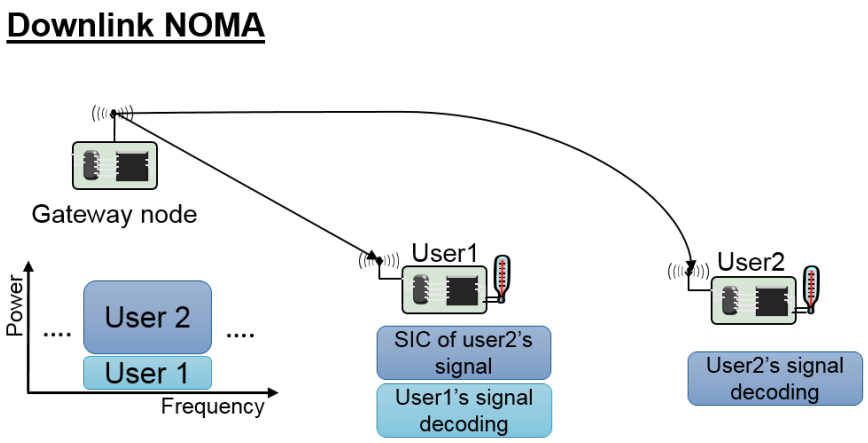
For simplicity, we have considered NOMA downlink transmission with only two transmitters and one gateway node. The gateway node will transmit a signal $X = \sum_{i=1}^{N} \sqrt{P_i} X_i$, where X_i is the transmitted signal with power P_i for i^{th} user among N users NOMA cluster. For downlink NOMA, high transmission power is allocated to the users with poor channel conditions and vice versa. The received signal at the i^{th} node will be $Y_i = H_i X + \sigma^2$. There are two methods to decode both signals at receiver, one is by using cooperative NOMA and non-cooperative NOMA. In non-cooperative NOMA transmission, each user decodes their corespondent signal normally. While in cooperative NOMA transmission the nodes with stronger channel conditions act as relays to help other nodes with weaker channel conditions. Cooperative NOMA transmission scheme is divided into direct transmission and cooperative transmission. In direct transmission phase, the gateway node will broadcasts a combined signal for all the nodes in the NOMA cluster. Cooperative transmission phase will carry out the SIC process at the node with better channel conditions and this node will be forward to the decoded signal to the other user. Therefore, two or more copies of the messages will receive at the other user. SIC gives the advantage that the messages to the users with weaker channel conditions have already been decoded by the users with stronger channel conditions. Hence, it is useful to utilize the nodes with stronger channel conditions as relays. As a result, the reception reliability of the nodes with weaker channel conditions can significantly improve.

Uplink NOMA

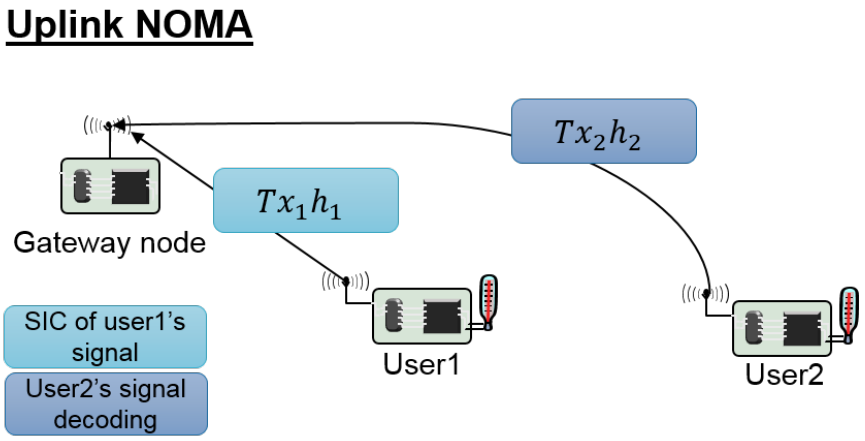
In uplink, each node transmits a signal x_i with a transmit power P_i such that the received signal at the gateway can be defined as $Y = H_i \sqrt{P_i} X_i + \sigma^2$. The power transmitted by each node is limited by the its maximum battery power. All nodes can independently use their battery powers. If the channel gains are too close, power control can be used to boost up the performance of the node with better channel gain, while maintaining the performance of the node with weaker channel gains at a certain level. Moreover, to apply SIC at the gateway node, it is important to maintain the separation of all signals that are superimposed within Y. Since the channels are different in uplink for each user, each signal will experiences distinct channel attenuation. As a result, the received signal power P_i corresponds to the strongest channel user is likely to be the strongest at the gateway. Therefore, this signal can be decoded first and will experience interference from all signals in the cluster with relatively weaker channel conditions. So, the transmission from the highest channel gain node will experience interference from all other active nodes within its cluster. And the transmission from the lowest channel gain node will receive no interference from other nodes in the same cluster. A two user example is demonstrated in the Figure 4.15.

5.2 Successice Interference Cancellation

As we have described earlier, in LoRa, collision occurs due the simultaneous arrival of two or more packets with same SF at the receiver. According to capture effect, strongest packet can be received successfully and other packet will be considered as interference. However, successive



(a)



(b)

Figure 4.15: Illustration of Downlink and Uplink NOMA model with two users and one gateway node.

