POLITECNICO DI TORINO

Master degree course in CCNE
Communication and Computer Network Engineering

Master Degree Thesis

Internet of Things

applications in the industry, health-care, and environment

Supervisors
Prof. Giorgio Taricco

Candidates
Amir MAHROKHZAD
Number: 250219

ACCADEMIC YEAR 2020-2021

Summary

The Internet of Things (IoT) has been considered by industries and research communities as an important element for wireless communications. It covers a wide range of applications, from telecommunications to the creation of smart communities, which improve many aspects of our daily lives.
Internet of Things is powered by the latest advances in smart sensors, RFID, communication technologies and Internet protocols.
This thesis provides an overview of the Internet of Things (IoT) according to application issues, enabling technologies and protocols.
The following is an overview of some of the main challenges of the Internet of Things. The relationship between the Internet of Things and big data analytics and cloud computing and fog are also considered.
Then, some IoT communication technologies have been studied in which Lora technology has been selected to be implemented in the tracking scenario.
In the next steps, Lora is simulated in Matlab to be studied in terms of performance in the AWGN channel model with fixed gain (no fading).

Acknowledgements

First of all, I would like to express my gratitude to my patient and supportive supervisor, Professor Giorgio Taricco, from the department of electronic and telecommunication at Politecnico di Torino.
I want to thank my parents for their love and their unwavering support and patience during the research and writing this thesis and my years of education.

Contents

List of Tables

List of Figures

Part I

Comprehensive survey of IOT

Chapter 1

Introduction

1.1 How IoT works

When we talk about the Internet of Things, we actually mean that everything, anywhere, anytime, is connected [1]. In fact, the IoT is an ecumenical network infrastructure with specific standards and protocols. All "things" from virtual to physical have identities, characteristics and personalities. [1]. The integration of sensors, actuators, RFID tags and communication technologies can be the foundation of the IoT and can show that all of these collaborations work together to achieve a common goal [1]. IoT has been used in many fields such as industrial control, smart cities, healthcare, traffic congestion, waste management, structural health, security, emergency services and logistics [1].
IoT makes a network for things to share the information with other connected things [4]. The first phase of the IoT may be the current Internet and mobile technology. The premise is the presence of sensors that must be intelligent to work without humans [4]. The conversion of objects from traditional to intelligent is done by using its underlying technologies such as communication technologies, sensors, Internet protocols, embedded devices, and all-encompassing computing [4].
In order to create a common environment for delivering quality products to companies, architectural standardization for IoT is the backbone [4]. Moreover, Considering management and supervision is very important to have high quality delivery services to the customer at an efficient cost [4].
RFID is a fundamental technology for the Internet of Things technology. It allows microchips to transmit information of identification to the reader via wireless communication. People can automatically detect, track and monitor

any objects attached with RFID tags by using RFID readers[4].
Wireless sensor networks (WSNs) is another fundamental IoT technology, which mainly use interconnected smart sensors for sensing and monitoring. Environmental monitoring, healthcare monitoring, industry monitoring, traffic monitoring, are some of its applications [4].

RFID and WSN are mainly effective in the IoT, and advances in them help the development of the IoT [4]. Moreover, we can mention many other technologies and devices like bar-codes, smart phones, social networks, and cloud computing, which are being used to form a wide network in order to support IoT [4].

In order to provide high-quality services to end users, IoT technical standards should be designed to define the specification for communication between things, processing and information exchange. Standardization is the key of IoT success that provides reliability, interoperability, compatibility and effective operation on the scale of global. IoT standards can cause great economic benefits, so, lots of organizations and countries are interested to extension of IoT standards. In the development of IoT standards, too many organizations are involved, so, a strong cooperation is necessary between them. Implementing IoT application and services by developers and users in a large scale can be done if accepted standards obtain [4].

1.2 Contributions of this thesis

This thesis comprehensively reviews the IoT using Lora communication technology developed by Semtech.
Lora is available today with over 167 million devices worldwide. IHS Markit predicts that by 2023, 43% of all LPWAN connections will be based on LoRa [25].
In the first part, the study of the Internet of Things is done through a systematic review of the scientific literature. Therefore, the main features and applications were discussed and also important challenges and issues were considered. With them in mind, Lora can be used as a complete technology to meet many of the needs of smart life.
In the second part, the tracking scenario is selected to be simulated to achieve the results of using Lora. Therefore, considering different parameters, Lora simulation was performed and the results were obtained. Taking into account various Spreading factors, we obtained new results on the bit error rate as

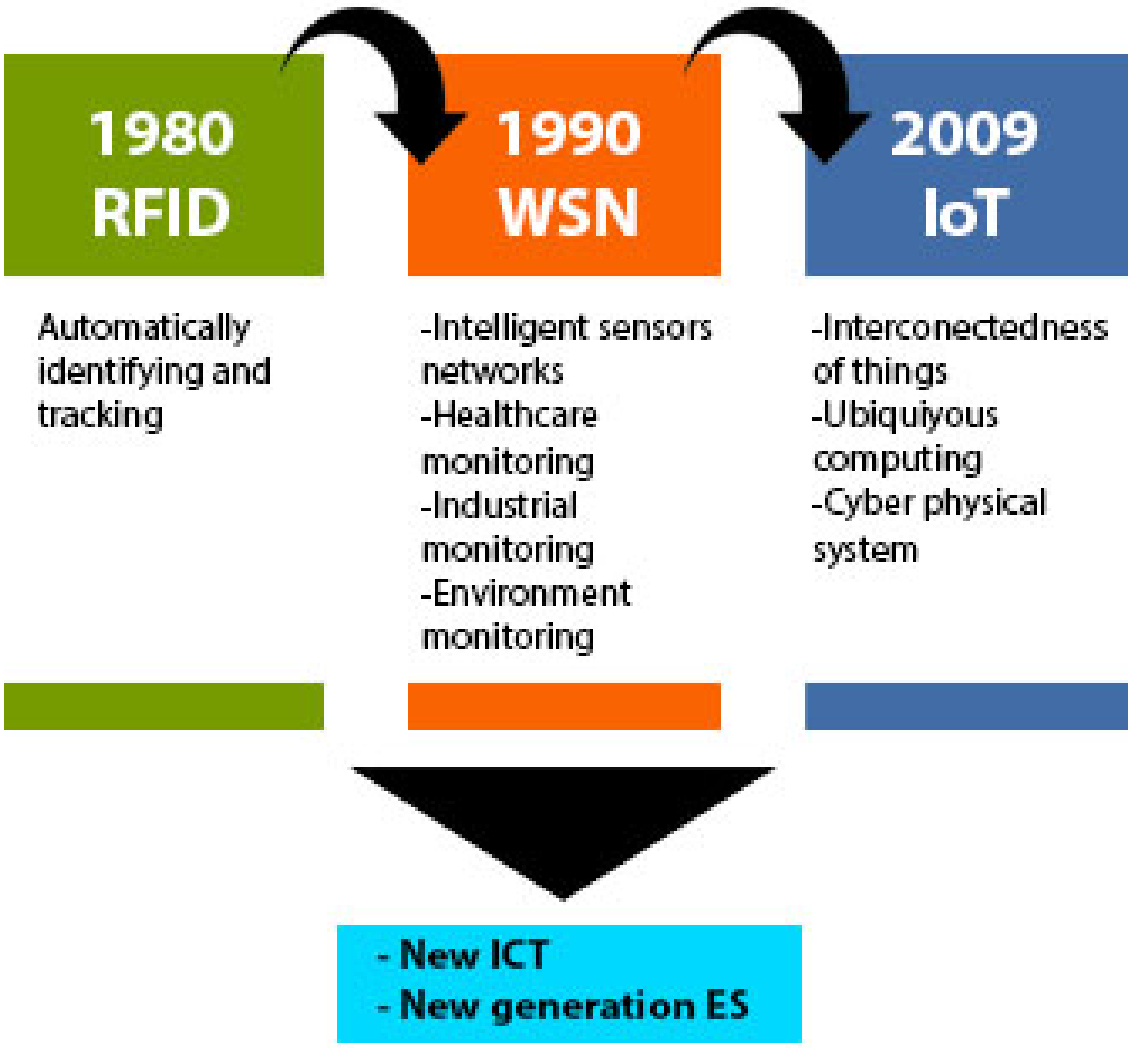

Figure 1.1. Technology of IoT-related and impact of IoT on new ICT

well as the bit rate. At different bandwidths, we encountered different behaviors of spreading factor and also sensitivity.
structure of this thesis is as follow:
Chapter 2 presents the existing IoT architectures, And the main elements of the Internet of Things were briefly reviewed. In Chapter 3, IoT challenges were discussed. Security is discussed in Chapter 4, and in Chapter 5, IoT technologies are introduced in general. Chapter 6 and 7 covered healthcare and industries in IoT. In the following, Chapter 8 surveyed the basis of simulation based on the AWGN channel model. In the second part, chapter 9,

Lora is comprehensively introduced to be used in Chapter 10 for the tracking scenario. Finally, in this chapter, Lora was studied analytically and then simulated and the results obtained. The conclusions are presented in the last chapter.

Chapter 2

IoT Architecture

IoT must have a potential to connect more than billions objects which are heterogeneous by the means of Internet. So, architecture of IoT need to be much flexible. There is a number of proposed architectures which still need to be a reference model. In the meanwhile, based on the analysis of needs, some projects try to design an architecture to be common. Between these proposed models, a 3-layer architecture consisting of the Application layer, Network layer, and Perception Layer is the basic model.

2.1 Three-Layer Architecture

2.1.1 Perception Layer

In the IoT architecture perception layer is implemented as the bottom layer. This layer interacts with physical device components by smart devices such as sensors, actuators, etc. Its main purposes are to connect things to the network of IoT and to measure, collect and process the information through deployed smart devices and then to transfer the processed information to upper layer through layer interfaces [2].

2.1.2 Network Layer

Network layer consist of various devices such as hub, switching, gateway, and also various communication technologies such as Bluetooth, Wi-Fi, long-term evolution, are integrated. So, it is the most important layer in the architecture of IoT. Network layer should receive processed information that perception layer provided them and then determine the routes in order to

transmit data and information via integrated network to the IoT hub, devices and also applications. This layer has to transmit data from or to different things or applications [2].

2.1.3 Application Layer

This layer is implemented in the architecture of IoT as the top layer. It receives transmitted data from the network layer and then provides operations or services which are required. As an example, this layer can provide storage in order to make backup of the received data. In this layer, many applications which have various requirements exist [2].

The three-layer architecture is fundamental to IoT and has been designed and implemented in many systems. However, despite the simplicity of the IoT multi-layered architecture, the operations and functions in the network and application layers are varied and complex. As an example, the network layer not only have to route and transfer data, even it should provide data services. So, in order to establish a flexible multi-layer architecture to provide data services in IoT, a service layer should be developed between application layer and network layer. Accordingly, in order to support the IoT, service-oriented architectures (SoA) have been developed [2].

2.2 Service-Oriented Architecture (SOA) for IoT

SOA can be used to support IoT for interacting heterogeneous devices as a key technology. It used in some research areas as WSNs, vehicular network and cloud computing [4]. Table 2.1 shows four layered SOA of IoT from the functionality point of view, and four layers' interaction can be seen in figure 2.1.

IoT architecture is related to architectural styles, smart objects, application and web services, networking and communication, security, cooperative data processing and etc. From the technology point of view, IoT architecture designing should consider modularity, scalability, extensibility and interoperability between heterogeneous devices. Since things need to move and to interaction with environment in real-time, an architecture is needed to dynamically help for interacting devices with other things. Since the nature of

Layers	Description
Sensing layer	This layer is integrated with existing hardware (RFID, sensors, actuators, etc) to sense/control physical world and acquire data.
Networking layer	This layer provides basic networking support and data transfer over wireless or wired network.
Service layer	This layer creates and manages services. It provides services to satisfy user needs.
Interface layer	This layer provides interaction methods to users and other applications.

Table 2.1. A four layered architecture for IoT

IoT is decentralized and heterogeneous, IoT requires that the architecture provides IoT efficient event-driven capability. A good approach has been considered by SOA to reach interoperability between heterogeneous devices [4].

2.2.1 Sensing Layer

When number of devices which equipped with sensors or RFID increase, interconnecting between Things will be simpler. So, the wireless systems which are equipped with sensors, in the sensing layer can sense and also exchange information automatically with other devices. Also, in some industry sections, universal unique identifier (UUID) and intelligent service deployment schemes are assigned to any required devices or services. UUIDs are much important to deploying successful services in a network. Any device with UUID can be identified and retrieved simply [4].

2.2.2 Networking Layer

This layer has capability of collecting information from existing IT infrastructure such as healthcare system, transportation system and business system. Moreover, it has a main role which is all the things should connect together and share information. In SOA, usually services which are provide by Things are located in a heterogeneous network and all related things are brought in to service Internet. QoS control and management is probably involved in

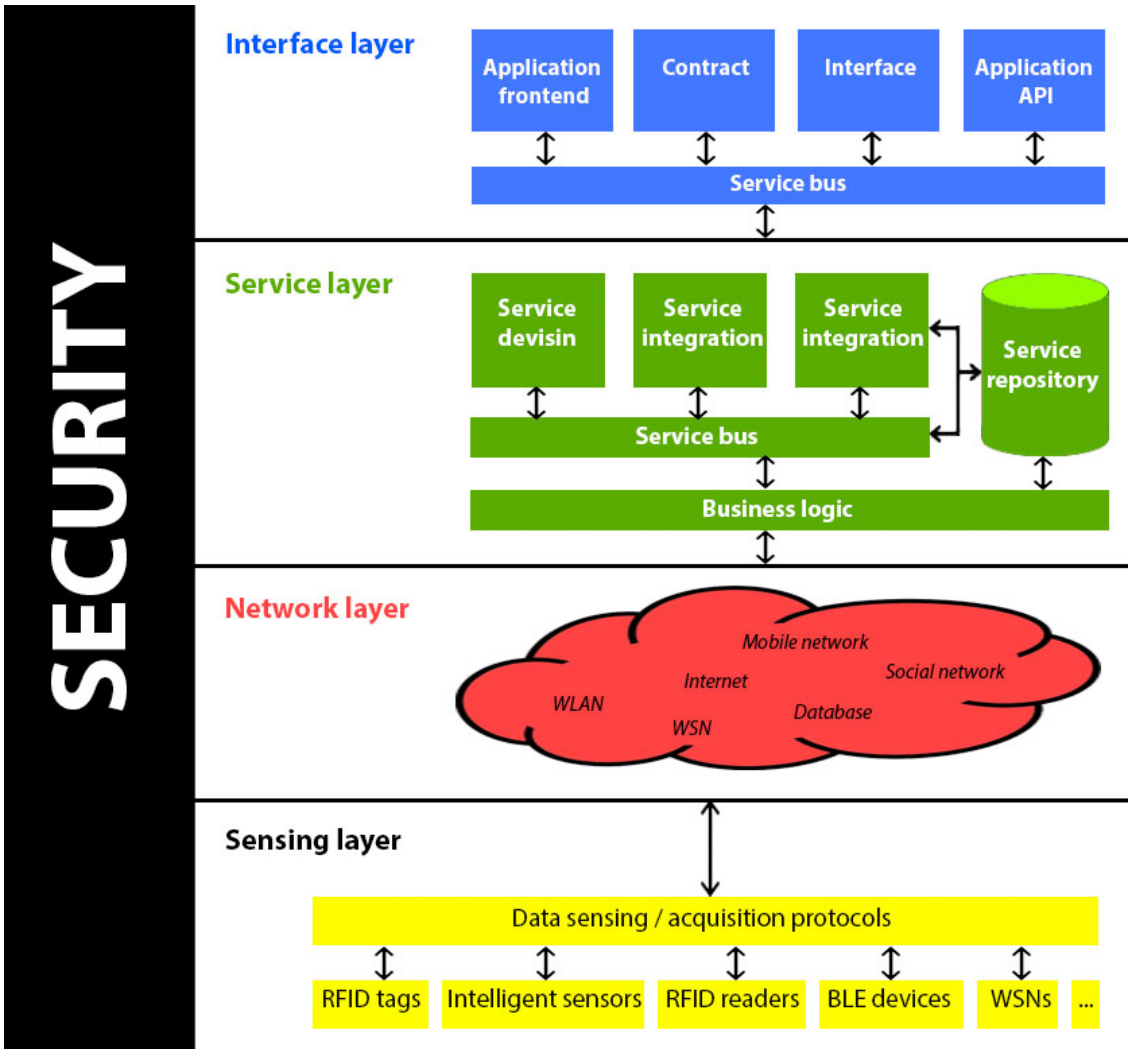

Figure 2.1. SOA for IoT

this process according to the requirements of users-applications. It should be mention that in a dynamically changing network an important thing is to discover and map objects automatically. To establish, manage, and plan behaviors, things must be automatically assigned to roles and be able to change to any other role whenever necessary. These capabilities enable devices to perform tasks together. In order to design the IoT network layer, designers should consider technologies of management for heterogeneous, requirements of QoS, network energy efficiency, data and signal processing, service discovery and retrieval, privacy and security [4].

