Building a Test Bed for Simulation Analysis for the Internet of Things

Lucas Novelli

Dissertation presented to the School of Technology and Management to obtain the Master's Degree in Information Systems

Research under mentoring:

Luísa Maria Garcia Jorge

Paulo Melo

André Koscianski

Bragança

October 2018

Acknowledgment

I acknowledge and express thanks to both Universidade Tecnológica Federal do Paraná – Campus Ponta Grossa and the Instituto Politécnico de Bragança for the opportunity provided and for all the effort made by these two institutions and the people who represent it, that allowed me this study and research opportunity.

I appreciate all the help and patience that my mentors Drª Luisa Maria Garcia Jorge, Dr. Paulo Melo, and Dr. André Koscianski had with me during our work in this project.

I also like to thank my mother Raquel for the support during all my years at college, without her it would not be possible to achieve this point. All the friends that I made in Bragança for supporting me in both good and bad moments.

Abstract

The Internet of Things (IoT) enables the mix between the physical and informational world. Physical objects will be able to see, hear, think together, share information and coordinate decisions, without human interference in a variety of domains. To enable this vision of IoT in large scale is expected of the equipment to be low-cost, mobile, power efficient, computational constrained, and wireless communication enabled.

This project performs an extensive overview of the state-of-the-art in communication technologies for IoT, simulation theory and tools. It also describes test bed for IoT simulation and its implementation.

The simulation was built with Castalia Simulator (i.e. Wireless Sensor Networks (WSN) network) and INET framework (i.e. IP network), both extends OMNeT++ features. There are two independent networks that communicate through files and exchange information about source, destination, payload and simulation time.

Analyzing the outputs is possible to assure that the routing protocol that is provided in the Castalia Simulator does not provide any advantage in terms of packets loss, packets reception or energy consumption.

Key Words: IoT, Simulation, Protocols, Networks

Resumo

A Internet das Coisas (IoT) permite a mistura entre o mundo físico e informacional. Objetos físicos serão capazes de ver, ouvir, pensar juntos, compartilhar informações e coordenar decisões, sem interferência humana em uma variedade de domínios. Para permitir essa visão de IoT em larga escala, espera-se que o equipamento seja de baixo custo, móvel, eficiente em termos de energia, com restrições computacionais e possibilite a comunicação sem fio.

Este projeto faz uma extensa visão geral do estado da arte em tecnologias de comunicação para IoT, teoria de simulação e ferramentas. Também descreve o banco de testes para simulação de IoT e sua implementação.

A simulação foi construída com o Simulador Castalia (ou seja, rede WSN) e o framework INET (ou seja, rede IP), ambos estendem os recursos do OMNeT ++. Existem duas redes independentes que se comunicam através de arquivos e trocam informações sobre origem, destino, carga útil e tempo de simulação.

Analisando os resultados é possível garantir que o protocolo de roteamento que é fornecido no Simulador Castalia não oferece qualquer vantagem em termos de quebra de pacotes, recepção de pacotes ou consumo de energia.

Palavras-chave: Internet das Coisas; Simulação; Protocolos; Redes.

Contents

List of Acronyms

6LoWPAN: IPv6 over Low power Wireless Personal Area Networks, 3, 26, 27, 33, 45, 8, 10, 11, 13

ADR: Adaptive Data Rate, 48

AMQP: Advanced Messaging Queuing Protocol, 14, 19

APL: Application, 33

ARP: Address Resolution Protocol, 57

ATT: Attribute Protocol, 38

BAN: Body Area Network, 2, 9, 58, 65

BER: Bit Error Rate, 37

BFSK: Binary Frequency Shift Keying, 35

BGP: Border Gateway Protocol, 57

BLE: Bluetooth Low Energy, 36

BPSK: Binary Phase Shift Keying, 49

CoAP: Constrained Application Protocol, 14, 15

CSMA-CA: Carrier-Sense Multiple Access with Collision Avoidance, 33

CSS: Chirp Spread Spectrum, 47

CW: Continuous Wave, 40

DCCP: Datagram Congestion Control Protoco, 25

DCPS: Data-Centric Publish-Subscribe, 16, 17

DDS: Data Distribution Service, 14, 16

DES: Discrete Event Simulation, 51

DLRL: Data-Local Reconstruction Layer, 17

DNS: Domain Name Server, 14, 21, 22, 23

DNS-SD: Domain Name Server Service Discovery, 22

DODAG: Destination Oriented Directed Acyclic Graph, 23

Index of Figures

Index of Tables

Chapter 1 Introduction

This chapter gives a quickly overview about the Internet of Things concepts and the technologies e.g. protocols, wireless networks, radio communication. This chapter also presents the simulation approach, the objectives, methodology and the structure of the upcoming chapters.