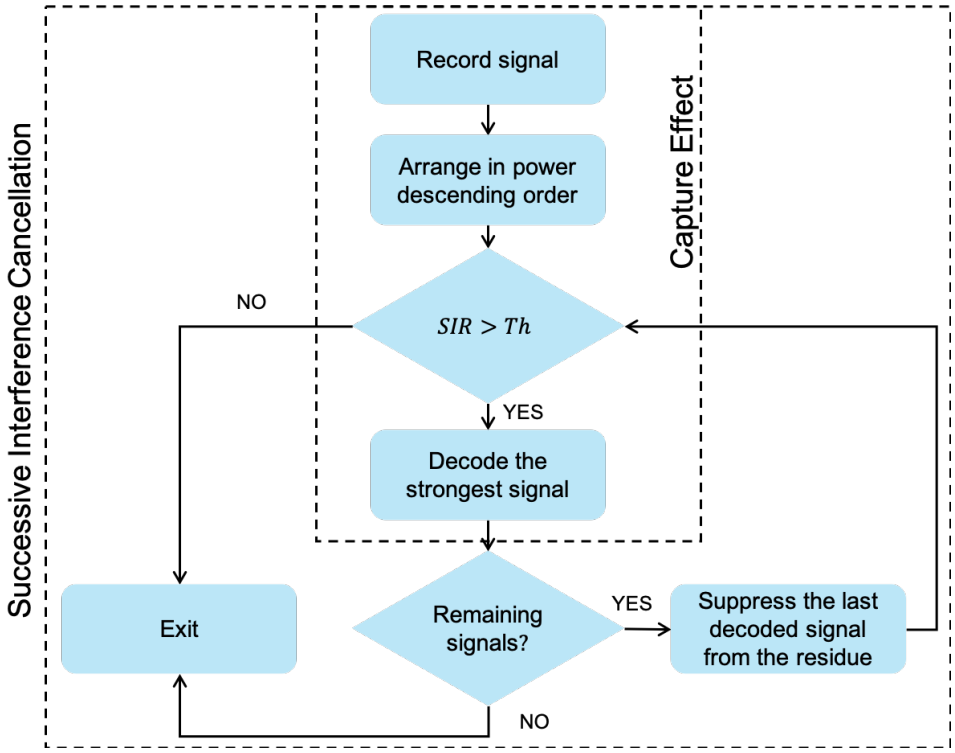


Figure 4.16: Block diagram for CE and SIC

interference cancellation facilitates the recovery of weaker packet too, as well. First stronger signal is decoded normally, decoded signal is then subtracted from the combined signal. Then weaker signal is extracted from the residue [112]. In case of multiple interferences, this can lead to an iterative process. The strongest signal is detected first from the received signal and then the next strongest and so on. After each signal's decoding, the received signal for that user can be reconstructed by recreating the transmit signal and applying an estimate of the channel to it. This can be subtracted from the composite received signal, which then allows subsequent users to experience a cleaner signal. A block diagram of CE and SIC is shown in Figure4.16.

In this work, we have considered SIC as pure receiver technique, that means it does not require any type of modification on the transmitter. Let assume a LoRa network consisting on an gateway node (receiver) and N transmitters scattered around the gateway node in Poisson field $\Phi = \{(L_i, H_i)\} \subset \mathbb{R}^d \times \mathbb{R}^+$. Where L_i represents the location of each transmitting node, H_i is the channel attenuation coefficient. The SIR based successful decoding of single user is defined in (4.33). Which says that a particular signal X at $L \in \Phi$ can be decoded successfully if its SIR is greater than some threshold. Any signal x_r can be decoded successfully from the residue of received signal $Y(t) = \sum_{active\ users} X_i(t)H_i(t)$, if its signal to residual interference plus noise power ratio SI_rR_x is:

$$SI_rR_x = \prod^{n} \frac{P_i}{P_{i+1}} \geq Th \tag{4.36}$$

Figure 4.17 and 4.18 show the probability of successful decoding of the packets and throughput of LoRa based system with collisions, consecutively. We have varied the network size N. The

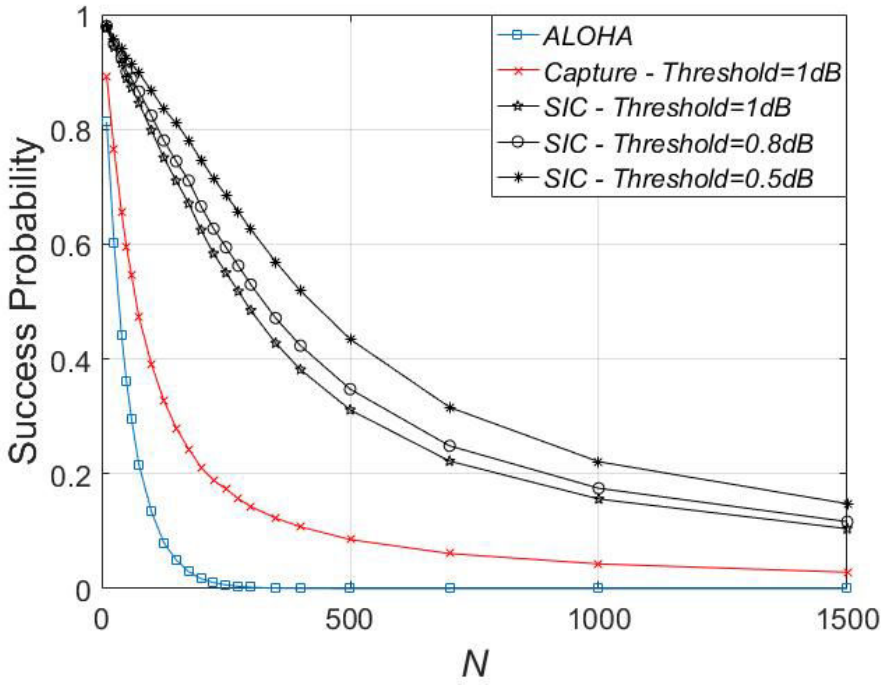


Figure 4.17: Probability of Successful Transmission.

assumption is that all the nodes are using same SF, and their packet generation rate is given in (4.9). It can observe that CE and SIC can significantly improve the network performance. This studied case is in the sense of worst case scenario because no channel coding is used which would significantly improve the performance.

6 Integration in CupCarbon

PHY layer of LoRa is added in CupCarbon as explained in this chapter. Data from the MAC layer is provided from the user (from SenScript), then a communication protocol can be assigned (whether 802.11 or 802.15.4 or LoRa). Figure 4.19 shows the block diagram of the whole procedure. In this case the user must assign the spreading factor too. The receiver technique should also neccesory to mention in the input. For this (if the communication protocol is LoRa), system needs spreading factor and the receiver tecnique. By default the receiver technique is none, the receiver will work normally without considering capture effect or successive interference cancellation.

Figure 4.20 shows the CupCarbon environment for the assignment of the parameters in case of selecting LoRa communication protocol. It shows interconnected sensor nodes placed arbitrarily. All the nodes are assigned the LoRa protocol. Nodes should be in the radio range of each other to communicate. Other parameters are explained in Chapter 3. In case of LoRa, the user can assign SF value. The SF is between $[7, 8, 9, 10, 11,$ and $12]$. The MAC uses ALOHA protocol, in this case a node transmits a packet as soon as it has data to send. ALOHA limits the system performance as it can only give 18.39% of maximum efficiency (4.25). However, CE and SIC can be used to enhance the system performance.