2.2.3 Service Layer

Service layer prepare operations to integrate services and applications in IoT. It should be mention that it relies on middle-ware technologies. Prepare IoT platform by middle-ware technology is cost-efficient in which, it reuses software and hardware platforms. In the service layer, service specification for middle-ware is the main activity. And also, all service-oriented problems (such as data management, information exchange and storage and search engines) will process by this layer. This layer, can identify usual requirements of applications and provides protocols to support applications, required services and user needs [4]. Following components are included by this layer:

1. Service discovery: Find objects that in an efficient way can provide the required services and information.

2. Service composition: Interact and communicate between connected objects. This is to plan more appropriate services for achieve reliable services. In discovery phase, relationships between different objects will uses to find desired services.

3. Trustworthiness management: determine mechanisms of trust and reputation as aims. In order to create reliable system, it can use and evaluate information which prepare by other services.

4. Service APIs: It supports interactions among IoT services.

2.2.4 Interface Layer

Similar to other networks, in IoT, many devices which are made by different manufacturer are involved, and they have not always same standards. So, there are too many interaction issues with communication between Things and information exchange. Moreover, increasing the number of objects which join to IoT, makes dynamically connect and disconnect, operate and communicate more difficult. Interface layer is necessary in order to management and interconnection of thing simply. As a subset of services standards, IPF (interface profile) supports interaction with network applications. As an example, a IFP that related to implementing Universal Plug and Play (UPnP), defines a protocol for making things interaction simpler. IFPs are used in order to explain characteristics among services and applications. In order to find new services for an application, services on the service layer run on limited network structure as they connect to network. For interact effectively between

services and applications a SOCRADES integration architecture (SIA) has been suggested. In the past, service layer for application provides universal API, but, in the new result of SOA-IoT research, SPP (service provisioning process) can provide interaction effectively among services and applications [5].

2.3 Five-layer architecture

We present a brief study of these five layers.

2.3.1 Objects Layer (Perception layer)

This layer represents physical sensors of IoT that collect and process information. Objects layer includes sensors and actuators. To configure heterogeneous objects, it is necessary for the perception layer to use standard plug-and-play mechanisms. The perception layer digitizes the data and through secure channels transmits them to the object abstraction layer [5].

2.3.2 Object Abstraction Layer

This layer transfers generated data by the objects (perception) layer through secure channels to the service management layer. Data transfer can be done through various technologies such as RFID, 3G, GSM, UMTS, infrared, ZigBee, Wi-Fi, Bluetooth Low Energy, etc [5].

2.3.3 Service Management Layer

Service Management also called as Pairing layer or Middleware layer. This layer based on addresses and names, pairs a service with its requester. It processes data which are received, makes decisions, and delivers the required services over the network protocols. It also enables IoT programmers to work with heterogeneous objects regardless of a particular hardware platform. [5].

2.3.4 Application Layer

The services which are requested by customers have been provided by this layer. In order to meet customers‘ need, this layer has the ability to provide the high quality smart services. Many virtual market have been covered

by this layer such as smart building, smart home, industrial automation, transportation and smart healthcare [5].

2.3.5 Business Layer

It also called management layer. This layer manages the overall services and system activities of IoT. based on the received data from the Application layer, business layer builds business models, charts, graph charts and etc. The Business Layer (based on Big Data analysis) makes it possible to support decision-making processes [5].

2.4 IoT elements

As illustrated in figure 2.2, six key elements needed to deliver IoT performance:

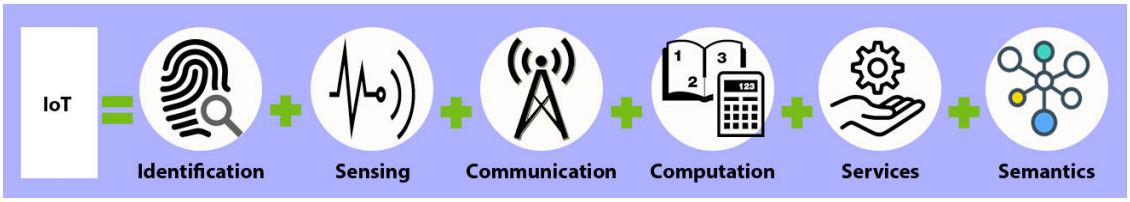

Figure 2.2. Elements of IoT

2.4.1 Identification

In order to call and match services with their request, identification is vital for the IoT. There are too many methods for identification in IoT. e.g. electronic product codes (EPC) and ubiquitous codes (uCode). Moreover, addressing is important in order to distinction between ID of object and its address. Address of object, refers to its address through communication network. While, object ID refers to the name of the object (e.g. "H1" for a specific humidity sensor). Furthermore, addressing objects contain IPv4 and IPv6. Between object ID and address is important to individuate, because identification in not unique globally. So, in this step, it needs to be assist by addressing. We should consider that, within a network, objects probably do not use private IPs, and use public one [5].

2.4.2 Sensing

Collecting data from dependent Things through a network by sensors (such as smart sensors, actuators or wearable sensing devices) and send to a database, or cloud is IoT sensing. These data will be evaluated in order to take a particular operation. These operations are based on required services [5].

2.4.3 Communication

In order to deliver particular smart services, heterogeneous objects connect together by IoT communication technologies. Usually, operating IoT nodes are in persistence of noisy and lossy communication links and using low power energy. There are some communication protocols which are used for IoT such as Bluetooth, Wi-Fi, LTE-Advance, Z-Wave, IEEE 802.15.4. Also, some other particular are in use as ultra-wide bandwidth (UWB), Near Field Communication (NFC) and RFID [5].

2.4.4 Computation

Brain of IoT is consist of processing unit (e.g. microprocessors and microcontrollers) and software applications. Various software platforms are employed in order to provide IoT capabilities as operating systems. Moreover, many hardware platforms are available in order to run applications of IoT such as Raspberry PI, Arduino, FriendlyARM and UDOO [5].

2.4.5 Services

We address four classes of IoT services as follow [5]:

1. Identity-related services: An application that want to bring an objects of of real world in to the virtual world, should identify those objects. In comparison with other types of services, these are the most important and also basic ones.

2. Information Aggregation Services: Data which need to be processed will collect and summarize and then report to the IoT application.

3. Collaborative-Aware Services: They act on top of Information Aggregation Services and use the data to make immediate decisions and reactions.

4. Ubiquitous Services: Aim to provide Collaborative-Aware Services whenever and wherever that are needed by anyone.

2.4.6 Semantics

Semantic is a brain for IoT which sends requests to the right resource. In order to provide services which are required, we need data which are collected by different machine. In IoT, ability of extracting smartly these data referred by semantics. Main steps of Extracting knowledge are as follow [5]:

- Discovering and using resources
- modeling information
- recognizing and analyzing data for making right decision in order to provide the exact requested service

Chapter 3

IoT challenges

3.1 Main challenges:

A lot of challenges such as availability, reliability, mobility, performance, scalability, interoperability, security, management, and trust should be consider [5].

3.1.1 Availability

Services for customers must be provide anytime and anywhere in both hardware and software. Software availability is capability of IoT applications in order to offer services to everyone in different locations altogether. Hardware availability is about the devices which are compatible with IoT features and protocols and also should always be exist [5].

3.1.2 Reliability

Based on specification, system should work properly. In comparison with availability, it is more important and It has stricter requirements for emergency response programs. In these systems, communication is an important part of the network and must resist failures to achieve reliable information distribution. Reliability should implement in all the layers of IoT in both hardware and software [5].

3.1.3 Mobility

Most of the services are expected to be delivered to mobile users, so, in the implementation of IoT, mobility is a challenge. When mobile devices change their gateway, a service interruption can happens. It is important that users connect continuously to their requested services while they are moving. Caching and tunneling allow applications to access IoT data if they cannot access to resources temporary [5].

3.1.4 Performance

IoT devices, in order to provide performance in the best level should be monitored and also evaluated. Of course, affordable price is important. From the performance point of view, we can estimate speed of processing and communication, cost and device form factor [5].

3.1.5 Management

Connection of a huge amount of smart devices need to be managed by service providers. In order to handle this management, new light-weight management protocols should be developed. An effective factor is managing IoT devices and applications. An example of a managing platform for IoT which makes it simpler anywhere and real-time is MASH [5].

3.1.6 Scalability

It refers to a capability of adding new devices, services and also functions for users without any bad effect in the quality of other services. It is not an easy task due to existence of various hardware and protocols of communications. A general architecture of IoT has three layers of 1) virtual object, 2) composite virtual object, 3) service layer. A guarantee for scalability and interoperability in IoT is presenting these 3 layers with intelligence, automation, and zero configuration (A way to connect two or more devices that can be connected to a network without the need for network configuration) in each object [5].

3.1.7 Interoperability

Manufacturers of device and developers of application should consider interoperability. The main reason is to ensure delivery of services to all of

customers without considering the hardware platform which they use. As an example, in order to guarantee the interoperability, common communication network such as Wi-Fi, GSM and NFC have been supported by most of smartphones [5].

3.1.8 Security and Privacy

In the implementing IoT, security is an important challenge. Actually, still there is no common architecture and also standard for IoT security. For heterogeneous network is also hard to guarantee security and user privacy. Key distribution among devices is an open security issue in the standards which should be consider. Moreover, operation of profile access between devices of IoT without any interferences are important and critical [5].

3.2 Technical challenges

3.2.1 Designing SOA

Designing SOA for the IoT is a major challenge in which service-based cases may suffer from performance and cost constraints. In addition, scalability issues often arise as more physical objects are connected to the network. When the number of things is large, scalability at various levels such as data transfer and networking, data processing and management, and service delivery is problematic [4].

3.2.2 Networking

In terms of networking, the IoT is a very complex heterogeneous network that involves connection of different types of networks through different communication technologies. Currently, there is no commonly accepted common system that hides heterogeneity of underlining networks/communication technologies and provides a transparent naming service for different applications. Large amounts of simultaneous data transmission on the network can also cause frequent delays, conflicts and communication problems. Developing network technologies and standards that can allow data (which collected by a large number of devices) to move effectively on IoT networks is a challenging task. Connected management is a research challenge in terms of facilitating cooperation between different institutions[4].

3.2.3 Services

In terms of services, the lack of a commonly accepted service description language, makes it difficult to develop services and integrate resources of physical objects into value-added services. Developed services may not be compatible with different communication and implementation environments. In addition, to spread IoT technology, powerful service discovery methods and services of object naming must be developed [4].

3.2.4 Information

Because the IoT is often developed based on a traditional ICT environment and is influenced by everything connected to the network, it is used to integrate the IoT with existing IT systems or older systems in an integrated information infrastructure which needs a lot work. In addition, with a large number of things connected to the Internet, a large amount of real-time data flow will be automatically generated by connected things. Data may not be valuable unless individuals find an effective way to analyze and understand it. Analyzing or extracting large amounts of data generated from both existing IoT applications and existing IT systems requires large data analytic skills to obtain valuable information, which can be a challenge for many end users [4].

3.3 Research trends

3.3.1 Big data analytics in support of the IoT

In order to analyze big data, some platforms like Apache Hadoop and SciDB can be useful. But, due to the huge amount of data which have to be processed, these tools are not strong enough to satisfy the needs of big data in IoT. Also, these platforms need to work real-time in order to support IoT. A common platform of big data analytic which is necessary for IoT, can be delivered to IoT applications as a service. The overhead of overall IoT ecosystem, should not be increase dramatically by these analytic services [5].

3.3.2 Cloud computing for the IoT cloud computing (CC)

It offers a mechanism of management for big data which processing data will enable and intended knowledge will extract from it. It has some challenge as follow [5]:

1. Synchronization: In order to provide services in real-time, there is a challenge of synchronization between different cloud vendors. It is considerable that services are built on different platforms of cloud.

2. Standardization: Again, because of different vendors which IoT cloud-based services have to inter-operate with them, standardization is a big challenge.

3. Balancing: Making a balance between IoT requirements and environments of general cloud service, another challenge exist that is because of differences in infrastructure.

4. Reliability: This challenge is because of existence of difference in mechanism of security between cloud platform and IoT devices.

5. Management: Both of IoT systems and CC have different components and resources, So, managing these two is another challenge.

6. Enhancement: cloud-based IoT services should be validate in order to providing good services. Many cloud platforms which have different abilities can be used by IoT.

In the table 3.1 you can see specifications of some IoT cloud platforms:
fog computing has the potential to offer services that deliver better delay performance

3.3.3 Fog computing in support of the IoT

Between large scale cloud computing, storage services and smart devices, fog computing is like a bridge. Extending services of cloud computing in to the edge devices is a good possibility of fog computing. A good potential of fog computing is to offer services that deliver better delay performance. This is due to their adjacency to end-users. Typically, between fog and cloud a considerable difference in scale. In comparison to fog,massive computational,

Platform	Gateway	Provision	Assurance	Billing	Application protocols			
					REST	COAP	XMPP	MQTT
Arkessa	-	+	+	-	+	-	-	+
Axeda	+	+	+	+	+	-	-	-
Etherios	+	+	+	-	+	-	-	-
LittleBits	-	-	-	-	+	-	-	-
NanoService	+	+	+	-	+	+	-	-
Nimbits	-	-	-	-	+	-	+	-
Ninja Blocks	+	-	-	-	+	-	-	-
OnePlatform	+	+	+	-	+	+	+	-
RealTime.io	+	+	-	-	+	-	-	-
SensorCloud	+	+	-	-	+	-	-	-
SmartThings	+	+	-	-	+	-	-	-
TempoDB	-	-	-	-	+	-	-	-
Thingworx	-	+	+	-	+	-	-	+
Xively	+	+	+	+	+	-	-	+

Table 3.1. IoT cloud platforms and their characteristics [5]

storage and communications abilities. Possible providers of fog computing are mobile operators which can offer services of fog such as SaaS, PaaS or IaaS. Fig. 3.1 shows the relations between cloud, fog and end users [5].

Following features can show that for the IoT designer, fog computing is an optimal choice [5]:

1. Location: resources of fog are located between cloud data centers and smart objects. So, better delay can provide.

2. Distribution: Fog computing is based on "micro" centers which compared to the cloud has limited storage, processing and communication abilities. So, it is possible to deploy many such micro centers closer to the end-users as their cost is usually negligible compared to cloud data centers.

3. Scalability: Fog allows IoT systems to be more scalable. So, if the number of end-user increase, in order to cope with the increasing load,

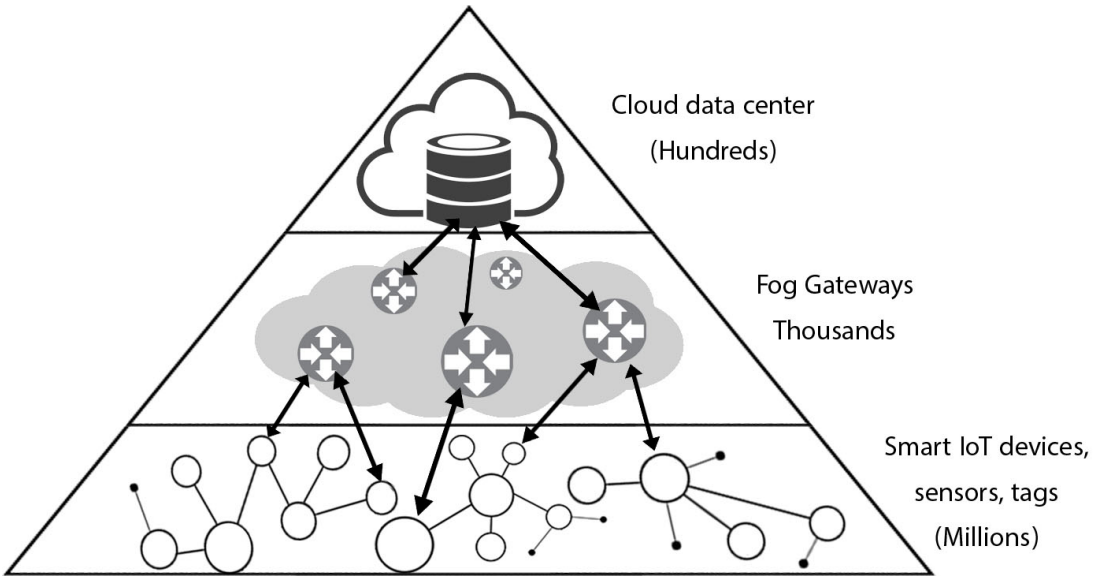

Figure 3.1. Task of cloud and fog resources in order to provide services of IoT to users

number of deployed micro fog centers will increase as well. Deploying of new data center in cloud needs much cost, so it cannot achieve by cloud.