1.1. Internet of Things

The Internet of Things (IoT) enables the mix between the physical and informational world. Physical objects will be able to see, hear, think together, share information and coordinate decisions in a variety of domains, such as: transportation, healthcare, house and industrial automation, prevention of natural and man-made disasters, elderly assistance [1], intelligent energy manager and smart grids, automotive, traffic management [2], and so on.

It is possible to categorize the implementation of the IoT vision in two major areas, namely: The use of Wireless Personal Area Networks (WPAN) to enable Wireless Sensor Networks (WSN) for house and industry automation, and the use of Low Power Wide Area (LPWA) networks to construct solutions for smart cities, smart metering, and smart buildings.

1.2. Simulation models and tools

Discrete Event Simulation (DES) provides a model, composed by variables representing the state of the simulation, but it may, however, display poor scalability for IoT scenarios [3].

To overcome the structural scalability problems of DES, Parallel Discrete Event Simulation (PDES) and Parallel and Distributed Simulation (PADS) divide the simulation model, processing each part in different CPUs or different hosts, increasing the simulation scalability. However, these approaches bring some drawbacks including complex partitioning, synchronization algorithms, and is required a distribution data management [3].

OMNeT++ is an object-oriented PDES, created using the C++ language and totally open source. This simulator can be used for a wide variety of purposes, that can include modeling wired and wireless communication networks and queueing networks, and validating protocols [4]. In general, it is useful for any problem that can be modeled by discrete events and that can be mapped in entities that exchange messages for communication [5].

To minimize the amount of effort to build Internet Protocol (IP) networks in OMNeT++, the INET Framework was created. INET contains the implementation of many useful components and protocols, including: Internet Protocol version 4 (IPv4), Internet Protocol version 6 (IPv6), Transmission Control Protocol (TCP) and User Datagram Protocol (UDP).

The Castalia Simulator is widely used to simulate WSN and Body Area Networks (BAN), by expanding the OMNeT++ base system with advanced channels and radio models based on empirically measured data. It also include Medium Access Control (MAC) and routing protocols, and a highly flexible physical model [6].

1.3. Motivations

Many communication protocols (e.g. IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), Constrained Application Protocol (CoAP), Data Distribution Service (DDS), MQTT For Sensor Networks (MQTT-SN) and wireless standards (e.g. Institute of Electrical and Electronics Engineers (IEEE) 802.15.4, BLE, LoRa, NB-IoT, SigFox) were created or modified to support IoT requirements, which include power efficiency, low throughput, and reliability. Therefore, in real-world IoT scenarios, many combinations of these networks and protocols can be applied to ensure optimal operation given the huge number and the heterogeneity of devices, many environmental factors, costs, and devices distribution.

To analyze the behavior of IoT solutions, a simulation approach appears to be adequate, since it can include many protocols that can be applied both for WSN or LPWA networks.

The alternative approach would be the use of a physical implementation of the WSN and LPWA networks, however, this option would impose much higher costs, and an inflexible solution in terms of topology, setups, and configuration. Also, the possible combinations of application protocols would be restricted to the devices capability and the time to collect and analyze all the real traffic generated in the network would be another restriction.

1.4. Objectives

The main object of this work is to analyze the state of art wireless communication solutions for IoT, namely if they are achieving the needs and respecting the constraints this environment requires. To achieve this main objective, the following structure for the analysis was proposed:

- Make a survey of the state of the art in wireless communication and simulation;

- Describe and design simulation models for the cases under study;
- Implement the proposed simulation models in the OMNET++ and analyze the simulation results

1.5. Methodology

The methodology used to create the state-of-the-art survey consisted in bibliographical research using renowned indexed data bases including: IEEExplorer, ScienceDirect, Springer, Wiley and ACM. All the papers used to create the survey of state of the art in IoT communication and simulation, came from the data bases listed above, and their impact were also analyzed using online metrics tools, such Google Scholar.