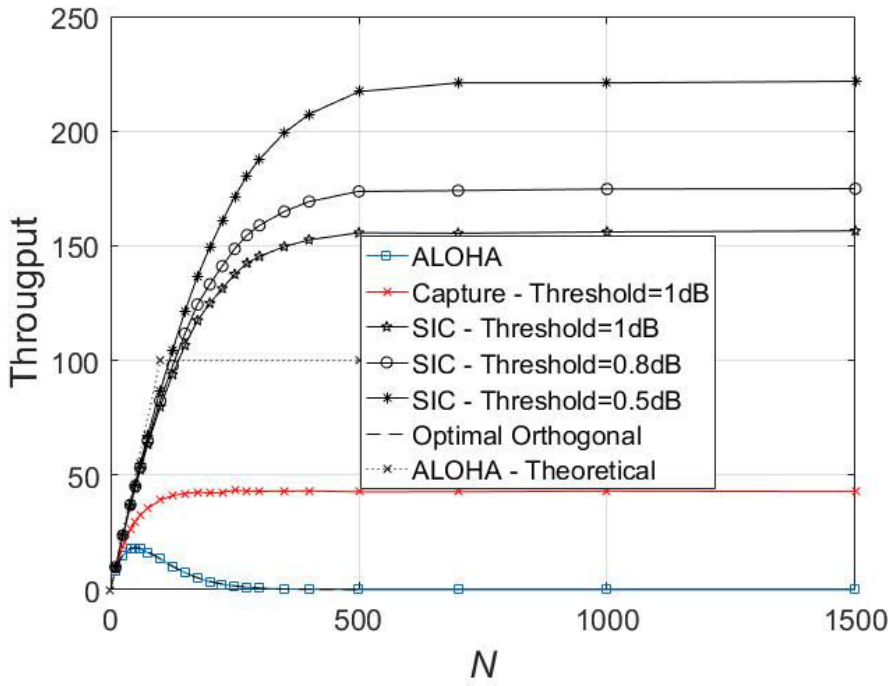


Figure 4.18: Throughput

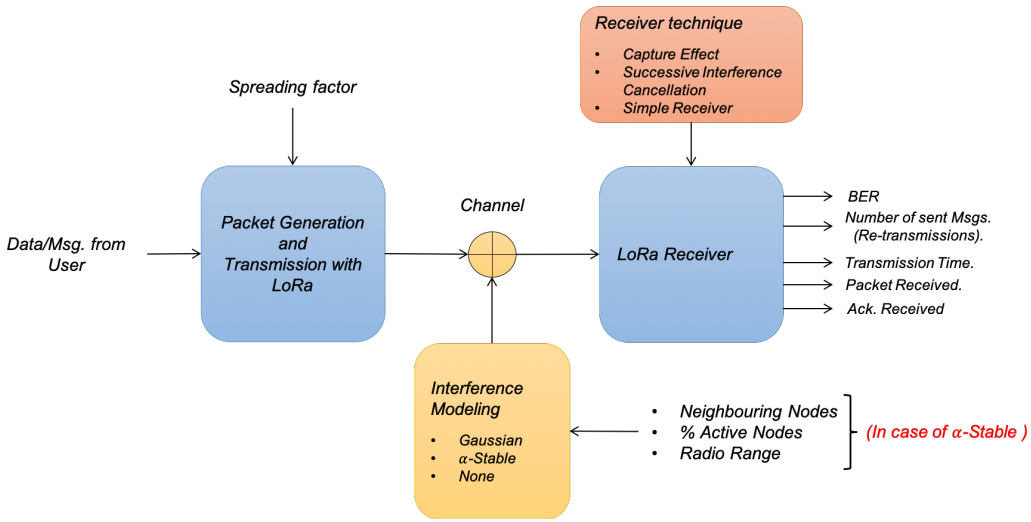


Figure 4.19: Schematic view of LoRa PHY, CE and SIC in CupCarbon.

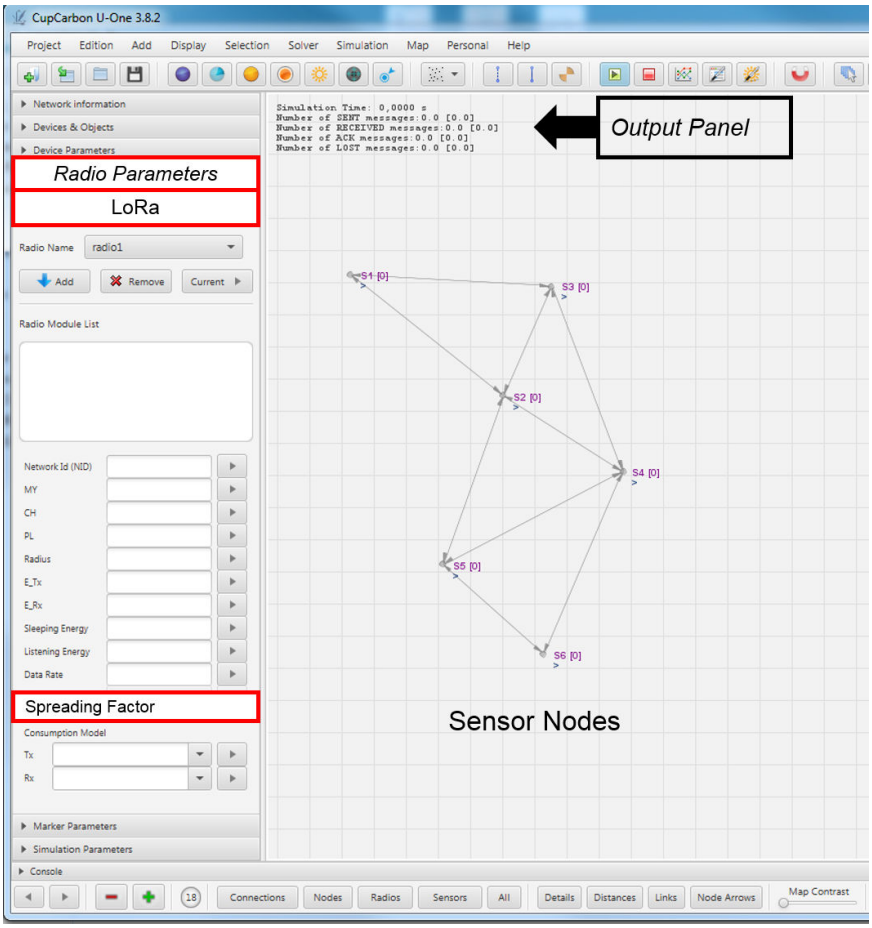


Figure 4.20: Radio parameter settings for LoRa in CupCarbon.

5

Conclusion and Perspective

The concept of the Internet of Things is maturing rapidly and soon we will be seeing the world wide interconnected network, integrating every physical device with other devices. The IoT has been focused exhaustively in recent years due to its wider applications like smart cities. The IoT constitutes a new step of evolution of the technology. On the other hand, it is also facing certain constraints regarding modern telecommunications. With a constantly growing number of connected wireless devices and a lack of available radio spectrum in the near future, interference can make impossible the efficient use of connected devices and further development of the IoT. Most of WSN uses unlicensed ISM frequency band that makes interference probable phenomenon on a given channel. System throughput gets influenced by the interference as it can congest wireless medium, cause packet drops, re-transmissions, link instability, and inconsistent protocol behavior. Most of the available literature on WSN have not considered interference effect from inside or outside of the network unless nodes are in radio range of each other and system have unlimited radio spectrum. In contrast, real-time WSN are band limited and nodes can interfere with other nodes even if they are beyond their communication range. To outcome this problem, the impact of interference on radio communications in IoT and particularly in WSN context needs to be studied and efficient adaptive communication strategies must be developed.