4. Density of devices: Fog helps to provide services which are resilient and replicated.

5. Mobility support: Resources of fog, work like cloud of mobile which is positioned close to the end users.

6. Real-time: Real time interactive services performance in fog is provided better.

7. Standardization: Resources of fog can interact with various providers of cloud.

8. On the fly analysis: Data aggregation can perform by fog resources in order to send processed data partially, instead of sending raw data to data centers of cloud for farther processing.

So, overall performance of IoT applications can increase by the potential of fog, because in local resources, it tries to do some of the high-level services provided by the cloud.

Chapter 4

Security in IoT

4.1 Security features of IoT

In this section, IoT security features, security and privacy issues, and possible solutions are discussed.

4.1.1 Confidentiality

It can certify that availability of data is just for users who are authorized through the process and cannot be interfered and overhear by non-authorized users. Because of integrity in a large number of measurement devices, confidentiality is an important principle of security. It is important that the security information does not reveal by the data which are collected by measurement devices [3].

4.1.2 Integrity

It can ensure that during the data delivery in communication networks, data cannot be tampered by intended or unintended interference, and finally providing the accurate data for authorized users. If IoT applications receive fake or tampered data, incorrect performance status can be estimated and erroneous feedback commands can be issued, which could further disrupt the operation of IoT applications [3].

4.1.3 Availability

Availability can ensure that for authorized users, devices and the data are available. In the IoT, services are usually requested in real time. If the requested data cannot be delivered in a timely manner, services cannot be scheduled and provided [3].

4.1.4 Identification and authentication

Identification can ensure that devices or applications which are not authorized cannot connect to the IoT. Authentication can ensure the legality of data provided over the network, as well as ensure that the devices or applications requesting the data are legitimate. Since, so many different objects make up an IoT network, it is difficult to identify and authenticate all data and objects. Therefore, it is crucial to designing efficient mechanisms in order to deal with the authentication of objects or things [3].

4.1.5 Privacy

Privacy can ensure that only the corresponding user can control data, and it cannot be accessed or processed by other users. It also ensures that the user can only have certain controls based on the received data and cannot derive any other valuable information from the received data, but aim of confidentiality is to encrypt the data without being eavesdropped and interfered by non-authorized users. Because of large number of devices, services, and people which sharing the same communication network, privacy is considered as one of important security principles [3].

4.1.6 Trust

Trust can ensure to achieve the above security and privacy objectives during the interactions among different applications, different IoT layers and also different objects. Goals of trust is divisible as trust between devices, trust between devices and applications and trust between each IoT layer. In order to implement these trust objectives in IoT, trust management systems should be designed [3].

Chapter 5

IoT networks

Internet of things is going to be real by the implementation of networks like SigFox or LoRa. By these technologies, any object can connect to the internet by using unlicensed bands. Meantime, 3GPP introduced Extended Coverage GSM (EC-GSM), Narrow band IoT (NB-IoT) and LTE - Machine communication type (LTE-M) that could use 2G and 4G network infrastructure and licensed bands.
There are so many IoT applications, but still we don't have a technology which can handle all of them [10].

5.1 Sigfox

A new category of wireless technology which is appropriate for IoT application is Low-Power Wide Area Networks (LPWANs) in order to support low energy consumption. SigFox is one of the most popular LPWAN technologies which is developed by SigFox company in 2009.
In the Sigfox network, about 1 billion IoT objects can communicate bidirectional in more than 50 countries. supporting IPv6 over Sigfox is under the developing by IETF LPWAN working group [9].

5.2 LoRa

A french company called Cycleo developed the LoRa technology, then Semtech which now are selling the Lora chips, acquired it. Lora modulation scheme is CSS (Chirp Spreading Spectrum). In order to decrease energy consumption,

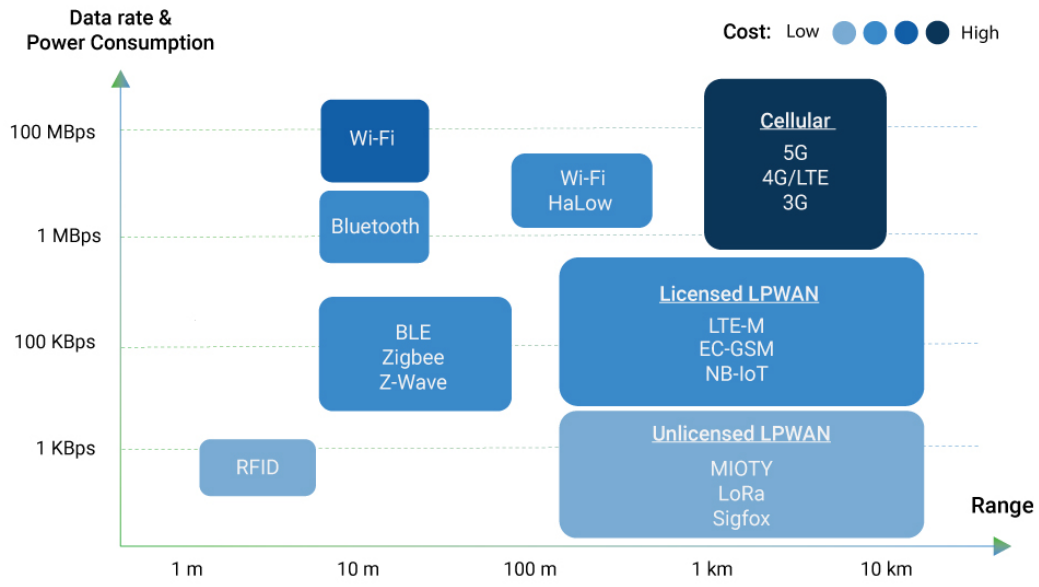

Figure 5.1. IoT Technologies Comparison [8]

improve range and control bit rate, different Spreading Factors (SF) are defined, which Lora uses it.
Operation of LoRa is over 169, 433, 868 and 915 MHz in Industrial, Scientific and Medical (ISM) bands. Depending on the used sub-band, and in order to limit interferences, regulatory authorities specified a duty cycle ranging from 0.1 to 1. Management of Lora network is open and implementing Lora networks and stations is possible for everyone in order to offer services, provided that it respects the rules of spectrum use [10].

5.3 Zigbee

In order to enable low-power, low-data-rate M2M in low cost IoT communications, Zigbee alliance developed a technology called Zigbee. Range of transmitting in Zigbee devices (depending on output power and radio frequency environment) are expect to 10 - 75 meters. Operation of Zigbee is in 2.4GHz, 915MHz and 868MHz ISM frequency bands. Network topologies which are supported by Zigbee are Mesh, Star and Cluster Tree [11].

5.4 NB-IoT

NB-IoT is a radio access technology which in some aspects reuses its previous technical components. As the name implies, whole of the system operates in narrow spectrum from 200 kHz. Due to the minimum need of frequency spectrum, it provides tremendous flexibility in deployment compare to legacy LTE.
Considering the narrow spectrum deployment, in order to support a combination of high coverage and high capacity in up-link, 200 kHz bandwidth is divided into channels with 3.75 kHz or 15 kHz spacing [12].

5.5 LTE-M

LTE-M as a potential technology to make LTE efficient for IoT has been introduced by many companies and operators. Initially, LTE was not yet optimized as a platform to support low data-rate required for devices such as sensors, in M2M communication.
However, the first project to present LTE-M which supports future wireless systems was EXALTED, which was in terms of better security, lower cost, support large number of users and efficient in energy consumption. In the proposed architecture, LTE-M has been integrated in the network of LTE cellular. So, LTE mobile phones and devices of LTE-M communicate through the same LTE/LTE-M network [13].

5.6 IEEE 802.11ah

Initial design of Wi-Fi was providing access of broadband wireless Internet for few devices. Due to the use of transmission bands of sub 1GHz (S1G) and narrower channels, new amendment to the Wi-Fi standard IEEE 802.11ah can handle up to 800 stations and also increases access point coverage up to $1km^2$. To date, for a large number of IoT devices located in a large area the only version of Wi-Fi that supports low-energy communications is 802.11ah [14].

Chapter 6

IoT in healthcare

6.1 Introduction

One of the most absorbing application areas of IoT is in the field of medical-care, which has a great potential to use lots of medical applications such as remote health monitoring, chronic diseases, and elderly care. In fact, Compliance with treatment and medication at home and by healthcare providers is the main potential of this application. Therefore, core in the IoT medical care is consist of various medical devices, sensors, and diagnostic and imaging devices [1].

Common goals of these parts, are increasing the quality of user‘s life and also reducing the cost of treatment. Also, from the perspective of healthcare providers, through the remote servicing, downtime of devices can be reduce. Healthcare IoT network which get up-to-date by technologies of wireless, are expected to support real time monitoring, early diagnosis, chronic illnesses, and medical emergencies. Also, medical servers and databases and even gateways, have important roles in creating health records and on-demand health service delivery. To have these services, stakeholders must be authorized. It should be noted that based on the wireless sensor network (WSN) in the field of healthcare services, R&D activities (Research and development) can be considered as initial IoT-based healthcare research. The progressive trend is to implementing IP-based networks using IPv6-based low-power wireless personal area network (6LoWPAN) [1].

6.2 IoT healthcare networks

IoT healthcare network or "IoThNet" supports IoT backbone access, and simplify medical data transmission and reception [1]. Also, makes it possible to use appropriate communication with health care.

6.2.1 The IoThNet topology

The IoThNet topology refers to the regulation of various IoT health elements. A heterogeneous computational network collects large amounts of vital signs and sensor data such as body temperature and blood pressure and forms a typical IoThNet topology. In the fig. 6.1 we can see patient's health profile. All signs are captured by medical devices and sensors which are attached to the patient's body. All data which are captured will analyzed and stored in order to monitor patients [1].

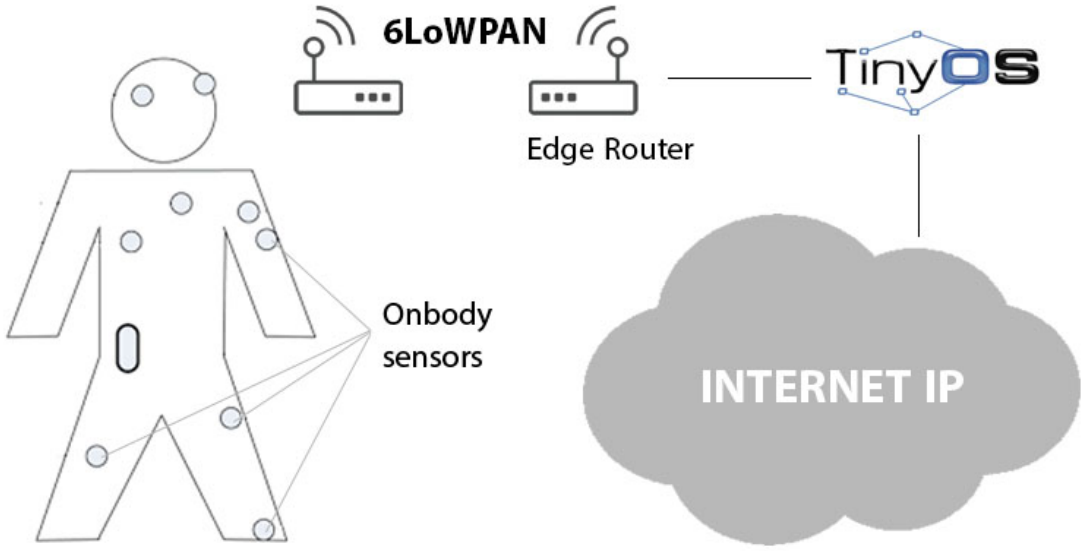

Figure 6.1. wearables and personalized health care sensors and remote monitoring

In the fig. 6.2, there is topology of IoThNet in which role of gateway is showing. iMedPack or intelligent pharmaceutical packing is an IoT device which by this, all problems of misusing medicine will manage. So, this ensures

medicine compliance. iMedBox or intelligent medicine box, is a gateway with several sensors and interfaces. IoT devices and different wearable sensors are connected to gateways wirelessly. These are connecting environment of patient to the IoT healthcare cloud and to heterogeneous network or HetNet. By HetNet, clinical recognition and other analyses will enable. And also gateways can consider, store, and shows data which are collected [1].

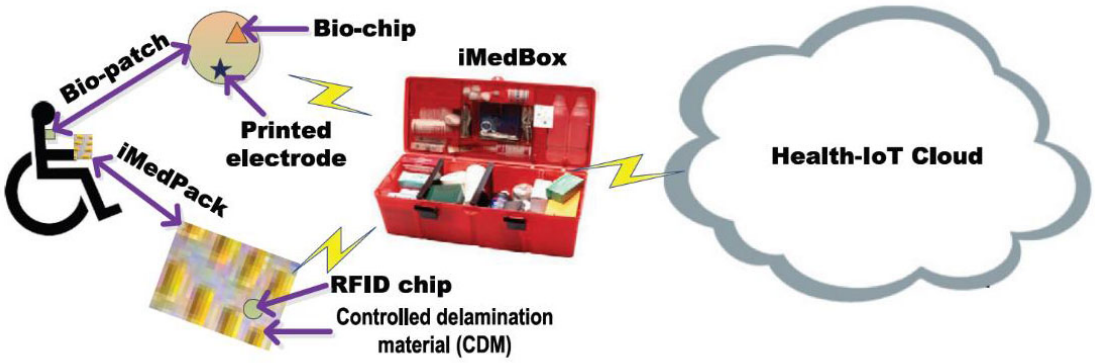

Figure 6.2. An IoThNet topology with an intelligent healthcare gateway [1].

6.2.2 The IoThNet architecture

Architecture of IoThNet addresses to physical elements of IoThNet, their functional structure, techniques and principles of working. The basis of IoThNet is 6LoWPAN. Structure of layers in the 6LoWPAN can be seen in the fig 6.3.
[1]. Medical devices are like vehicular networks which capture and examine health data through IPv6 application servers. The protocol of lightweight auto configuration has introduced for V2I (vehicle-to-infrastructure) communications in IoThNet as illustrated in fig. 6.4. In the routing table, IPv6 route has been used in this protocol as a default route.

6.2.3 The IoThNet platform

The platform of IoThNet addresses to both the network platform model and platform of computing. In the fig.6.5 illustrated a framework of service

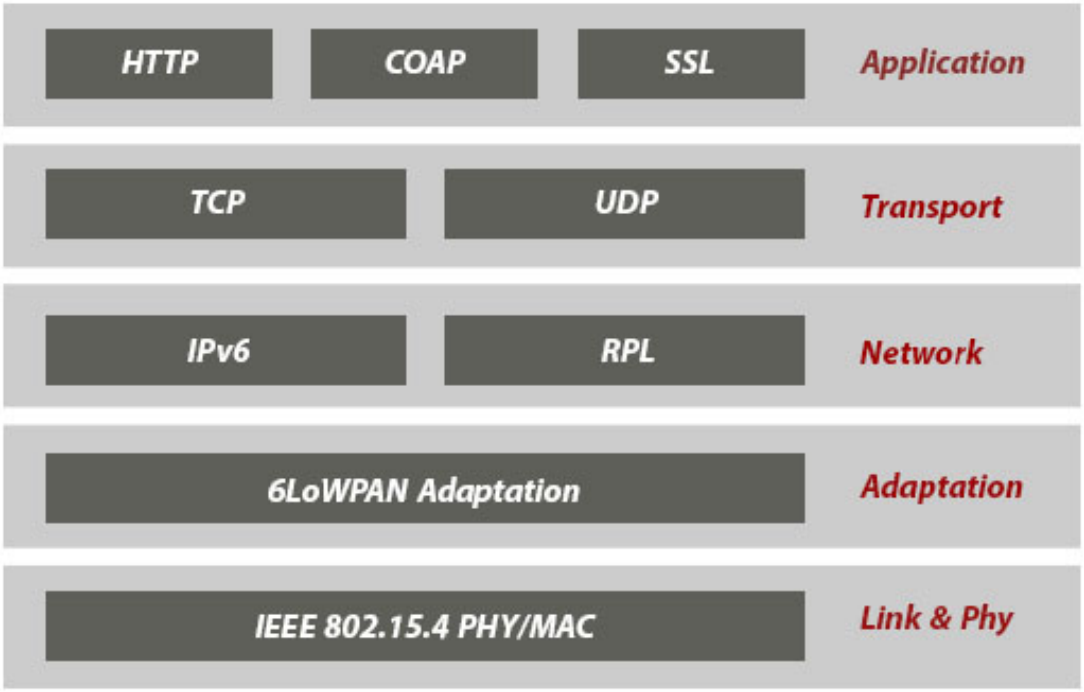

Figure 6.3. 6LoWPAN protocol stack

platform which is focusing in health information of residents. This framework shows a regular hierarchical model of how caregivers or agents can access different databases from the application layer with the help of a support layer [1].