To achieve a better comprehension about the simulation tools, the manuals provided by the tool's developers were very useful, providing an overview about what was possible to do and what was not.

During the development process, many of the issues were resolved using knowledge provided by the community in forums, blogs and sites. Their description of previous difficulties and experience were a great help for this project.

1.6. Document Structure

Chapter 2 presents the state of art of the communication process in an IoT environment. It presents the application of the many available standards, protocols, networks and architectures in terms of their requirements.

Chapter 3 describes the proposed testbed, its characteristics, specificities, application and market.

Chapter 4 and Chapter 5 offers an introduction to computer simulation as well as an extensive overview about the features present in the simulation tools that were applied in this project.

Chapter 6 shows all the steps that were taken to achieve a good representation of the testbed in the deployment phase.

Chapter 7 presents the simulations and their analysis. Finally Chapter 8 provides the conclusion and some thoughts of possible future work.

Chapter 2 Internet of Things and Case Studies Definition

This chapter explains the IoT main concepts, an overall description of enabling communication technologies and describes the use case proposed for this project.

2.1. An IoT Solution

Given the complexity involved in implementing an IoT solution, it can require up to six distinct elements that perform different services [1]; these elements are illustrated in Figure 1.

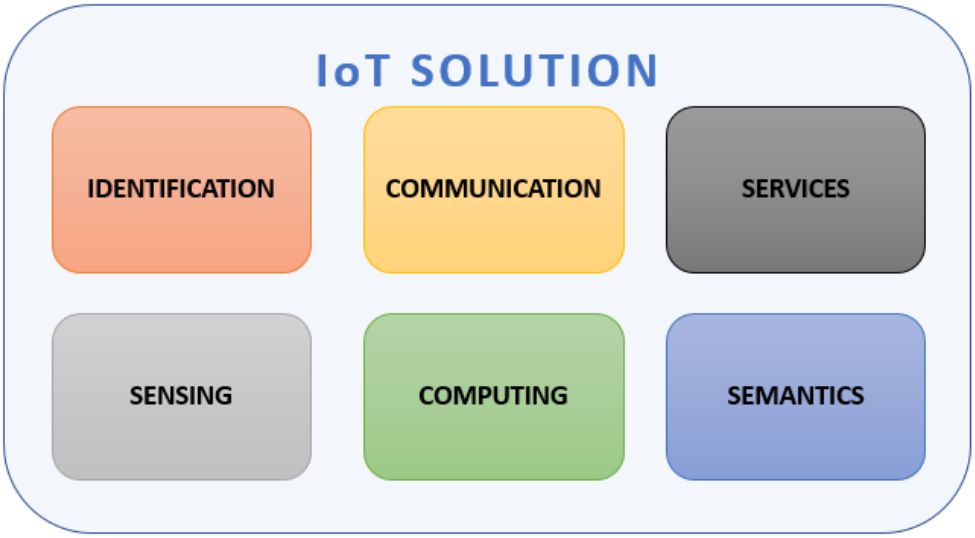

Figure 1: Components of an Internet of Things solution (Adapted from A. Al-Fuqaha et al., 2015).

These elements and their roles are described in [1] as:

- Identification: Each object in an IoT solution must have a unique identification / address;
- Sensing: Devices that can retrieve data from the environment or from an object and send this to a processing service;
- Communication: Technologies that are used to connect heterogeneous devices together in the same network;
- Computation: Computation is performed on the data that has been collected. It can happen within the IoT device itself, using microcontrollers, microprocessors; or in an external service, usually through cloud computing;
- Services: IoT applications achieve ubiquity by exposing services in the Internet. This way they can have all the services available anywhere, anytime and for anyone who wants to access it (subject to authorization);
- Semantics: refers to being able to extract knowledge from different machines; for this, all of them must understand a common language.