The rapid growth in the field of WSNs entails the need of creating new simulators that have more specific capabilities to tackle interference and multipath propagation effects that are present in the wireless environment. Finding a suitable simulation environment that allows researchers to verify new ideas and compare proposed solutions in a virtual environment is a difficult task. Most of the existing simulators are mainly used to develop routing protocols. However, They do not offer real-time interference modeling. The existing simulators, like NS-2/3 [29], OMNET (Castalia) [31], TOSSIM [32], OPNET [54], WSNet [30], etc., are mainly used to develop new routing protocols. However, in the context of smart cities and IoT, their interference models are very simple and do not take into account the real city environment.

This thesis was proposed in the ANR project PERSEPTEUR (http://pagesperso.univ- brest.fr/ bounceur/anr/pers
The thesis was conducted in the two research laboratories: Lab-STICC and IEMN (IRCICA). The proposed thesis is part of a research on the deployment of sensor networks in urban environment. It covers the interference modeling and communication protocol between sensor nodes. The main idea behind this thesis is propose an optimal simulator which also takes into account a realistic evaluation of the wireless interference in a $3D$ environment.

1 Objectives

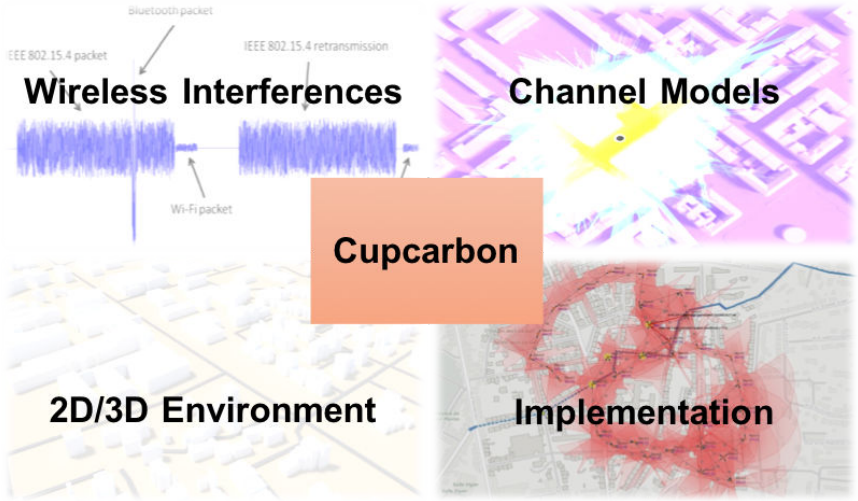


Figure 5.1: Main components of the CupCarbon platform.

CupCarbon is an open source WSN simulator which is recently developed under the PERSEPTEUR project. Its plateform is composed of 4 main parts: a 2D/3D environment block, an interference block, a radio channel block and an implementation block (shown in the Figure 5.1). The interference block is the main contribution from this thesis. It can be further divide into two catagories:

1. PHY Layer: a significant originality of CupCarbon is to be able to implement the true physical layers of some of the major LPWAN communication protocols like ZigBee, Wi-Fi, and LoRa. The evaluation of a link quality can now be based on accurate transmission conditions that consider the radio channel and the data encoding. This approach, however, can be very time consuming if we need to generate all the signals in the city. That is why we have also proposed and included some statistical approaches to model interference.

2. Interference: interference is a significantly limiting factor in dense networks. An exact evaluation of interference is feasible and CupCarbon allows a generation of all the received signals with either ZigBee, Wi-Fi or LoRa protocol. However, such an approach can be very time consuming in a big network due to the possibility of high number of interfering sources. The simulator has to generate all the interfering signals, simulate all the associated channels and then combine all interferers to see the actual interference. To avoid these heavy calculations, it is possible to use statistical models. However, as shown in [70][28], a Gaussian model is not accurate enough to represent wireless conditions. One solution is to generate the global interference with a single distribution. Our proposal is to use α-stable distribution with parameters depending on the radio channel statistics and the interferer density.

2 Contributions

We have analyzed the performance of PHY layers of wireless communication systems based on IEEE standards 802.15.4, 802.11 and LoRa, adopting α-stable interference model with its parameter estimation in more realistic way, for the deployment of WSN in the urban environment. We have presented the mathematical model of the distribution of network interference. For a flexible and accurate representation, interference is modeled by α-stable distribution function and we have estimated the values of its parameters, depending on a total number of nodes in specified area in the network. This allows the better estimation of wireless channel conditions. We have also proposed to add capture effect and successive interference cancellation in the LoRa receiver. The results show this can help to significantly improve the system performance.

We started with the study of the feasibility of the adding interference models in a WSN simulator CupCarbon. The first step in this was to add PHY layers in an optimal way in the simulator. Here the goal was to optimally minimize the simulation time and get the same results. And the comparison results for all the protocols (Wi-Fi, Zigbee, LoRa) in CupCarbon show that their performance is same. We have also added the simplified MAC in CupCarbon, which give results with some trade off. Data from the MAC layer is provided from the user (from SenScript), then a communication protocol is assigned (whether 802.11, 802.15.4 or LoRa).

As mentioned earlier, most of the WSN simulators either does not use interference modeling or only use simplified Gaussian model. This blind use of the simple models, gives false estimations with an unacceptable margin of error for the received power levels. Thus, the idea here is to use the more realistic interference model. Two interference models are used in the CupCarbon: the Gaussian model and the α-stable model.

Then we study the performance of above mentioned systems under both kinds of background noise (i.e., Gaussian and α-stable). From these simulation results, we conclude that if an impulsive background is mistakenly assumed as Gaussian in the evaluation, the system performance is largely overestimated. We have then estimated two out of four parameters of α-stable distribution depending on spatial density of wireless devices in specified area around the wireless node. Interference modeling with α-stable distribution is less time consuming approach. We have also compared the true results with this statistical approach.

As LoRa PHY is not openly available, we also provide a comprehensive analysis of technical features of the PHY layer associated with LoRa. We explore the performance and the capacity limitations of LoRaWAN and LoRa networks. We measure the network throughput by calculating the probability of successful transmissions with varied network sizes.

It was concluded that sensing the channel is not resolve the collision and interference problem because of the very wide coverage of access points. Then capture effect and successive interference cancellation are incorporated in the LoRa based system and simulation results show significant improvement in the probability of successful reception rate. According to capture effect, strongest packet can be received successfully and other packet will be considered as interference. However, SIC facilitates the recovery of weaker packet too, as well. This proves that collision does not always lead to loss of all frames. In some cases, even all of colliding packets can be decoded successfully. It is also observed that overlapping length and interferer signal's received power affect the capture and SIC performance.

3 Impact of the Thesis

This thesis led to six publications. The first two publications were related to the work with adding OFDM and Gaussian modeling of interference in the CupCarbon. Then other two were related to the work on the ZigBee and α-stable distribution. Here we estimated the parameters of the α-stable distribution based on the network density. The last two publications were related to the including LoRa with CE and SIC in CupCarbon.