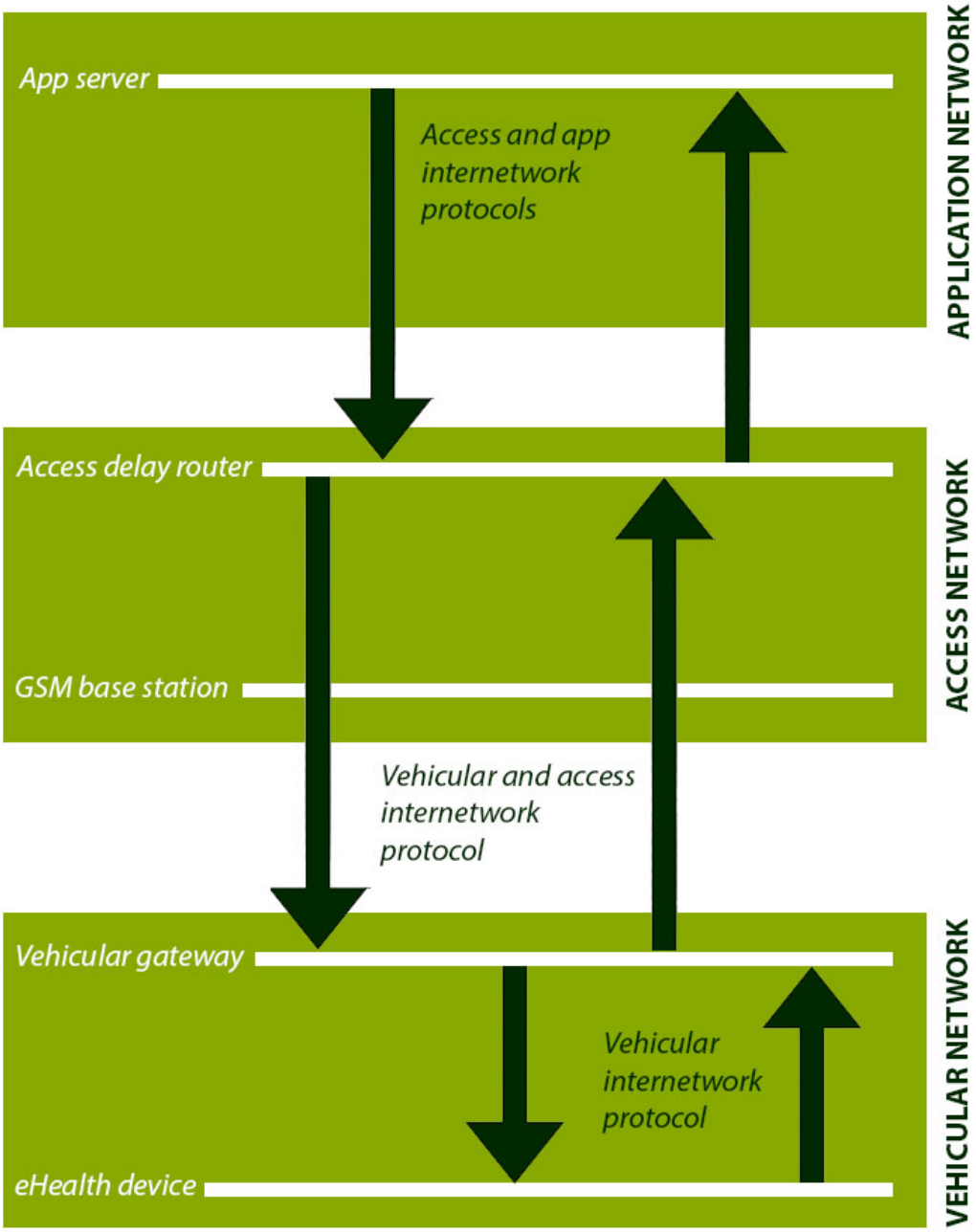

Figure 6.4. V2I communications in the IoThNet

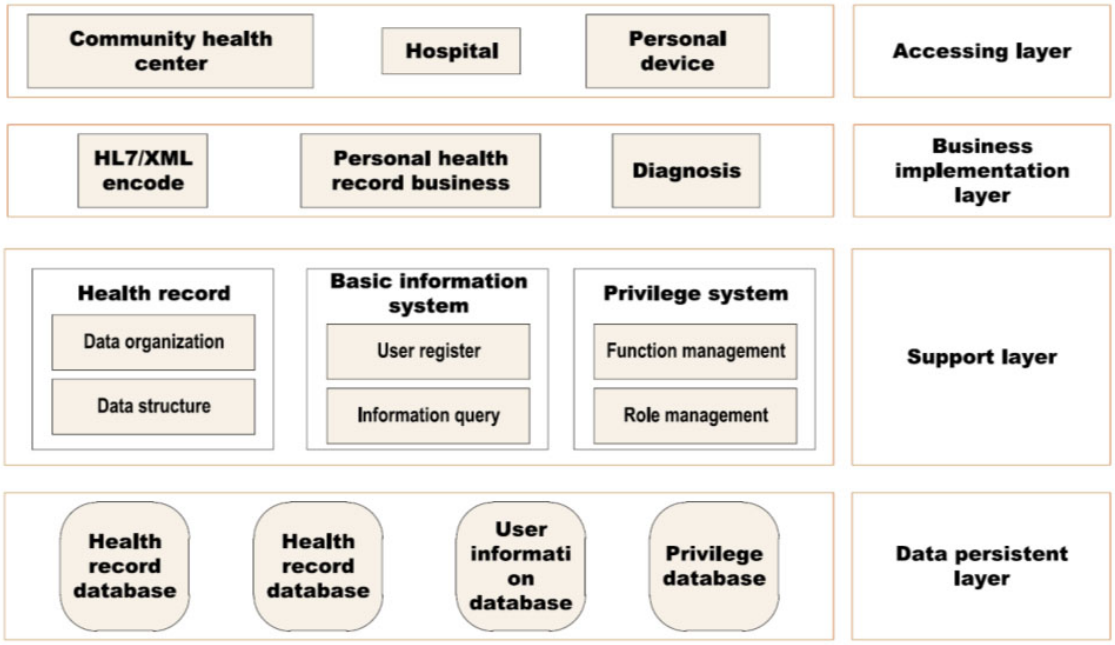

Figure 6.5. A health information service model functional framework [1].

Chapter 7

IOT in industries and environment

7.1 Key IoT applications in industries

IoT usage is growing rapidly. But still applications of IoT are in its primary stage. Now, there are a few applications of IoT such as transportation, food supply chain (FSC), inventory and production management, security, and etc, which are developed and deployed in different industries. We address some of the IoT applications in industry [4].

7.1.1 Using IoT in FSC

The FSC has many stakeholders, large temporal and geographical scale, and complexity in performance processes. This complexity in the management of quality, performance and safety of public food has created many problems. Some potentials have been offered by IoT in order to address challenge of traceability, visibility, and controllability. For example, as a typical one for FSC, Food-IoT has three parts as follow [4]:

- Field devices as user interface terminals, RFID readers/tags, WSN nodes.
- Backbone systems as servers and databases.
- Communication of infrastructures as cellular, WLAN, satellite, Ethernet, etc.

7.1.2 Using IoT for safer mining production

For many countries, working in the underground mines is a big concern. Using IoT technology in order to sense disasters can prevent and reduce mining accidents. Using biological and chemical sensors which obtain biological information from human organs in order to earlier detecting disease of miners and also to recognize harmful gases and hazardous dust [4].

7.1.3 Using IoT in transportation and logistics

One of the main role of IoT is in logistics and transportation. Objects which equipped with RFID tags, sensors or bar-codes, can be monitor real-time across manufacturing, shipping, distribution, and so on. Moreover, in the transportation system, IoT technologies can be used to increase capabilities of sensing, networking, communication, and data processing in vehicles. As an example, BMW developed an intelligent informatics system calling iDrive which has various sensors in order to monitor the environment as road condition. We expect to see autopilot for driving automatically and detect pedestrians and other cars in order to avoid collisions [4].

7.1.4 Using IoT in firefighting

Safety is another field which IoT is used. It used in order to earlier warning for fire incident. It works by applying RFID tags, mobile RFID readers, smart camcorders, wireless communications networks and sensor networks. By using automatic detection, firefighting authority can make to achieve early warning fire and real-time environmental monitoring [4].

7.2 Smart city

7.2.1 Concept and services

Smart city market comes from a link between industry and sectors of servicing such as Smart Buildings, Smart Mobility and Smart Environment. For a number of technical, political and financial problems, the Smart City market has not really flourished yet. Under the political dimension, attributing decision-making power to different stakeholders is the main obstacle. One possible way to break this barrier is to institutionalize the entire decision-making and implementation, focus strategic planning, and manage aspects

of the smart city in a dedicated part of the city. Technically, the most important issue is the lack of cooperation of heterogeneous technologies that are currently used in the development of cities and towns. In this sense, the IoT landscape can be a building block for the realization of a single urban-scale ICT platform. In terms of finances, a clear business model is still lacking, although initiatives have recently been taken to fill this gap.
In the following, we review some of the services that may be enabled by the IoT model and have potential in the smart city context, as they can enhance the win-win situation of quality and service [6].

7.2.2 Technologies

A smart city composed of information amassing, processing and sending technologies inspired by the invention of a tool to improve the quality of life. A smart city covers a set of entities, such as health, transportation, education, entertainment and food.

Traffic Management

The main goal of a smart city is to provide the most advanced transportation and traffic services known through the intelligent traffic management system. Smart traffic management aims to provide traffic information to the residents of a smart city to be more aware of the routes which can lead to smart and harmless choices. To better manage the transportation system, various skills such as vehicle navigation, traffic signal adjustment and motion circuits can be used, leading to a combination of real-time data that can be used for safe driving and parking guidance. With the help of moving vehicles or measuring tools, smart traffic management obtains transportation information. Vehicle movement information is obtained through several methods such as smartphone monitoring and Global Positioning System (GPS).
A system based on metering devices is the second method for obtaining traffic information. Measuring instruments consist of induction loop detectors, audio detectors, video detectors or any other sensor. In the inductive ring detection method, the car is detected when it passes through the magnetic zone of the ring. Sound detection estimates traffic congestion through audio signals, which are generated by passing vehicles on the road. The film is detected by cameras mounted on beams near highways.
Currently, traffic data obtained through video trackers and induction loops cannot be retrieved, advertised and evaluated because they have not been

fully developed and standardized. Therefore, standardizing this data requires an intelligent traffic management system. So that, it can then be analyzed and distributed for a secure driving [7].

Emergency Systems

One of the goals of a smart city is to provide security and protect citizens. To achieve this goal, an intelligent emergency system is needed which can be used to enforce the law, detect crime, and manage natural disasters. Various tools such as traffic sensors and CCTV cameras have been developed to aggregate data. This data, together with analytical evaluation, make the possibility of increasing the quality of data used by hospitals, fire departments and police departments. The focus of law enforcement agencies shifts from identifying individual perpetrators to organizing units based on hazard classes. In general, extensive surveillance in a smart city brings benefits in terms of security and well-being. These benefits are obtained through the collection of information through measuring devices and CCTV cameras. However, constant monitoring raises privacy concerns due to the evaluation of information collected by a particular company. A combination of data exchange, big data analysis and measurement tools is essential for institutionalizing an intelligent emergency system. In the event of any natural disaster or accident, for the life of the masses, it is necessary to combine accurate data and quickly implement this data in an accurate manner. Therefore, creating an intelligent emergency system for the exchange of information between different organizations is very important, so that, it allows quick response and issuance of operational instructions [7].

Energy consumption in the city

The Internet of Things may, in conjunction with the Air Quality Monitoring Service, provide services to monitor the entire country energy consumption. Thus, enabling officials and citizens to have a clear and accurate view of the amount of energy required by various services (such as public lighting). It will be possible to identify the main sources of energy consumption and set priorities to optimize their behavior. This is in line with European guidelines for improving energy efficiency in the coming years. To achieve such services, traction monitors must be integrated with the city network. In addition, it is possible to increase these services with active capabilities to control local power generation structures [7].

Smart parking

Smart parking service is based on road sensors and smart displays that guide drivers to the best route for parking in the city. The benefits of this service are manifold: faster time to find a parking space means less CO emissions from cars, less traffic and happier citizens. In addition, short-range communication technologies, such as radio frequency identification (RFID) or near field communication (NFC), can be used as an electronic parking permit process system in a dedicated slot for residents or people with disabilities. So, a service to citizens who can legitimately use these slots. The smart parking service can be integrated directly into the IoT infrastructure. [7].

Chapter 8

Simulation basis

8.1 Simulation procedure

At the beginning, we generate a random binary sequence consisting of '-1's' and '+1's' which are input to the transmitter. The transmitter modulates the bit vector and then sends it through the channel (AWGN channel). Then some noise will be added. This signal, which is now noisy, is considered as input to the receiver. In this step, demodulation should be done and in the last step, we should compare the sent bits and the received bits.

8.1.1 AWGN channel

In the industrial IoT, industrial objects such as sensors and robots must communicate strongly with each other through wireless connections. These wireless links affect from different channel disorders such as fading, random fluctuation, attenuations and (NLoS) due to existence of obstacles. n order to model the industrial environment, there are several channel models. One of the available wireless channel models that can be used in industrial IoT is the AWGN channel model, which we are focusing on with Lora technology [22].
From many natural sources such as the sun and the vibration of conductive atoms, Gaussian white noise can be obtained. When a signal passes through the AWGN channel, Gaussian white noise is added to that signal. The frequency response of the amplitude in this channel is flat and for all frequencies the phase response is linear. There is no amplitude loss or phase distortion for the modulated signals passing through this channel, so there is no fading for this case and only AWGN distortion exists [19]. The received signal is

simplified to:

$$r(t) = x(t) + n(t)$$

Where n(t) is noise, and x(t) is transmitted signal.

Bit Error Rate

The bit error rate (BER) is defined as the percentage of bits that have an error relative to the total number of bits received in a transmission, that is:

$$\text{BER} = \text{Number of bits with error / Total number of bits sent}$$

High BER increases packet loss and latency as well as decreases throughput. Bit error rate is a key parameter used in evaluating system performance. There is a possibility of error when data is transmitting through the link [19].

Signal to Noise Ratio (SNR)

The ratio of the received signal strength to the SNR noise resistance is called one of the important parameters in the physical layer of the wireless LAN (LAWN). As mentioned earlier, the power of noise can interfere with other signals as well as environmental noise. In a multi-channel environment, it is difficult to determine the exact relationship between BER and SNR [21]. The signal-to-noise ratio (SNR) is denoted by the following equation, measured in decibels.:

$$\text{SNR} = 10 \log_{10}(\text{Signal Power / Noise Power}) \text{ dB}$$

Energy per bit to noise power spectral density ratio (E_b/N_0)

In the data transmission and digital communication, one of the important parameter is E_b/N_0, known as the "SNR per bit" which is normalized SNR measure.In different modulation schemes, E_b/N_0 is very useful for comparing BER performance regardless of bandwidth.
In the concept of forward error correction (FEC) using E_b/N_0 usually refer to the energy per information bit [21].

8.1.2 Channel model

In this chapter, we are going to describe and simulate the AWGN channel model with fixed gain (NO FADING) obtained from the link budget:

$$\mathrm{y(t)} = \sqrt{G}\,.\,x(t) + z$$

Where:
$G = Gtx\,.\,Grx\,.\,(\lambda/4/\pi/d)^2$

Where:
Gtx = Transmitter antenna gains
Grx = Receiver antenna gains
λ = Wavelength
d = Distance
x = Transmitted signal
z = Receiver noise: z $\sim CN(0, B\,.\,N_0)$

Transmitter

In the first step we generate a random binary sequence consisting of '-1's' and '+1's' as follows:

```
x=(2*floor(2*rand(1,Bit_Length)))-1;
```

Establish SNR

Usually, SNR is decibel expressed, but before using the SNR further, we convert the decibels to the normal ratio as follow:

$$SNR = 10^{\frac{SNRdB}{10}}$$

And in the Matlab, we find the ratio of $\frac{E_b}{N_0}$ as:

```
SNR = 10^(SNRdB/10);
```

Calculate EbNo

$\frac{E_b}{N_0}$ can be obtained by dividing SNR to "gross" link spectral efficiency in (bit/s)/Hz, as follow [26]:

$$\frac{E_b}{N_0} = SNR/\frac{f_b}{BW}$$

Where B is channel bandwidth, and f_b is channel data rate.