4 Perspectives

This work has open multiple new directions for future work. Some points in broad way are mentioned below.

- For CupCarbon, other architectures can be developed in the future, where we shall add the possibility of a direct, ad hoc, communication between nodes using other LPWAN protocols.
- A study of spatial dependence in the interference model using Copulae can also be included in the CupCarbon.
- Now that we know that with capture effect and successive interfernce cancellation, we can recover from packet collisions, it could be worth studying whether it is helpful to adjust the MAC in such a way that it allows for collisions that can be recovered with high probability. As a result, the MAC layer could be tuned to exploit the SIC capability and thus increase the network capacity
- Motivated by the work in this disertion, we believe that future research activities in the field of urban IoT for smart cities should pay proper attention to WB-IoT.

List of Publications

1. Umber Noreen, Ahcène Bounceur and Laurent Clavier. LoRa-like CSS-based PHY layer, Capture Effect and Serial Interference Cancellation (24th European Wireless 2018, Catania Italy).
2. Umber Noreen, Ahcène Bounceur and Laurent Clavier. A study of LoRa low power and wide area network technology (International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), 2017)
3. Umber Noreen, Ahcène Bounceur and Laurent Clavier. Modeling Interference for Wireless Sensor Network Simulators (The International Conference on Future Networks and Distributed Systems (ICFNDS 2017) Cambridge, UK).
4. Umber Noreen, Ahcène Bounceur and Laurent Clavier. A Study of LoRa Low Power and Wide Area Network Technology (INTERNATIONAL CONFERENCE on ADVANCED TECHNOLOGIES FOR SIGNAL and IMAGE PROCESSING (ATSIP'17) Fez, Morocco).
5. Umber Noreen, Ahcène Bounceur and Laurent Clavier. Performance Evaluation of IEEE 802.15.4 PHY with Impulsive Network Interference in CupCarbon (International Symposium on Networks, Computers and Communications (ISNCC'16) Hammamet, Tunisia)
6. Umber Noreen, Ahcène Bounceur and Laurent Clavier. Integration of OFDM Based Communication System with Alpha-Stable Interference using CupCarbon Simulator (International Conference on Internet of Things, Data and Cloud Computing (ICC'16) Cambridge, United Kingdom)
7. Ahcène Bounceur, Madani Bezoui, Umber Noreen, Reinhardt Euler, Farid Lalem, Mohammad Hammoudeh, Sohail Jabbar. LOGO: A New Distributed Leader Election Algorithm in WSNs with Low Energy Consumption. (ICFITT EAI International Conference on Future Internet Technologies and Trends)
8. Ahcène Bounceur, Laurent Clavier, Pierre Combeau, Olivier Marc, Rodolphe Vauzelle, Arnaud Masserann, Julien Soler, Reinhardt Euler, Taha Alwajeeh, Vyas Devendra, Umber Noreen, Emilie Soret, Massinissa Lounis. CupCarbon: A new platform for the design, simulation and 2D/3D visualization of radio propagation and interferences in IoT networks (15th IEEE Annual Consumer Communications and Networking Conference (CCNC), 2018)

Conference

1. Umber Noreen, Ahcène Bounceur and Laurent Clavier. Integration of OFDM-based Communication System in CupCarbon Simulator (11th Symposium of the GDR SoC-SiP, Nantes, France)
    - **Poster**: Integration and Performance Evaluation of OFDM-based PHY with Impulsive Network Interference in CupCarbon Simulator.
2. Umber Noreen, Ahcène Bounceur, Laurent Clavier and Olivier Marc. LoRa: LPWAN Technology for Smart City (12th Symposium of the GDR SoC-SiP, Bordoux, France)
    - **Poster**: Modeling Interference for LoRa Based WSN in CupCarbon.

Bibliography

[1] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, Aug 2002. 3

[2] Chien-Chung Shen, C. Srisathapornphat, and C. Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, 8(4):52–59, Aug 2001. 3

[3] U. Raza, P. Kulkarni, and M. Sooriyabandara. Low power wide area networks: An overview. *IEEE Communications Surveys Tutorials*, 19(2):855–873, Secondquarter 2017. 4, 11, 77

[4] S. R. J. Ramson and D. J. Moni. Applications of wireless sensor networks — a survey. In *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, pages 325–329, Feb 2017. 4

[5] M. Z. Win, P. C. Pinto, and L. A. Shepp. A mathematical theory of network interference and its applications. *Proceedings of the IEEE*, 97(2):205–230, Feb 2009. 4, 15, 20, 21

[6] O. Vermesan and P. Friess. *Digitising the Industry Internet of Things Connecting the Physical, Digital and Virtual Worlds*. River Publishers, 2016. 6

[7] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014. 6

[8] S. Jirka I. Simonis T. Everding C. Stasch S. Liang A. Bröring, J. Echterhoff and R. Lemmens. New generation sensor web enablement. *Sensors*, 11(3):2652–99, 2011. 7

[9] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet, 2002. 8

[10] Lora alliance. `https://www.lora-alliance.org`. Accessed: 2018-06-25. 8

[11] Sigfox. `http://www.sigfox.com`. Accessed: 2018-06-25. 8

[12] Shahin Farahan. *ZigBee Wireless Networks and Transceivers*. Elsevier/ Newnes, 2008. 8

[13] Pascal Thubert, Tim Winter, Anders Brandt, Jonathan Hui, Richard Kelsey, Phil Levis, Kris Pister, Rene Struik, Jp Vasseur, and Roger Alexander. Rpl: Ipv6 routing protocol for low power and lossy networks. *Internet Engineering Task Force (IETF)*, RFC 6550, 03 2012. 8

[14] Raj Jain. Channel models: A tutorial. *WiMAX Forum AATG.*, 2007. 10

[15] S. Kurt and B. Tavli. Propagation model alternatives for outdoor wireless sensor networks. In *2013 IFIP Wireless Days (WD)*, pages 1–3, Nov 2013. 10

[16] Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer. *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pages 1–225, April 2012. 13, 14

[17] S. Shakkottai, T. S. Rappaport, and P. C. Karlsson. Cross-layer design for wireless networks. *IEEE Communications Magazine*, 41(10):74–80, Oct 2003. 14

[18] Salah R. Makar M. Badawi G. Kenawy A. Halawa H. Refaat T. Daoud R. Amer H. ElSayed H. Onsy, M. and M. Soudani. Performance of wsns under the effect of collisions and interference. *Wireless Sensor Network*, (6):93–103, 2014. 15, 43

[19] P. G. Georgiou, P. Tsakalides, and C. Kyriakakis. Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise. *IEEE Transactions on Multimedia*, 1(3):291–301, Sept 1999. 15, 16

[20] P. C. Pinto and M. Z. Win. Communication in a poisson field of interferers–part i: Interference distribution and error probability. *IEEE Transactions on Wireless Communications*, 9(7):2176–2186, July 2010. 15

[21] A. Mahmood, M. Chitre, and M. A. Armand. Detecting ofdm signals in alpha-stable noise. *IEEE Transactions on Communications*, 62(10):3571–3583, Oct 2014. 15

[22] Ata Elahi Adam Gschwender. *ZigBee Wireless Sensor and Control Network*. Prentice Hall, October 2009. 15

[23] Md. Mizanur Rahman A. Z. M. Touhidul Islam Nuzhat Tasneem Awon, Md. Ashraful Islam. Effect of awgn & fading (raleigh & rician) channels on ber performance of a wimax communication system. International Journal of Computer Science and Information security(IJCSIS), November 2012. 15, 16

[24] Ilan Shomorony and Amir Salman Avestimehr. Worst-case additive noise in wireless networks. *CoRR*, abs/1202.2687, 2012. 15

[25] Gao P. Wang X., Li K. and Meng S. Research on parameter estimation methods for alpha stable noise in a laser gyroscope's random error. *Sensors*, 15(8), 2015. 17

[26] M. Shao and C. L. Nikias. Signal processing with fractional lower order moments: stable processes and their applications. *Proceedings of the IEEE*, 81(7):986–1010, July 1993. 17

[27] Min Shao Chrysostomos L. Nikias. *Signal processing with alpha-stable distributions and applications*. John Wiley & Sons Inc, New York, USA, 1995. 17

[28] H. E. Ghannudi, L. Clavier, N. Azzaoui, F. Septier, and P. Rolland. α-stable interference modeling and cauchy receiver for an ir-uwb ad hoc network. *IEEE Transactions on Communications*, 58(6):1748–1757, June 2010. 17, 21, 39, 100

[29] The network simulator ns-2. `http://www.isi.edu/nsnam/ns/ns-documentation.html`. Accessed: 2015-06-10. 22, 28, 99

[30] Wsnet simulator for large scale wireless sensor networks. `http://wsnet.gforge.inria.fr/`. Accessed: 2015-11-15. 22, 99

[31] Castalia. a simulator for wireless sensor networks and body area networks. user's manual. `http://castalia.npc.nicta.com.au/pdfs/Castalia-UserManual.pdf`. Accessed: 2010-11-15. 22, 32, 99

[32] Tossim - tinyos documentation wiki. `http://docs.tinyos.net/index.php/TOSSIM`. Accessed: 2016-11-15. 22, 29, 99

[33] A. Bounceur, O. Marc, M. Lounis, J. Soler, L. Clavier, P. Combeau, R. Vauzelle, L. Lagadec, R. Euler, M. Bezoui, and P. Manzoni. Cupcarbon-lab: An iot emulator. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–2, Jan 2018. 25

[34] Mehdi Kamal, Massinissa Lounis, Ahcène Bounceur, and M. Tahar Kechadi. Cupcarbon: a multi-agent and discrete event wireless sensor network design and simulation tool. In *SimuTools*, 2014. 25

[35] Cupcarbon, anr project persepteur. cupcarbon simulator. http://www.cupcarbon.com. Accessed: 2015-11-15. 25, 63

[36] C. P. Singh, O. P. Vyas, and M. K. Tiwari. A survey of simulation in sensor networks. In *2008 International Conference on Computational Intelligence for Modelling Control Automation*, pages 867–872, Dec 2008. 25

[37] M. Youssef and N. El-Sheimy. Wireless sensor network: Research vs. reality design and deployment issues. In *Fifth Annual Conference on Communication Networks and Services Research (CNSR '07)*, pages 8–9, May 2007. 26

[38] J. Heidemann, K. Mills, and S. Kumar. Expanding confidence in network simulations. *IEEE Network*, 15(5):58–63, Sept 2001. 27

[39] A. S. Martínez-Sala P. Pavón-Mariño J. García-Haro E. Egea-López, J. Vales-Alonso. Simulation tools for wireless sensor networks. *Summer Simulation Multiconference - SPECTS*, 2005. 27

[40] M. Mekni and B. Moulin. A survey on sensor webs simulation tools. In *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, pages 574–579, Aug 2008. 27

[41] WeiChung;Hu MingLun;Lee TzungShian;Tsai Hewijin Christine Jiau. A gui simulation model in supporting embedded software design. http://asiair.asia.edu.tw/ir/handle/310904400/5742, December 2009. Accessed: 2017-11-15. 27

[42] David M. Nicol. Scalability of network simulators revisited. *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2003. 28

[43] Ekram Hossain Teerawat Issariyakul. *Introduction to Network Simulator NS2*. Springer Science+Business Media, LLC, 2009. 28

[44] Matt Welsh Philip Levis, Nelson Lee and David Culler. Tossim: accurate and scalable simulation of entire tinyos applications. *SenSys '03 Proceedings of the 1st international conference on Embedded networked sensor systems*, November 2003. 29

[45] Glomosim. http://pcl.cs.ucla.edu/projects/glomosim/. Accessed: 2017-11-15. 30

[46] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. In *Proceedings. Twelfth Workshop on Parallel and Distributed Simulation PADS '98 (Cat. No.98TB100233)*, pages 154–161, May 1998. 31

[47] Avrora. http://compilers.cs.ucla.edu/avrora. Accessed: 2017-11-15. 31

[48] Ben L. Titzer and et al. Avrora: Scalable sensor network simulation with precise timing. In *IN PROC. OF THE 4TH INTL. CONF. ON INFORMATION PROCESSING IN SENSOR NETWORKS (IPSN*, pages 477–482, 2005. 31

[49] Sense. http://www.ita.cs.rpi.edu/sense/index.html. Accessed: 2018-06-25. 31

[50] Gilbert Chen, Joel Branch, Michael Pflug, Lijuan Zhu, and Boleslaw Szymanski. *Sense: A Wireless Sensor Network Simulator*, chapter Chqpter 1, pages 249–267. Springer US, Boston, MA, 2005. 31

[51] G. Simon, P. Volgyesi, M. Maroti, and A. Ledeczi. Simulation-based optimization of communication protocols for large-scale wireless sensor networks. In *2003 IEEE Aerospace Conference Proceedings*, volume 3, March 2003. 32

[52] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt. Cross-level sensor network simulation with cooja. In *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pages 641–648, Nov 2006. 32

[53] F. Österlind. A sensor network simulator for the contiki os. *Swedish Institute of Computer Science*, 2006. 32

[54] Joakim Eriksson, Fredrik Österlind, Niclas Finne, Nicolas Tsiftes, Adam Dunkels, Thiemo Voigt, Robert Sauter, and Pedro Marrón. Cooja/mspsim: interoperability testing for wireless sensor networks, 01 2009. 32, 99

[55] Sung Park, Andreas Savvides, and Mani B. Srivastava. Sensorsim: a simulation framework for sensor networks, 01 2000. 32