Using Matlab, we can obtain it as follow:

```
ebn0(k)=SNR(k)*(BW/fb(k));
%where "k" is in range of different SNRdB
```

Determination of Eb

Duration of one bit multiplied by the average signal power is energy-per-bit (E_b) which can be seen as follow:

$$E_b = \frac{1}{N \cdot f_{bit}} \sum_{n=1}^{N} x^2(n)$$

Where f_{bit} is the bit rate in bits-per-second and N is the total number of samples in the signal.

In order to find energy-per-bit (E_b) of signal "x" in Matlab:

```
eb = sum(x.^2)/(length(x) * fb(k))
%where "k" is in range of different SNRdB
```

Calculating N_0

For calculation of the one-sided power spectral density of the noise (N_0), which is the amount of noise power in 1.0 Hz signal bandwidth, first we should convert SNRdB to $\frac{E_b}{N_0}$, then divide energy-per-bit (E_b) by $\frac{E_b}{N_0}$. So, in Matlab, we have:

```
n0 = eb/ebn0(k)
%where "k" is in range of different SNRdB
```

Calculate noise variance (σ_n)

White Gaussian noise has a power spectral density commonly denoted by N0/2. This means that if we filter the noise with an ideal filter of bandwidth B, the noise at the filter's output is $\sigma^2 = BN0$. Noise has zero mean, so, its variance is identical to its power. We should know the noise bandwidth in

order to find the variance of the noise (average power of the noise). The noise bandwidth of the sampled signal at fs Hz, is half of the rate of sampling ($\frac{f_s}{2}$). So, average noise power can be calculated by noise bandwidth multiplied by power spectral density of the noise:

$$\sigma_n = \frac{f_s}{2} \,.\, N_0$$

In order to find noise variance (average power of noise) in Matlab, we use below code:

```
sigma= n0 * fs/2
```

Generate noise

The noise vector we want to produce must be the same length as the 'x(n)' signal vector. Also this noise vector must have variance of σ_n. In the Matlab we can use "randn" function in order to generate random numbers normally distributed with zero mean and variance equal to 1. Then, we should multiply this vector by $\sqrt{\sigma_n}$. So, noise vector can be obtained as:

```
n = (sqrt(sigma))*randn(1,length(x))
```

Transmission in AWGN channel

Therefore, we need to transmit this input signal through the AWGN channel, which based on our channel model can be represented as follows:

$$\text{y}=\sqrt{G} \,.\, \text{x}+\text{n}$$

Using Matlab we can use this block of code:

```
y=sqrt(G)*x+n;
```

For calculating received signal in different SNR_{dB} we should use bellow line of code:

```
RxSignal=awgn(y,snr_db(y),'measured');
```

Demodulation

The transmitted signal must be demodulate on the receiver side. Next, we need to compare a bit sequence that has been demodulate with the x vector to understand the number of error bits.

Error detection

In order to check for errors, we should determine the offset which is between the received bits and the transmitted bits. In this step, we can use soft detection in which we multiply each transmitted bit by each corresponding received bit (y. * X), since "x" vector contains "-1" or "+1", in any case of error we must obtain a negative number. So, we can detect all error bits.
For example:
If x(n) = -1 & y(n) = -1 $\Longrightarrow$ y .* x = +1 (No error)
But, if x(n) = +1 & y(n) = -1, $\Longrightarrow$ y .* x = -1 (Error detected)

To perform this scenario in Matlab, we can use the following code, where it finds the detected errors and then counts the number of errors based on different SNR values:
BER_simulated(k)=length(find((y.*x)<0));
%where "k" is in range of different SNRdB

Now, we have a vector "BER_Simulated" which contains number of errors in different SNRs that exist in different columns.Then, the bit error rate can be obtained by dividing the number of bit errors by the total number of bits transmitted as follows:
BER_Simulated=BER_Simulated/Bit_Length;
%where "Bit_Length" is the total number of transmitted bits

Plot SNR vs BER in Matlab

In order to plot SNR vs BER 8.1, we can use following line of code:

```
semilogy(SNRdB,BER_Simulated,'k->','linewidth',2.0);
title('BER of my model over AWGN Simulation');
xlabel('SNR in dB');
ylabel('BER');
axis tight;
grid on;
```

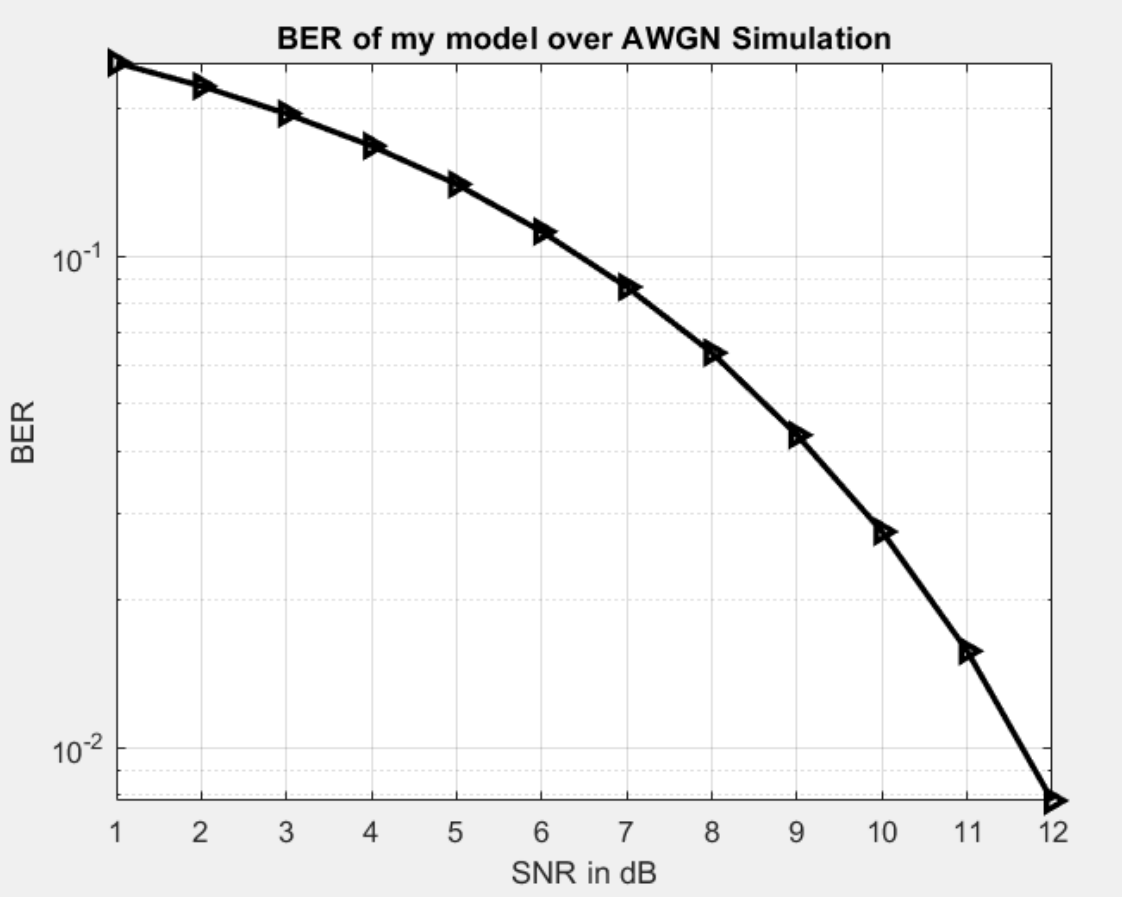

Figure 8.1. BER of my model over AWGN Simulation

Part II

Tracking system scenario based on LoRa

Chapter 9

LoRa

9.1 Lora overview

The use of LoRa radio in the sensor network has interesting aspects. First, because the range is relatively wide, networks without routing over many hops can span large areas. In many cases, one hop from each node to the sink is possible. Secondly, the transmission is orthogonal with the same carrier frequency but different spreading factor. This allows the channel to be subdivided into virtual channels. Thirdly, when the transfer occurs simultaneously with the same parameters, the strongest transfer is likely to be received, i.e. Simultaneous transmissions are not destructive even if their contents are different. This feature is exploited by LoRaWAN in which all gateways broadcast beacons at the same time and an end device is able to demodulate the most powerful beacon [17].

9.2 LoRa characteristics

Lora modulation scheme is based on spreading spectrum. It uses a modified form of Chirp spread spectrum modulation (CSS), in which the carrier frequency of a sinusoid varies linearly over a specific bandwidth. [16]. Several spreading factor (SF) have been defined to control bit rates, reduce power consumption, and improve range. In fact, Lora chirp behavior is controlled by SF and BW (bandwidth parameter) [10]. The main features of Lora include long range, low energy consumption, multi-path resistance, and high robustness. Transceivers of Lora often deployed in Industrial, Scientific and Medical (ISM) bands, but, they can also operate in licensed bands [15].

9.2.1 Transmission parameters

Lora devices can be configured to use different transmission power (TP), carrier frequency (CF), spreading factor (SF), bandwidth (BW) and coding rate (CR) [15].

Transmission Power (TP)

Transmission Power in radios of LoRa can be set from -4 dBm to 20 dBm, in 1 dB steps. However, due to limitation of hardware implementation, the range often limits from 2 dBm to 20 dBm. Moreover, power levels higher than 17 dBm can just be used on a 1% duty cycle [15].

Carrier Frequency (CF)

Carrier frequency is the center frequency which in 137MHz to 1020 MHz can be programmed in the steps of 61 Hz. This range may be limited to 860 MHz to 1020 MHz depend on the specific Lora chip [15].

Spreading Factor (SF)

Spreading Factor is the ratio among chip-rate and symbol-rate which can be selected from 6 to 12. One of the increasing factor of SNR, sensitivity and coverage is higher SF. Higher spreading factor also increase packet airtime. By increasing SF, for each summation, transmission rate gets half and consumption of energy and also duration of transmission increase. It should be noted that networks can be separated using different SFs [15].

Bandwidth (BW)

BW is the width of the transmission band frequencies. Lower bandwidth gives lower data-rate, but, higher sensitivity. Higher bandwidth gives higher data-rate and due to integration of additional noise, gives lower sensitivity. The range of bandwidth which can be selected is 7.8 kHz to 500kHz, but, operation of a typical Lora network is at 125kHz, 250kHz or 500kHz [15].

Coding Rate (CR)

Coding Rate is the Forward Error Correction (FEC) rate which provides protection against bursts of interference. CR can be 4/8, 4/7, 4/6, 4/5. Higher CR increase time on air, but provides more protection [15].

9.2.2 Chirp Spreading Spectrum (CSS) modulation

Spread spectrum systems have both the ability to communicate covertly and to resistance interference, fading, and other forms of interference. Basically, the spread spectrum approach involves the transmission of a signal that requires W_m bandwidth over the much wider bandwidth $W_s >> W_m$. As a result, suppressing the power spectral density of the transmission below the noise floor. Due to this, LoRa can operate below the level of noise, which can demodulate signals below the noise floor from -7.5 dB to -20 dB [36].
According to [34] there are two types of chirps which called up-chirp and down-chirp that show condition in up and down of the chirp, Which indicate the nature or changing the shape of the chirp signals.

$$SF = \frac{Chip\,rate}{Symbol\,rate}$$

In a transmission, SF is used to calculate duration of symbol T_s [27]:

$$T_s = \frac{2^{SF}}{BW}$$

SF affects the sensitivity S of the receiver. S can be obtained as follow:

$$S = -174 + 10\log 10 + NF + SNR$$

Where: -174 = $10\log 10$ (k * T * 1000) is because of the receiver thermal noise in 1 Hz bandwidth.
NF is the Noise Figure at the receiver (which is fixed for a hardware configuration data).
and SNR is the signal to noise ratio.

If fixed bandwidth is used in modulation, as the SF increases, so does the symbol duration. This makes the message stronger. It should be noted that for higher SF, the number of symbols increases. Therefore, the frequency symbol errors become larger. In this case, if small data is used, synchronization between RX and TX signals is particularly critical.
According to [27], table 9.1 shows LoRa bit rates, symbol durations and sensitivity in different SF.

The binary data rate expressed as follows [27]:

$$R_b = SF \,.\, \frac{B}{2^{SF}}$$

Mode	Bit rate (b/s)	Symbol duration (ms)	Sensitivity (dBm)
SF = 12	293	682	-137
SF = 11	537	365	-134.5
SF = 10	976	204	-132
SF = 9	1757	113	-129.5
SF = 8	3125	64	-127
SF = 7	5468	36	-124.5

Table 9.1. Lora bit rates, symbol durations and sensitivity vs SF [27]

9.2.3 Decoding LoRa frame

LoRa specifies a set of encoding values which are applied before modulation and transmission. The decoding steps are as follows:

1. Forward Error Correction (FEC):
   For FEC, Lora uses Hamming codes which is easy to implement. The code information word length is 4 bits and code word length is in range of 5 to 8 bits. The minimum hamming distance is 1, 2, 3, 4 , for the code rate of 4/5, 4/6, 4/7, 4/8 respectively ($\frac{4}{4+CR}$). By studying [27], error correction and error detection in different code-rate are described in the table 9.2.

Code Rate	Error Correction (bits)	Error detection (bits)
4/5	0	0
4/6	0	1
4/7	1	2
4/8	1	3

Table 9.2. Error correction and detection capabilities of Lora [27]

As shown in table 9.2, 4/5 has no advantage over no coding, a code equivalent to 4/6 only increases the number of error detections, a code rate of 4/7 introduces error correction, and at a code rate of 4/8, it improves error detection capability. However, it does not improve error correction.
It should be noted that the introduction of coding and the use of a code rate of 4/7 increases the payload length by 75% compared to no coding.

2. Data whitening:
   In order to provide more features for the receiver to recover the clock, data whitening is applied to the symbols to induce randomness. Symbols which are received, can be de-whitened by XOR with the same sequence of whitening which is used in transmitter side.

3. Interleaving:
   Data bits can be putted through the packet by using interleaving technique. With the most significant two reversed bits, a diagonal interleaver is implemented by Lora. Each diagonal word is moved or rotated by the desired number of bits. So, within each code word, the bits are reversed.

4. Gray indexing:
   In order to map a block of SF bits in S symbols in the constellation, Gray Indexing is used. To increase the chances of the channel code and correct possible errors it should ensure that two adjacent symbols differ by no more than 1 bit. The data rate can also expressed as follow:

   $$R_b = SF \cdot \frac{B}{2^{SF}} \cdot \frac{4}{4+CR}$$

   CR is the code rate, where CR $\in$ 1,2,3,4.

Chapter 10

Tracking scenario

10.1 Introduction

One of the most important services for people in the community is rescuing people, which, for example, can be done in vulnerable groups of people in case of loss or fall. In such cases, a wearable device can be useful for monitoring them. These devices can provide the minimum current position of vulnerable groups of people.
In general, to implement wearable devices, we should implement them in wireless networks. These devices require low power consumption for long-term operation as well as the ability to transmit data over long distances [20].

10.2 Implementation of LoRa

10.2.1 Lora Procedure

Generated bits pass to Lora modulation and then transfer to channel (AWGN) and then this signal will be demodulated.
After generating random bits, they must be converted from binary to gray based on different SFs, and then gray to decimal numbers. Lora signal generation is done by producing Lora signal with the corresponding preamble. In order make process efficient and easy, signal channelization converts signal of Lora to a base-band signal. After that, this signal enters in to AWGN channel. Now, the transmitted signal must be multiplied by the down-chirp signal. For extracting information from the data, calculating FFT is important to identify information on energy levels in this step. After that, data

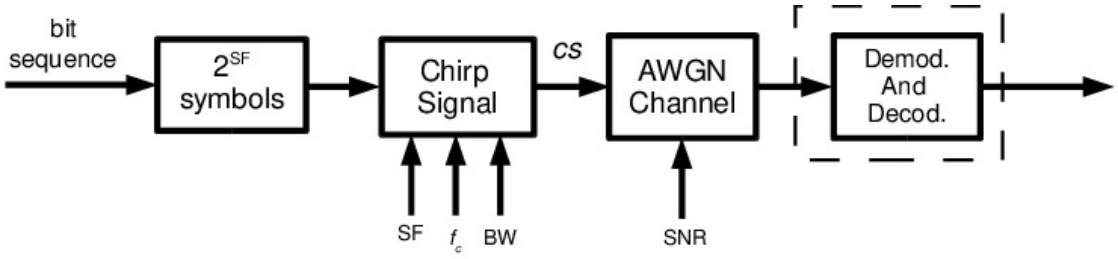

Figure 10.1. Communication System Block of LoRa [24]

should be decoded, and then, they have to be convert from decimal to gray and also gray to binary. So, the bits are received [16].