[56] Sung Park, A. Savvides, and M. B. Srivastava. Simulating networks of wireless sensors. In *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*, volume 2, pages 1330–1338 vol.2, Dec 2001. 33

[57] Alexander Kröller, Dennis Pfisterer, Sándor P. Fekete, and Stefan Fischer. *Algorithms and Simulation Methods for Topology-Aware Sensor Networks*, pages 380–400. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. 33

[58] A. Kröller, D. Pfisterer, C. Buschmann, S. P. Fekete, and S. Fischer. Shawn: A new approach to simulating wireless sensor networks, 2005. 33

[59] J. Elson, S. Bien, N. Busek, V. Bychkovskiy, A. Cerpa, D. Ganesan, L. Girod, B. Greenstein, T. Schoellhammer, T. Stathopoulos, and D. Estrin. Emstar: An environment for developing wireless embedded systems software, 2003. 33

[60] Lewis Girod, Jeremy Elson, Thanos Stathopoulos, Martin Lukac, and Deborah Estrin. Emstar: a software environment for developing and deploying wireless sensor networks. In *In Proceedings of the 2004 USENIX Technical Conference*, pages 283–296, 2004. 34

[61] J-sim. http://nsr.bioeng.washington.edu/jsim/. Accessed: 2017-11-15. 34

[62] Ahmed Sobeih and Jennifer C. Hou. A simulation framework for sensor networks in j-sim. Technical report, University of Illinois at Urbana-Champaign, 2003. 34

[63] Ahmed Sobeih, Wei-Peng Chen, J. C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, and Honghai Zhang. J-sim: a simulation environment for wireless sensor networks. In *38th Annual Simulation Symposium*, pages 175–187, April 2005. 34

[64] Visualsense. http://ptolemy.eecs.berkeley.edu/visualsense/. Accessed: 2017-11-15. 34

[65] Philip Baldwin, Sanjeev Kohli, Edward A. Lee, Xiaojun Liu, Yang Zhao, Contributors C. T. Ee, Christopher Brooks, N. V. Krishnan, Stephen Neuendorffer, Charlie Zhong, and Rachel Zhou. Visualsense: Visual modeling for wireless and sensor network systems. Technical report, UCB ERL Memorandum UCB/ERL M04/8, 2004. 34

[66] Jprowler. http://www.isis.vanderbilt.edu/projects/nest/jprowler/index.html. Accessed: 2017-11-15. 35

[67] A. Bounceur, L. Clavier, P. Combeau, O. Marc, R. Vauzelle, A. Masserann, J. Soler, R. Euler, T. Alwajeeh, V. Devendra, U. Noreen, E. Soret, and M. Lounis. Cupcarbon: A new platform for the design, simulation and 2d/3d visualization of radio propagation and interferences in iot networks. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–4, Jan 2018. 37

[68] A. Bounceur, O. Marc, M. Lounis, J. Soler, L. Clavier, P. Combeau, R. Vauzelle, L. Lagadec, R. Euler, M. Bezoui, and P. Manzoni. Cupcarbon-lab: An iot emulator. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 1–2, Jan 2018. 38

[69] K. S. Ting, G. K. Ee, C. K. Ng, N. K. Noordin, and B. M. Ali. The performance evaluation of ieee 802.11 against ieee 802.15.4 with low transmission power. In *The 17th Asia Pacific Conference on Communications*, pages 850–855, Oct 2011. 39

[70] M. L. de Freitas, M. Egan, L. Clavier, A. Goupil, G. W. Peters, and N. Azzaoui. Capacity bounds for additive symmetric α -stable noise channels. *IEEE Transactions on Information Theory*, 63(8):5115–5123, Aug 2017. 39, 100

[71] Taha Alwajeeh, Pierre Combeau, Ahcène Bounceur, and Rodolphe Vauzelle. Efficient method for associating radio propagation models with spatial partitioning for smart city applications. pages 1–7, 03 2016. 41

[72] T. Alwajeeh, P. Combeau, R. Vauzelle, and A. Bounceur. A high-speed 2.5d ray-tracing propagation model for microcellular systems, application: Smart cities. In *2017 11th European Conference on Antennas and Propagation (EUCAP)*, pages 3515–3519, March 2017. 41

[73] Ginseng: Performance control in wireless sensor networks. `http://www.ict-ginseng.eu/`. 43

[74] M. Franceschinis, L. Gioanola, M. Messere, R. Tomasi, M. A. Spirito, and P. Civera. Wireless sensor networks for intelligent transportation systems. In *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pages 1–5, April 2009. 43

[75] Octav Chipara, Chenyang Lu, Thomas C. Bailey, and Gruia catalin Roman. Reliable clinical monitoring using wireless sensor networks: Experiences in a step-down hospital unit. Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, 2010. 43

[76] K. Hamied and G. Labedz. Amps cell transmitter interference to cdma mobile receiver. In *Proceedings of Vehicular Technology Conference - VTC*, volume 3, pages 1467–1471 vol.3, April 1996. 43

[77] Cisco. Wireless rf interference customer survey result. Technical report, Cisco and/or its affiliates, 2010. 44

[78] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey, 2002. 44

[79] P. Amirshahi, S. M. Navidpour, and M. Kavehrad. Performance analysis of uncoded and coded ofdm broadband transmission over low voltage power-line channels with impulsive noise. *IEEE Transactions on Power Delivery*, 21(4):1927–1934, Oct 2006. 45, 46, 49

[80] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, June 2005. 45, 47

[81] Carl Eklund, Roger B. Marks, Kenneth L. Stanwood, and Stanley Wang. Ieee standard 802.16: A technical overview of the wirelessman air interface for broadband wireless access. *IEEE COMMUNICATIONS MAGAZINE*, 40:98–107, 2002. 45

[82] M. S. Corson, R. Laroia, A. O'Neill, V. Park, and G. Tsirtsis. A new paradigm for ip-based cellular networks. *IT Professional*, 3(6):20–29, Nov 2001. 45

[83] R. Berezdivin, R. Breinig, and R. Topp. Next-generation wireless communications concepts and technologies. *IEEE Communications Magazine*, 40(3):108–116, March 2002. 45

[84] L. Kleinrock and F. Tobagi. Packet switching in radio channels: Part i - carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12):1400–1416, December 1975. 46

[85] W. Stallings. Ieee 8o2.11: wireless lans from a to n. *IT Professional*, 6(5):32–37, Sept 2004. 46

[86] Theodore S. Rappaport. *Wireless Communications*. Pearson, January 2002. 47

[87] Louis R. Litwin and Michael Pugel. The principles of ofdm multicarrier modulation techniques are rapidly moving from the textbook to the real world of modern communication systems. In *signal processing*, 2001. 47

[88] L. C. Tran, A. Mertins, Tadeusz A. Wysocki, L. C. Tran, A. Mertins, and T. A. Wysocki. Quasi-orthogonal spacetime-frequency codes. In *in MB-OFDM UWB communications," Proc. International Conference on Signal Processing and Communication Systems ICSPCS2007*, 2007. 49

[89] Mustafa Ergen. *Principles of OFDM*, pages 109–175. Springer US, Boston, MA, 2009. 49

[90] J. A. C. Bingham. Multicarrier modulation for data transmission: an idea whose time has come. *IEEE Communications Magazine*, 28(5):5–14, May 1990. 50

[91] Ceilidh Hoffmann, Submitted To, Kai yeung Siu, Arthur C. Smith, and Ceilidh Hoffmann. Code-division multiplexing, 2004. 50

[92] J. M. Cioffi. *Multi-Channel Modulation*, chapter 4. 09 Jan. 2011. 52

[93] Werner Henkel and Oana Graur. *OFDM - Concepts for Future Communication Systems*. Springer, 01 2011. 55

[94] ZigBee Alliance. Zigbee specification. Technical report, ZigBee Standards Organization, 2400 Camino Ramon, Suite 375 San Ramon, CA 94583, 2005. 56

[95] Paolo Baronti, Prashant Pillai, Vince Chook, Stefano Chessa, Alberto Gotta, and Y. Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and zigbee standards. *Computer Communications*, pages 1655–1695, 2007. 56

[96] IEEE Std. Approved ieee draft amendment to ieee standard for information technology-telecommunications and information exchange between systems-part 15.4:wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans): Amendment to add alternate phy (amendment of ieee std 802.15.4). *IEEE Approved Std P802.15.4a/D7, Jan 2007*, 2007. 56

[97] Anis Koubaa, Mário Alves, and Eduardo Tovar. Ieee 802.15.4 for wireless sensor networks: A technical overview. Technical report, IPP-HURRAY! GROUP, POLYTECHNIC INSTITUTE OF PORTO (ISEP-IPP), PORTUGAL, 01 2005. 56, 59

[98] IEEE Std. Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans). *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pages 1–314, Sept 2011. 56

[99] IEEE Std. Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.3: Wireless medium access control (mac) and physical layer (phy) specifications for high rate wireless personal area networks (wpan). *IEEE Std 802.15.3-2003*, pages 1–360, Sept 2003. 58, 59

[100] U. Noreen, A. Bounceur, L. Clavier, and R. Kacimi. Performance evaluation of ieee 802.15.4 phy with impulsive network interference in cupcarbon simulator. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6, May 2016. 61

[101] Z. Yun and M. F. Iskander. Ray tracing for radio propagation modeling: Principles and applications. *IEEE Access*, 3:1089–1100, 2015. 65

[102] Lora alliance. `https://www.lora-alliance.org/What-Is-LoRa/Technology`. Accessed: 2016-11-15. 78

[103] Semtech. `http://www.semtech.com/images/datasheet/an1200.22.pdf`. Accessed: 2017-11-15. 79, 82

[104] David J. Wetherall Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, 2010. 80

[105] ETSI EN 300 220-1 V2.4.1. Electromagnetic compatibility and radio spectrum matters (erm) short range devices (srd) radio equipment to be used in the 25 mhz to 1000 mhz frequency range with power levels ranging up to 500 mw. Technical report, European Telecommunications Standards Institute 2012., 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, 2012. 81

[106] B. Reynders and S. Pollin. Chirp spread spectrum as a modulation technique for long range communication. In *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, pages 1–5, Nov 2016. 81

[107] Utz Roedig Martin Bor, John Vidler. Lora for the internet of things. *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, pages 361–366, 2016. 89, 90

[108] Martin C. Bor, Utz Roedig, Thiemo Voigt, and Juan M. Alonso. Do lora low-power wide-area networks scale? In *MSWiM*, 2016. 89, 90

[109] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. Mitigating inter-network interference in lora networks. In *:*, 2017. 89, 90

[110] Selahattin Kosunalp, Paul Daniel Mitchell, David Grace, and Tim Clarke. Experimental study of the capture effect for medium access control with aloha. *ETRI Journal*, 37(2):359–368, 4 2015. 89, 90

[111] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the capture effect for collision detection and recovery. In *The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II.*, pages 45–52, May 2005. 89, 90

[112] J. G. Andrews. Interference cancellation for cellular systems: a contemporary overview. *IEEE Wireless Communications*, 12(2):19–29, April 2005. 95

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

UBO
Université de Bretagne Occidentale

Titre : Modélisation d'Interférence pour Simulateur 3D de Réseaux de Capteurs Dédiés aux Villes Intelligentes

Mots clés : Distribution alpha-stable ; CupCarbon ; IEEE 802.11 ; IEEE 802.15.4 ; Internet des objets ; LoRa ;

Résumé : La plupart des réseaux WSN utilisent une bande passante industrielle, scientifique et médicale (ISM) sans licence, qui crée un phénomène probable d'interférence sur un canal donné. Le débit du système est influencé par les interférences car il peut être encombré, causant des pertes de paquets, des retransmissions, une instabilité de liaison et un comportement de protocole incohérent. Dans la recherche sur les réseaux de capteurs sans fil, la simulation est l'une des approches essentielles pour évaluer et évaluer un protocole de système ou de performance. La précision des résultats estimés dépend des paramètres de simulation sélectionnés. Dans les analyses existantes sur WSN, des modèles d'interférence simples sont utilisés dans les simulations. Cependant, ces modèles d'interférence ne sont pas assez précis pour l'analyse pratique d'applications de réseau de capteurs sans fil. De plus, la croissance rapide dans le domaine des réseaux WSN implique la nécessité de créer de nouveaux simulateurs dotés de capacités plus spécifiques pour lutter contre les effets de propagation par brouillage et par trajets multiples. La recherche a pour objectif principal de rechercher un environnement de simulation approprié permettant aux chercheurs de vérifier de nouvelles idées et de comparer les solutions futures.

Title : Interference Modeling for 3D Simulator of Sensor Networks dedicated to Smart Cities

Keywords : Alpha-Stable Distribution ; CupCarbon ; IEEE 802.11 ; IEEE 802.15.4 ; Internet of Things ; LoRa ;

Abstract : Most of WSNs use unlicensed Industrial, Scientific and Medical (ISM) frequency band that makes interference probable phenomenon on a given channel. System throughput gets influenced by the interference as it can congest wireless medium, cause packet drops, re-transmissions, link instability, and inconsistent protocol behavior. In wireless sensor network research, simulation is one of the essential approaches to asses and evaluate system or protocol performance. The accuracy of estimated results depends on selected simulation parameters. In existing analysis on WSN, simple interference models are used in simulations. However, these interference models are not accurate enough for practical wireless sensor network applications analysis. Moreover, The rapid growth in the field of WSNs entails the need of creating new simulators that have more specific capabilities to tackle interference and multipath propagation effects. Finding a suitable simulation environment that allows researchers to verify new ideas and compare proposed future solutions is main task of this research.