10.2.2 Analytical study

The chain of LoRa transmitter does the whitening, Hamming encoding, interleavering, and Gray mapping before the chirp modulation. The receiver performs Gray demapping, deinterleaving, Hamming decoding, and de-whitening.

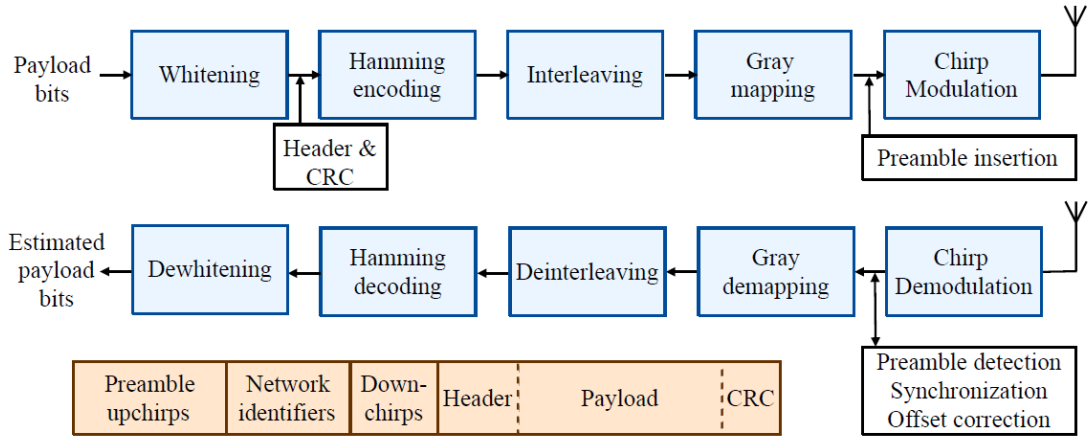

Figure 10.2. The LoRa packet structure [35]

Modulation

In CSS modulation, each LoRa chirp consists of a linear frequency sweep. Sweep is made during BW which is range of frequency. Duration of sweep

depends on value of SF and it called SD (Symbol duration). As I previously mentioned, in up-chirps, sweeps are in increase form and in down-chirps, sweeps are in decrease form [30].
Lora uses bandwidth B and chips per symbol equal to N, which means the bandwidth is divided to N frequency steps. A symbol s $\in$ S where $S = \{0, ..., N-1\}$ starts at $(\frac{sB}{N} - \frac{B}{2})$. Frequency increases by $\frac{B}{N}$ at each chip. When the frequency $\frac{B}{2}$ is reached, there is a frequency fold to $-\frac{B}{2}$ at chip $n_{fold} = N - s$ [33].
In the practical cases where sampling frequency is equal to bandwidth the equivalent equation of discrete-time baseband for a symbol (s) of Lora is:

$$x_s[n] = e^{j2\pi(\frac{n^2}{2N} + (\frac{s}{N} - \frac{1}{2})n)} \quad , \quad n \in S \text{ [33]}$$

After transmission over wireless channel with complex value channel gain $h \in \mathbb{C}$, the received LoRa symbol can be obtained as follow:

$$y[n] = hx_s[n] + z[n] \quad , \quad n \in S$$

where $z[n] \sim CN(0, \sigma^2)$ is complex additive white Gaussian noise with $\sigma^2 = \frac{N_0}{2N}$.

De-modulation

In order to demodulation the symbols, correlation of the received signal to all possible symbols $k \in S$ is as follow:

$$X_k = \sum_{n=0}^{N-1} y[n]x_k^*[n]$$

$$= |h| \sum_{n=0}^{N-1} e^{j2\pi(\frac{s-k}{N})n + \phi} + \sum_{n=0}^{N-1} z[n]x_k^*[n]$$

$$= |h| \sum_{n=0}^{N-1} e^{j2\pi(\frac{s-k}{N})n + \phi} + \tilde{z}_k$$

Where $\phi = \angle h$ is phase shift made by h (transmission channel) which for every transmitted packet is fixed. Generally it is distributed uniformly in $[0 - 2\pi)$ and $\tilde{z}_k \sim CN(0, N\sigma^2)$.
A symbol estimate $\hat{s}$ in a non-coherent receiver can be obtained as:

$$\hat{s} = \underset{k \in \mathcal{S}}{\text{argmax}}(|X_k|)$$

De-chirping is performed firstly which is multiplying the received signal by the complex conjugate of a reference signal x_{ref}. An Up-chirp is usually selected for the reference signal.

$$x_{ref}[n] = e^{j2\pi(\frac{n^2}{2N} - \frac{n}{2})} \quad , \quad n \in S$$

Normalized DFT will applied to make de-chirped signal in order to obtain:

$$\mathbf{Y} = \mathbf{DFT}\ (y \odot x^*_{ref})$$

Where $\odot$ is the Hadamard product (element wise vector multiplication)
$y = [y[0], ..., y[N-1]]$
$x_{ref} = [x_{ref}[0], ..., x_{ref}[N-1]]$

By choosing the frequency bin index with the maximum magnitude, demodulation can be performed.

$$\hat{s} = \underset{k \in \mathcal{S}}{\text{argmax}}(|Y_k|)$$

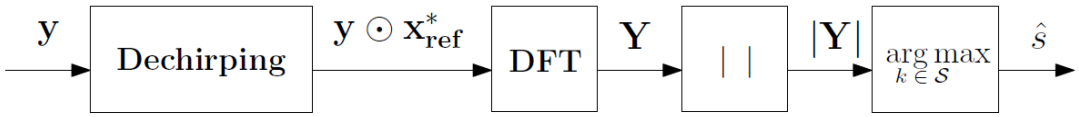

Figure 10.3. DFT-based LoRa demodulation chain illustration

In order to decode the symbol value, frontier of the symbol should be specified for the receiver. So, Lora uses preamble of some symbols (Usually 8) (in practice the receiver needs a minimum of 4 preamble symbols [31]). In up-link, the data includes of down-chirps and the preamble includes of up-chirps. While in down-link the data includes of up-chirps and the preamble includes of down-chirps [30].

10.3 Simulation study and results

For a probability of error in CSS, estimation of the analytical expression is as:

$$P_{e,CSS} = Q(\frac{log_{12}(SF)}{\sqrt{2}} \frac{E_b}{N_0}) \text{ [29]}$$

where Q(.) denote the Q-function.
It shows that for higher SF, the BER is more Intense. Energy depends on several parameters and vary from each systems and spreads over a large band. conversion from $\frac{E_c}{N_0}$ to $\frac{E_b}{N_0}$ is the problem, Where $\frac{E_c}{N_0}$ is the energy per chip-time to noise power spectral density ratio which obtained by:

$$\frac{E_c}{N_0} = \frac{E_b}{N_0} + 10 \,.\, log_{10}(\frac{R_c M_0}{SF}) + L \text{ [27]}$$

where R_c, M_0 is the code rate of the FEC, the modulation order in bits/sym. SF is the spreading factor.
and L is the additional implementation loss.
(It should be mentioned that for all spread spectrum applications $M_0 = 1$)
As we know, in CSS chip is equal to multiplication of SF by the symbol. So, the energy of each chip can be obtained by multiplying the energy per symbol to SF as follow:

$$\frac{E_c}{N_0} = \frac{E_s}{N_0} \,.\, SF$$

Where $\frac{E_s}{N_0}$ is the energy per symbol that can be express as follow:

$$\frac{E_s}{N_0} = 10 \log_{10}(\frac{f_{sym}}{f_{samp}}) + SNR \text{ [27]}$$

where f_{sym} denote the symbol frequency
f_{samp} is the sampling frequency
SNR is the signal-to-noise ratio

Therefore:

$$\frac{E_c}{N_0} = \frac{E_s}{N_0} \,.\, SF = [10 \log_{10}(\frac{f_{sym}}{f_{samp}}) + SNR] \,.\, SF$$

So, we obtain:

$$\frac{E_b}{N_0} = [10 \log_{10}(\frac{f_{sym}}{f_{samp}}) + SNR] \,.\, SF - 10.log_{10}(\frac{R_c M_0}{SF}) - L$$

And also:

$$P_{e,CSS} = Q(\frac{log_{12}(SF)}{\sqrt{2}}) \,.\, [10 \log_{10}(\frac{f_{sym}}{f_{samp}}) + SNR] \,.\, SF - 10.log_{10}(\frac{R_c M_0}{SF}) - L \text{ [27]}$$

According to [28], the relationship between SNR (dB) and $\frac{E_b}{N_0}$ (dB) can be as follow:

$$SNR(dB) = \frac{E_b}{N_0} + 10\log_{10}(R_s) + 10\log_{10}(k) + 10\log_{10}(R) - 10\log_{10}(BW_n)$$

This equation can be obtained by creating some relations as follows:

$$SNR = P_s/P_n$$

$$SNR(dB) = P_s - P_n$$

$$N_0 = \frac{P_n}{BW_n}$$

$$E_s = \frac{P_s}{R_s}$$

$$E_s(dB) = P_s - 10\log_{10}(R_s)$$

$$E_b = \frac{P_s}{R_b} = \frac{E_s}{kR}$$

$$E_b(dB) = E_s - 10\log_{10}(k) - 10\log_{10}(R)$$

$$E_b(dB) = P_s - 10\log_{10}(R_s) - 10\log_{10}(k) - 10\log_{10}(R)$$

$$\frac{E_b}{N_0}(dB) = P_s - 10\log_{10}(R_s) - 10\log_{10}(k) - 10\log_{10}(R) - P_n + 10\log_{10}(BW_n)$$

$$SNR(dB) = \frac{E_b}{N_0} + 10\log_{10}(R_s) + 10\log_{10}(k) + 10\log_{10}(R) - 10\log_{10}(BW_n)$$ [28]

Where
R_s : the symbol rate
k : the number of information bits per symbol
R : code rate
BW_n : the noise Bandwidth

10.3.1 Efficiency of the LoRa system (BER calculations)

We are going to compare the performance of different Lora SF from 7 to 12 for 125khz bandwidth, AWGN channel model is proposed system model in bandwidth of 125khz.

The code for the simulation has been written with help of the codes of Sakshama Ghoslya [32]. For the simulation, the Lora parameters are selected as follow:
SF = 7, 8, 9, 10, 11, 12
Bandwidth = 125000
Sampling Frequency = 125000
Preamble length = 8
Sync length = 2
Total bits to transmit = 27720

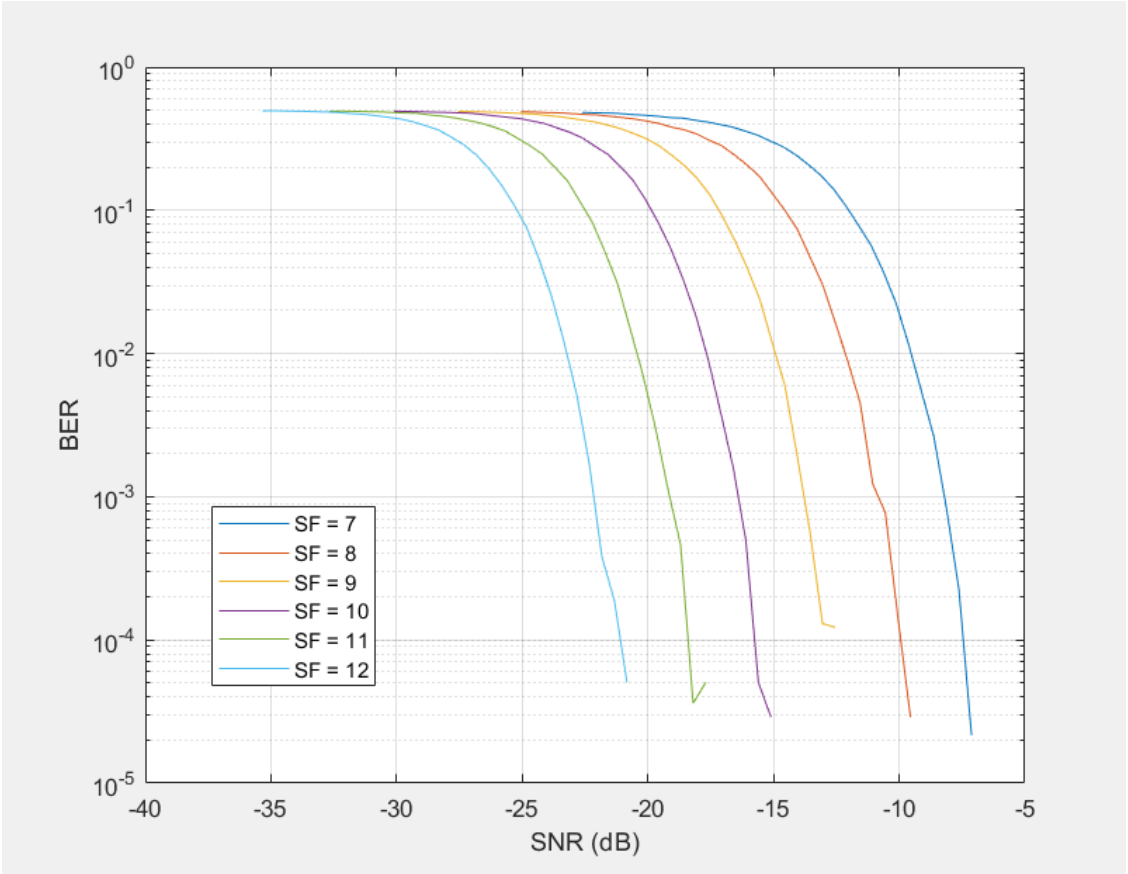

Figure 10.4. BER Estimation in different SF

As we can see in figure 10.4, In the higher SNR, the lower BER is seen. Therefore, with a higher SNR the probability of loss is lower. From BER point of view, higher SF is better and the signals are more resistant to noise,

but more Time on Air (ToA) which increases energy consumption, reduces the data rate, however, it improves communication range. Finally, increasing one unit of the SF, reduces the SNR within 3dB.
According to 9.2.2, transmission power is under the noise floor which is due to spread spectrum approach, thus, in the fig 10.4 negative SNR values have been illustrated. LoRa works below the noise level and SNR values in Lora are typically between -20dB and +10dB.

10.3.2 Spectrogram from Spreading Factor

Comparison of Spreading Factors in the 125 khz, 250 khz and 500 khz bandwidth showed at Fig 10.5, 10.6 and 10.7 which have been created with the help of [28].
As we can see, higher bandwidth gives a higher data rate and thus shorter Time on Air (ToA). But, due to integration of additional noise, it has a lower sensitivity. A lower bandwidth gives a lower data rate, but a higher sensitivity.

10.3.3 Useful bit-rate in Lora

LoRa includes a forward error correction code. According to the R_b formula, the code rate (CR) equals $4/(4 + C)$, with $C \in 1, 2, 3, 4$. By considering the fact that SF bits of information are transmitted per symbol, we can obtain figure 10.8 which compares bit-rate in different SF and different CR which is based on 125kHz bandwidth.
Transmitting a data packet with a higher SF infers extra bits are encoded into the chirp, so it reduced data rate, and increases Time on Air (ToA) in spite of the fact that it improves robustness to noise.

10.3.4 Sensitivity of LoRa receiver

The sensitivity of LoRa receiver as a function of SNR in different bandwidths is shown in the figure 10.9. It also shows that the sensitivity of LoRa receiver is affected by the increase of the bandwidth. This figure is based on spreading factor equal to 12. By using sensitivity equation we can have the result as table 10.1.

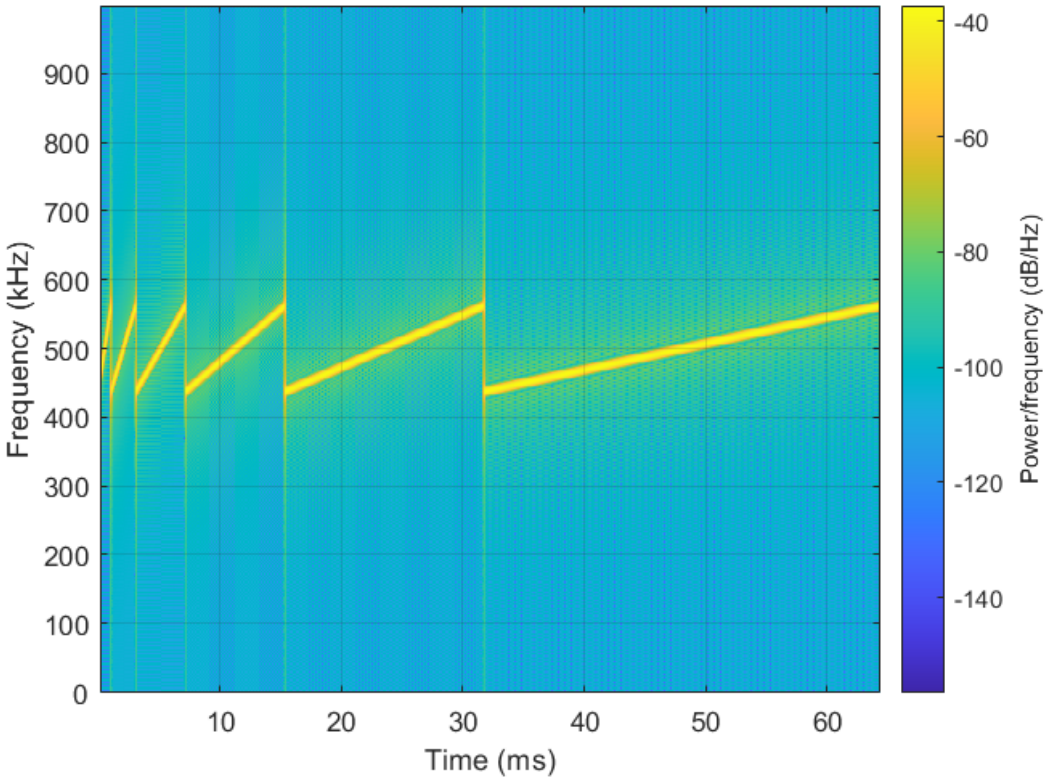

Figure 10.5. Comparison Spreading Factor (SF) on 125 kHz Frequency

Mode		Sensitivity (dBm)		
SF	SNR limit	125kHz	250kHz	500kHz
7	-7.5	-124.5	-121.5	-118.5
8	-10	-127	-124	-121
9	-12.5	-129.5	-126.5	-123.5
10	-15	-132	-129	-126
11	-17.5	-134.5	-131.5	-128.5
12	-20	-137	-134	-131

Table 10.1. Lora sensitivity vs SF

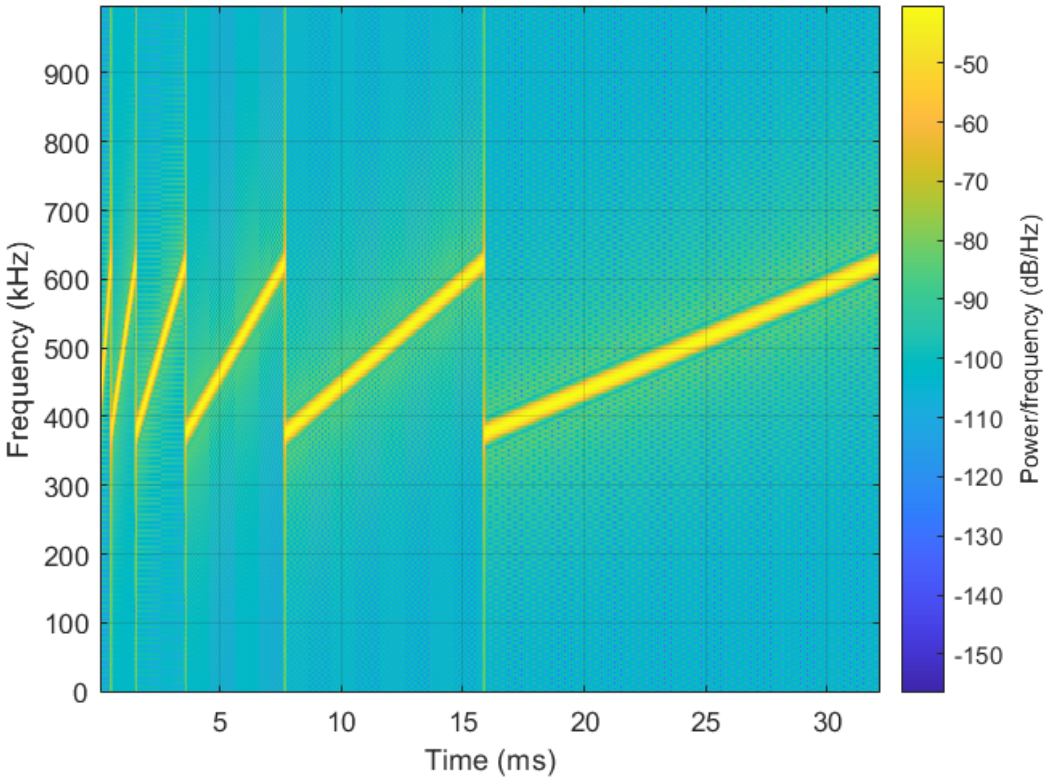

Figure 10.6. Comparison Spreading Factor (SF) on 250 kHz Frequency

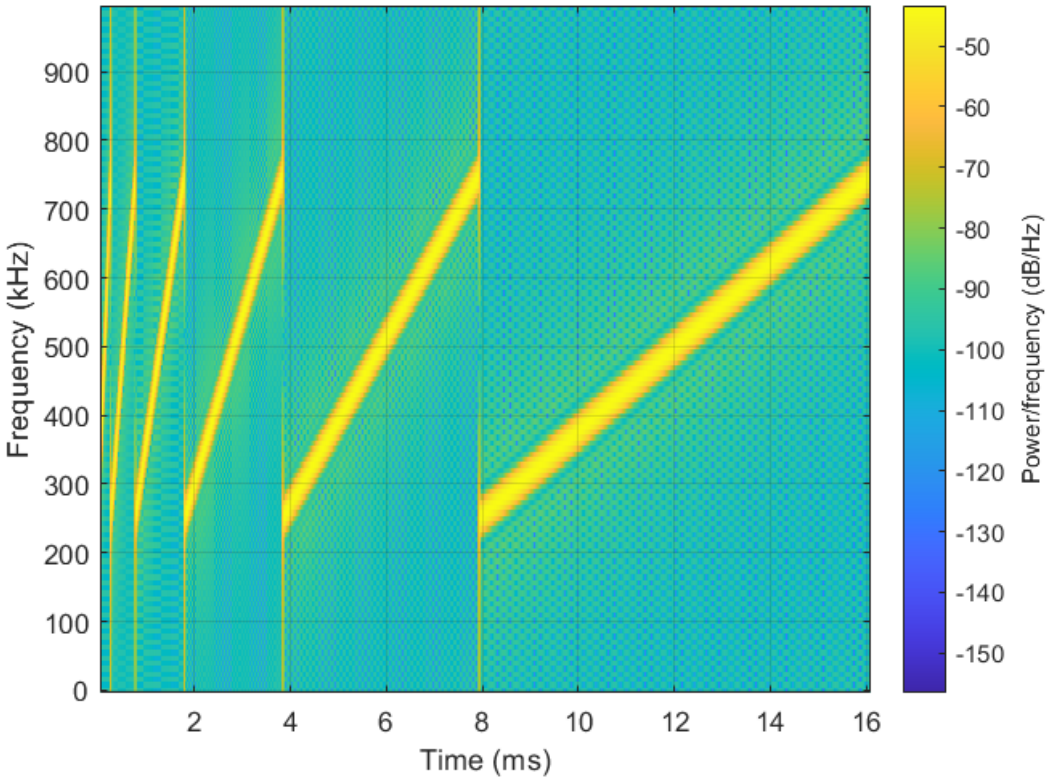

Figure 10.7. Comparison Spreading Factor (SF) on 500 kHz Frequency

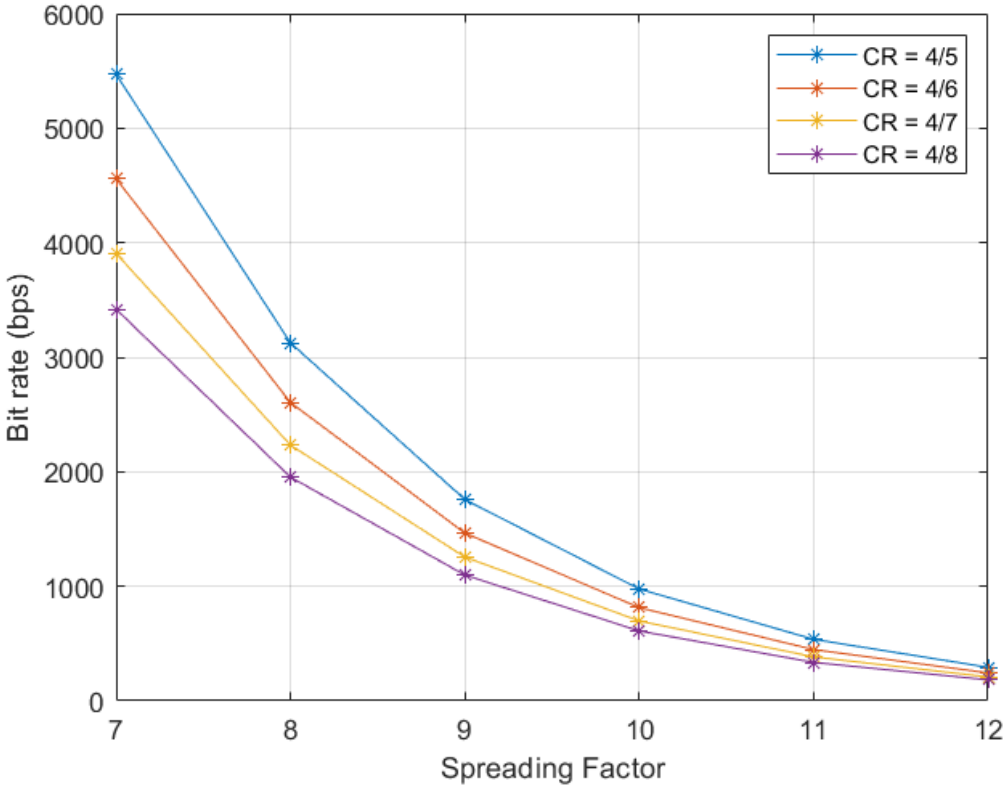

Figure 10.8. Comparison of bit-rate in different SF and CR

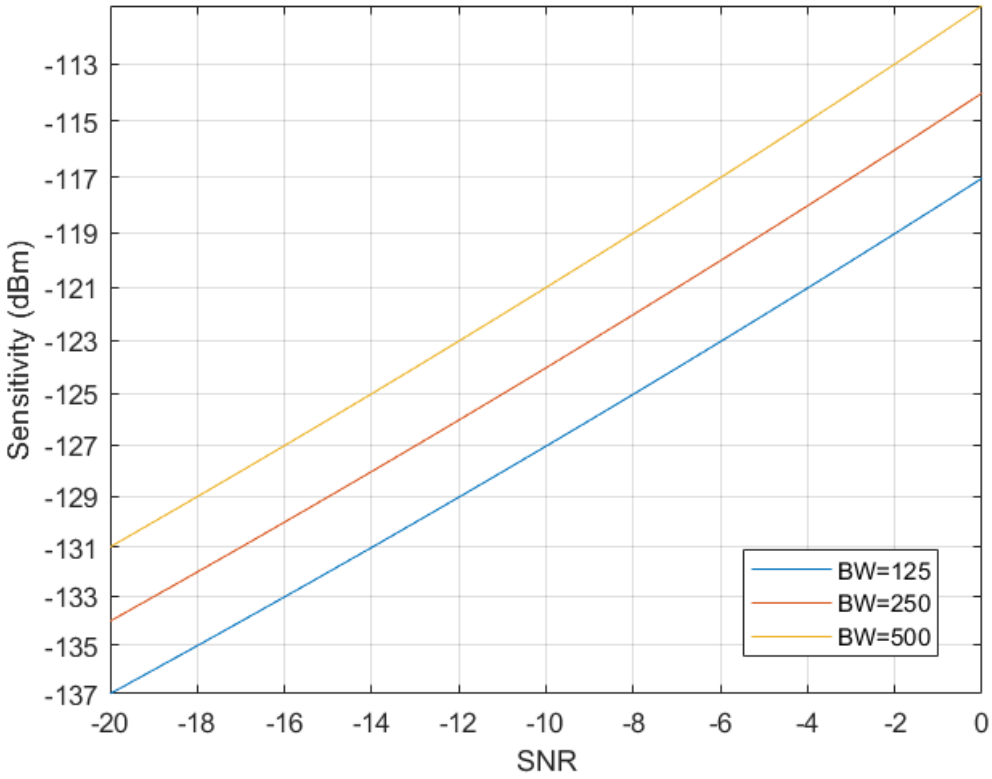

Figure 10.9. Sensitivity of LoRa receiver in different SNR for different bandwidth for SF = 12

Chapter 11

Conclusion

With the advent of the Internet of Things (IoT), everything around us can be automated. Our quality of life is improved by connecting many smart devices. Sensors and actuators become more powerful and cheaper, so makes them easier to use everywhere.
A comprehensive review of the Internet of Things, including architecture, security and privacy issues, potential technologies, and challenges, is presented in this thesis. The architectures that are possible for the IoT, including traditional three-layer architecture and four-layer SoA-based architecture, as well as possible technologies in layers (perception layer, network layer, and service layer) are described in detail.Also, the main issues of security and privacy that can affect the IoT have surveyed. In addition, the interaction between the IoT, big data, cloud and computing is discussed and some applications such as healthcare, industry and environment as well as smart cities are described.Security may be at the heart of many of these applications, so we surveyed the mechanisms needed to secure communications in IoT technologies. We described the advances in IoT-based healthcare as well as the benefits of stakeholders interested in IoT-based healthcare technologies.
Between some of the IoT technologies such as Sigfox, Lora, Zigbee, NB-IoT, LTE-M and IEEE 802.11ah which are briefly described, Lora has been selected for implementation in the "Tracking System Scenario".
In the next part, I covered various aspects of Lora technology such as transfer parameters, modulation in the Chirp-Spread-Spectrum, as well as the entire Lora process from transmitter, channel and receiver. The channel model is AWGN with a fixed gain, without fading. Then, Lora's behavior analysis was performed mathematically.

In the following, the physical layer of the IoT communication standard LoRa is studied. Approximation expression of the Bit Error Rate (BER) of LoRa is introduced and also approved by mathematical outcomes which is related to the physical and Data link layer parameters (spreading factor, coding rate, symbol frequency and sampling frequency)
In the next steps, Laura is simulated in Matlab to obtain results for wide-ranging SNR variations and different values of spreading factor and code rate.
It can be said that Low-Power-Long-Range (LoRa) is one of the most effective technologies in the field of IoT considering its long distance coverage and low energy consumption.

Appendix A

Appendix

A.1 Comparison of LoRa Spreading Factors

A.1.1 Main code

```
clear all;
close all;
clc;

% BW = 125000;      % 125kHz Bandwidth
% BW = 250000;      % 250kHz Bandwidth
BW = 500000;      % 500kHz Bandwidth
FS = 1000000;     % Sampling Frequency
inverse = 0;      % inverse = 0 for normal chirps

SF = 7;
SMPL_NUM = FS*(2^SF)/BW;  % Number of samples for SF = 7
[out_preamble1] = CSSmodulation(SF,BW,FS,SMPL_NUM,0,inverse);

SF = 8;
SMPL_NUM = FS*(2^SF)/BW;  % Number of samples for SF = 8
[out_preamble2] = CSSmodulation(SF,BW,FS,SMPL_NUM,0,inverse);

SF = 9;
SMPL_NUM = FS*(2^SF)/BW;  % Number of samples for SF = 9
[out_preamble3] = CSSmodulation(SF,BW,FS,SMPL_NUM,0,inverse);

SF = 10;
SMPL_NUM = FS*(2^SF)/BW;  % Number of samples for SF = 10
[out_preamble4] = CSSmodulation(SF,BW,FS,SMPL_NUM,0,inverse);
```

```
SF = 11;
SMPL_NUM = FS*(2^SF)/BW;  % Number of samples for SF = 11
[out_preamble5] = CSSmodulation(SF,BW,FS,SMPL_NUM,0,inverse);

SF = 12;
SMPL_NUM = FS*(2^SF)/BW;  % Number of samples for SF = 12
[out_preamble6] = CSSmodulation(SF,BW,FS,SMPL_NUM,0,inverse);

outp = [out_preamble1 out_preamble2 out_preamble3 ...
     out_preamble4 out_preamble5 out_preamble6];
samples = length(out_preamble1)/4;
spectrogram(outp,samples,samples-1,samples*2,FS,'yaxis');
grid on;
axis tight;
```

A.1.2 Lora CSS modulation function

```
%% Function of Lora modulation
function out_preamble = ...
    CSSmodulation(SF,BW,Fs,num_samples,symbol,inverse)

    %initialization
    phase = 0;
    Frequency_Offset = (Fs/2) - (BW/2);

    shift = symbol;
    out_preamble = zeros(1,num_samples);

    for k=1:num_samples

        %output the complex signal
        out_preamble(k) = cos(phase) + 1i*sin(phase);

        % Frequency from cyclic shift
        f = BW*shift/(2^SF);
        if(inverse == 1)
               f = BW - f;
        end

        %apply Frequency offset away from DC
        f = f + Frequency_Offset;
```

```
        % Increase the phase according to frequency
        phase = phase + 2*pi*f/Fs;
        if phase > pi
            phase = phase - 2*pi;
        end

        %update cyclic shift
        shift = shift + BW/Fs;
        if shift >= (2^SF)
            shift = shift - 2^SF;
        end
    end
end
```

Appendix B

Appendix

B.1 Lora BER estimation

B.1.1 Main code

```
clear all;
close all;
clc;

%% I will consider these parameters for the simulation
BW = 125e3; % Bandwidth
Fs = 125e3; % Frequency Sampling
LntPreamble = 8; % length of preamble
LntSync = 2; % length of sync
NumOfBits = 27720; % total transmitted bits (LCM for 7, 8, 9, 10, 11, 12)
SimRep = 5; % Total number of simulation repetition
D=1000; %distance
lambda=2398.339664; %wavelength unit is m
Gtx=2.25; % Antenna gain in transmitter
Grx=2.25; % Antenna gain in transmitter
G=Gtx*Grx*(lambda/4/pi/D)^2; % calculated based on link budget

%% Simulation over SF = 7
[Avg_BER7,snr_db7]=lorasimulate(7, G, BW, Fs, LntPreamble,...
    LntSync, NumOfBits, SimRep);
semilogy(snr_db7,Avg_BER7);
hold on

%% Simulation over SF = 8
[Avg_BER8,snr_db8]=lorasimulate(8, G, BW, Fs, LntPreamble,...
```

```
     LntSync, NumOfBits, SimRep);
semilogy(snr_db8,Avg_BER8);
hold on

%% Simulation over SF = 9
[Avg_BER9,snr_db9]=lorasimulate(9, G, BW, Fs, LntPreamble,...
     LntSync, NumOfBits, SimRep);
semilogy(snr_db9,Avg_BER9);
hold on

%% Simulation over SF = 10
[Avg_BER10,snr_db10]=lorasimulate(10, G, BW, Fs, LntPreamble,...
     LntSync, NumOfBits, SimRep);
semilogy(snr_db10,Avg_BER10);
hold on

%% Simulation over SF = 11
[Avg_BER11,snr_db11]=lorasimulate(11, G, BW, Fs, LntPreamble,...
     LntSync, NumOfBits, SimRep);
semilogy(snr_db11,Avg_BER11);
hold on

%% Simulation over SF = 12
[Avg_BER12,snr_db12]=lorasimulate(12, G, BW, Fs, LntPreamble,...
     LntSync, NumOfBits, SimRep);
semilogy(snr_db12,Avg_BER12);

%% Plotting details
xlabel('SNR(dB)');
ylabel('BER');
legend('SF = 7','SF = 8','SF = 9','SF = 10','SF = 11','SF = 12');
grid on;
```

B.1.2 Lora simulation function

```
%% Main function of simulation
function [Avg_BER,SNR_dB] = ...
    lorasimulate(SF, G, BW, Fs, LntPreamble,...
    LntSync, NumOfBits, SimRep)

%% Some initial calculation
NumOfSamples = Fs*(2^SF)/BW; % Number of Samples
EbNo_db = -10:0.5:10; % Eb/No (Energy bits/Noise) rate in dB
```

```
EbNo =  ...
    10.^(EbNo_db/10); % Eb/No (Energy bits/Noise) rate
SNR_dB = ...
    EbNo_db + 10*log10(BW/(2^SF)) + 10*log10(SF)-10*log10(BW); ...
    % signal to Noise rate in dB
SNR = 10.^(SNR_dB/10); % signal to Noise rate
Fb= (SF*BW/2^SF); %Bit rate
BER_vec = zeros(SimRep,length(SNR_dB)); % this will be used at the
    % end for counting errors

%% This will generate random number for transmitting
[InBinaryVector, lntInput] = LoraRandNum(NumOfBits,SF);
MatrixIn = reshape(InBinaryVector, SF, lntInput);
GaryIn = ...
BinaryToGray(MatrixIn); % converting Binary to Gray
DecIn = ...
bi2de(GaryIn','left-msb')'; % Converting Binary to Decimal
TotalSym = ...
LntPreamble + LntSync + lntInput; % Total symbols transmitted

%% Now This will generate Preamble
inverse = 0;
for i = 1:LntPreamble
[out_preamble] = CSSmodulation(SF,BW,Fs,NumOfSamples,0,inverse);
outp((i-1)*NumOfSamples+1 : i*NumOfSamples) = out_preamble;
end

%% And also generating sync symbols
inverse = 1;
for i = 1:LntSync
[out_sync] = CSSmodulation(SF,BW,Fs,NumOfSamples,32,inverse);
outp = [outp out_sync];
end

%% Now the main symbols generation
inverse = 0;
for i = 1:lntInput
[out_sym] = ...
    CSSmodulation(SF,BW,Fs, NumOfSamples,DecIn(i),inverse);
outp = [outp out_sym];
end
for ite = 1:1:SimRep % start simulation for "SimRep" times
for y=1:length(SNR_dB)

%%  channelization of the Signal(DDC)
t = 0:1/Fs:length(outp)/Fs-1/Fs;
```

```
outp = outp.*cos(2*pi*Fs*t);%  Bring signal to baseband

%% Transmission in AWGN channel with fixed gain (By using
%% parameter of G)
for k=1:length(SNR_dB)
eb = sum(outp.^2)./(NumOfBits*Fb); %energy-per-bit
n0 = eb./EbNo_db(k); %calculate noise density
sigma = n0 * BW; %Calculate noise variance
n = (sqrt(sigma))*randn(1,length(outp));
TxSignal=sqrt(G)*outp+n;
RxSignal=awgn(TxSignal,SNR_dB(y),'measured');
end

%% In the receiver we have to generate reverse chirp
inverse = 1;
[out_reverse] = CSSmodulation(SF,BW,Fs,NumOfSamples,0,inverse);

%% Now, multiplying with the reverse chirp
for n = 1:1:TotalSym
decoded_out((n-1)*NumOfSamples + 1 : n*NumOfSamples) = ...
RxSignal((n-1)*NumOfSamples + 1 : n*NumOfSamples).*out_reverse;
end

%% FFT calculation in order to implement a demodulator
%% to compute the dechirped vector
for m = 1:1:TotalSym
FFT_out(m,:) = ...
abs((fft(decoded_out((m-1)*NumOfSamples + 1 : m*NumOfSamples))));
end

%% Decoding the received data
k=1;
for m = LntPreamble+LntSync+1:1:TotalSym
[r,c] = max(FFT_out(m,:));
data_received(k) = c-1;
k = k+1;
end
BinOut = ...
de2bi(data_received,SF,'left-msb')';% Converting Decimal to Binary
GrayOut = ...
GrayToBinary(BinOut);% Converting Gray to Binary
DecOut = ...
reshape(GrayOut,NumOfBits,1);% Converting Matrix to array

%% It will Counts BER
BER_vec(ite,y) = sum(abs(DecOut - InBinaryVector))/NumOfBits;
```

```
end
end
Avg_BER(1,:) = mean(BER_vec); % Average BER over all the iterations
end
```

B.1.3 Lora CSS modulation function

```
%% Function of Lora modulation
function out_preamble = ...
    CSSmodulation(SF,BW,Fs,num_samples,symbol,inverse)

    %initialization
    phase = 0;
    Frequency_Offset = (Fs/2) - (BW/2);

    shift = symbol;
    out_preamble = zeros(1,num_samples);

    for k=1:num_samples

        %output the complex signal
        out_preamble(k) = cos(phase) + 1i*sin(phase);

        % Frequency from cyclic shift
        f = BW*shift/(2^SF);
        if(inverse == 1)
               f = BW - f;
        end

        %apply Frequency offset away from DC
        f = f + Frequency_Offset;

        % Increase the phase according to frequency
        phase = phase + 2*pi*f/Fs;
        if phase > pi
            phase = phase - 2*pi;
        end

        %update cyclic shift
        shift = shift + BW/Fs;
        if shift >= (2^SF)
            shift = shift - 2^SF;
        end
```

```
    end
end
```

B.1.4 Lora Random Number function

```
%% Function of generating random number for Lora
function [vectorRand_input, columns] = LoraRandNum(NumOfBits, SF)

 rows = SF;
 columns = ceil(NumOfBits/SF);
vectorRand_input = round(0.75*rand(1,NumOfBits))';
```

B.1.5 Binary To Gray converter function

```
%% Function of converting Binary to Gray code
function [Gray] = BinaryToGray(Binary)
 [row,column] = size(Binary);
 Gray = zeros(row,column);
 Gray(1,:) = Binary(1,:);
 for c = 1:1:column
 for r = 2:1:row % Xor of input bit with last input bit
 Gray(r,c) = xor(Binary(r,c), Binary(r-1,c));
 end
 end
```

B.1.6 Gray To Binary converter function

```
%% Function of converting gray Code to Binary
function [binary] = GrayToBinary(gray)
 [row,column] = size(gray);
 binary = zeros(row,column);
 binary(1,:) = gray(1,:); % Copying Firstbit

 for c = 1:1:column
 for r = 2:1:row % Xor of input bit with last output bit
```

```
binary(r,c) = xor(binary(r-1,c),gray(r,c));

end
end
```

Bibliography

[1] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain and K. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," in *IEEE Access*, vol. 3, pp. 678-708, 2015.

[2] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[3] J. Granjal, E. Monteiro and J. Sá Silva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294-1312, third quarter 2015.

[4] L. D. Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233-2243, Nov. 2014.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347-2376, Fourth quarter 2015.

[6] Ikram Ud Din, Suhaidi Hassan, Muhammad Khurram Khan, Mohammed Atiquzzaman, Syed Hassan Ahmed, "The Internet of Things: A Review of Enabled Technologies and Future Challenges," in *IEEE Access*, vol. 7, pp. 7606-7640, 2019.

[7] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities," in *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, Feb. 2014.

[8] https://behrtechnologies.com/blog/tag/lpwan-comparison/

[9] C. Gomez, J. C. Veras, R. Vidal, L. Casals, and J. Paradells, “A Sigfox Energy Consumption Model,” Sensors, vol. 19, no. 3, p. 681, Feb. 2019.

[10] G. Ferré and A. Giremus, "LoRa Physical Layer Principle and Performance Analysis," *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Bordeaux, 2018, pp. 65-68

[11] A. I. Ali, S. Z. Partal, S. Kepke and H. P. Partal, "ZigBee and LoRa based Wireless Sensors for Smart Environment and IoT Applications," *2019 1st Global Power, Energy and Communication Conference (GPECOM)*, Nevsehir, Turkey, 2019, pp. 19-23

[12] R. Mozny, P. Masek, M. Stusek, K. Zeman, A. Ometov and J. Hosek, "On the Performance of Narrow-band Internet of Things (NB-IoT) for Delay-tolerant Services," *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, Budapest, Hungary, 2019, pp. 637-642

[13] S. Dawaliby, A. Bradai and Y. Pousset, "In depth performance evaluation of LTE-M for M2M communications," *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, New York, NY, 2016, pp. 1-8

[14] E. Khorov et al., "Enabling the Internet of Things With Wi-Fi Halow—Performance Evaluation of the Restricted Access Window," in *IEEE Access*, vol. 7, pp. 127402-127415, 2019

[15] M. Bor and U. Roedig, "LoRa Transmission Parameter Selection," *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Ottawa, ON, 2017, pp. 27-34

[16] S. Stewart, H. H. Nguyen, R. Barton, and J. Henry, "Reducing the Cost of Implementing Filters in LoRa Devices," *Sensors*, vol. 19, no. 18, p. 4037, Sep. 2019.

[17] M. Bor, J. Vidler, and U. Roedig, "LoRa for the Internet of Things," *Proc. 2016 Int. Conf. Embed. Wirel. Syst. Networks*, pp. 361–366, March 2016.

[18] John Lampe, Zbigniew Ianelli, "Introduction to Chirp Spread Spectrum (CSS) Technology" *Project: IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs)*, November 2003.

[19] Sadeque, G. Bit Error Rate (BER) Comparison of AWGN Channels for Different Type's Digital Modulation Using MATLAB Simulink. *Am. Sci. Res. J. Eng. Technol. Sci.* pp. 61–71, June 2015.

[20] C. Bouras, A. Gkamas, V. Kokkinos and N. Papachristos, "Using LoRa Technology for IoT Monitoring Systems," *2019 10th International Conference on Networks of the Future (NoF)*, pp. 134-137, Oct 2019.

[21] B. L. Ahlem, M. Béchir Dadi and C. Belgacem Rhaimi, "Evaluation of

BER of digital modulation schemes for AWGN and wireless fading channels," *2015 World Congress on Information Technology and Computer Applications (WCITCA)*, pp. 1-5, June 2015.

[22] W. Wang, S. L. Capitaneanu, D. Marinca and E. Lohan, "Comparative Analysis of Channel Models for Industrial IoT Wireless Communication," in *IEEE Access*, vol. 7, pp. 91627-91640, 2019.

[23] B. Reynders, W. Meert and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," *2016 23rd International Conference on Telecommunications (ICT)*, pp. 1-6, May 2016.

[24] V. Fialho, F. Azevedo, "Wireless Communication Based on Chirp Signal for LoRa IoT Devices," *i-ETC:ISEL Academic Journal of Electronics, Telecommunications and Computers*, vol. 4, no. 1, 2018.

[25] Semtech LoRa Technology | https://www.semtech.com/lora/why-lora

[26] Noise, S/N and Eb/No, Richard Wolff, EE447, Fall 2011

[27] H. Mroue, A. Nasser, B. Parrein, S. Hamrioui, E. Mona-Cruz and G. Rouyer, "Analytical and Simulation study for LoRa Modulation," 2018 25th International Conference on Telecommunications (ICT), pp. 655-659, June 2018.

[28] Puput Dani Prasetyo Adi, Akio Kitagawa, "A Study of LoRa Performance in Monitoring of Patient's SPO_2 and Heart Rate based IoT", *(IJACSA) International Journal of Advanced Computer Science and Applications* ,Vol. 11, No. 2, pp. 238-251, January 2020.

[29] B. Reynders, W. Meert and S. Pollin, "Range and coexistence analysis of long range unlicensed communication," *2016 23rd International Conference on Telecommunications (ICT)*, pp. 1-6, May 2016.

[30] N. E. Rachkidy, A. Guitton and M. Kaneko, "Decoding Superposed LoRa Signals", *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, pp. 184-190, Oct 2018.

[31] C. Bernier, F. Dehmas and N. Deparis, "Low Complexity LoRa Frame Synchronization for Ultra-Low Power Software-Defined Radios," in *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 3140-3152, May 2020.

[32] All About LoRa and LoRaWAN "http://www.sghoslya.com/".

[33] O. Afisiadis, M. Cotting, A. Burg and A. Balatsoukas-Stimming, *"On the Error Rate of the LoRa Modulation With Interference,"* in IEEE Transactions on Wireless Communications, vol. 19, no. 2, pp. 1292-1304, Feb. 2020.

[34] Puput Dani Prasetyo Adi and Akio Kitagawa, *"Performance Evaluation of E32 Long Range Radio Frequency 915 MHz based on Internet of Things and Micro Sensors Data"* International Journal of Advanced Computer

Science and Applications(IJACSA), 10(11), 2019.

[35] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, *"An open-source LoRa physical layer prototype on GNU radio,"* in International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), May 2020,

[36] B. A. Bash, D. Goeckel, D. Towsley and S. Guha, "Hiding information in noise: fundamental limits of covert wireless communication," in IEEE Communications Magazine, vol. 53, no. 12, pp. 26-31, Dec. 2015