2.2. IoT Protocols and Technologies

Throughout the years, many protocols were created, or those that already exist were adapted, to provide efficient communication concerning the requirements for the components described above, some for sensors nodes in the WSN and others for constrained devices that integrate an IoT solution.

Figure 2 shows a direct link between the layers of the TCP/IP Model and the most significant protocols and technologies for IoT.

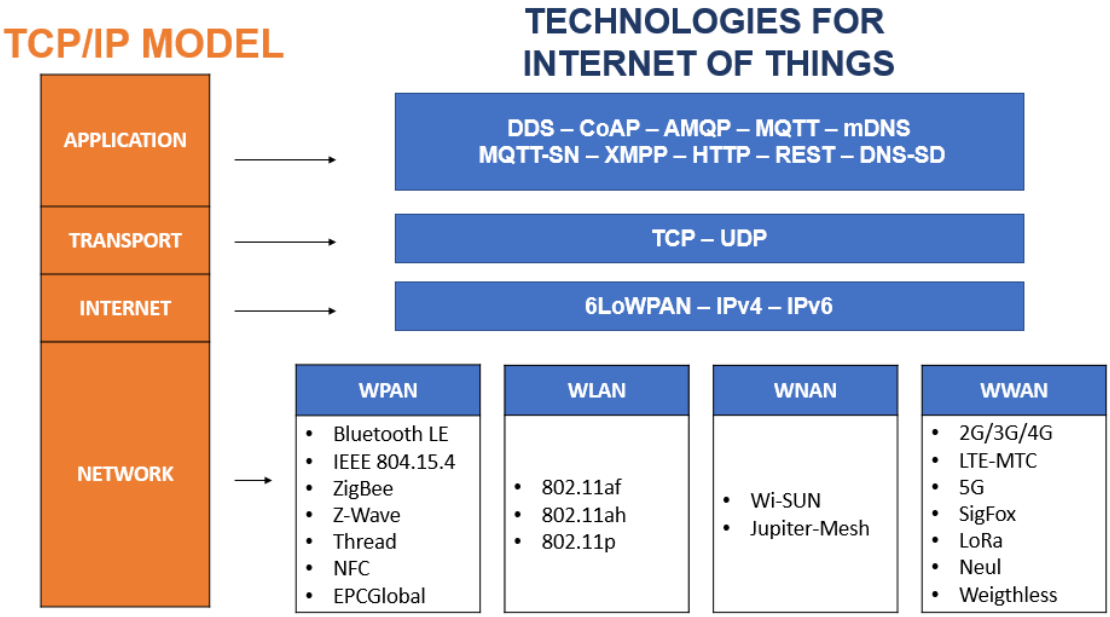

Figure 2: Mapping the TCP/IP Model in IoT Protocols and Technologies.

2.3. Wireless Sensor Network

A WSN is composed of many small, low-cost, low-power, multifunctional sensor nodes; those sensors can produce, and exchange data between them, and send them for processing [7].

The most relevant application fields for WSN are defined as: indoor residential, indoor office, industrial environment, BAN devices [8], military and security [7].

A WSN presents many challenges when compared with traditional computer networks and some of the principal challenges brought by this new paradigm: computational constraints, cost, amount of available RAM/ROM space, among others [9].

The primary constraint on sensors nodes is the low power consumption requirement, this happens given the limited (frequently irreplaceable) power source they carry [7].

To standardize the sensor nodes industrial production, IEEE proposed a technical standard to attend the specificity of the sensor nodes in a WSN, called the IEEE 802.15.4 standard.

The goal of IEEE 802.15.4 is to provide physical and MAC layers for sensor network devices, focusing on long-lived domains that require numerous low-cost nodes [10] with the energy efficiency for the data communication and geolocation [8].

Networks based on IEEE 802.15.4 are often unreliable, because the message could be disrupted by many obstructions, noises, and interferences, generally this kind of network are called Low power and Lossy Networks (LLN) [9].

2.4. Case Study A: WSNs for Home, and Industrial Automation

WSN enables home automation in a setting where each node in the house (constrained device) interacts with a home hub with the capacity to store and process data and act as communication gateway for other networks [11].

In the industrial context, Reduced Function Device (RFD) and WSN are already in use in supply chains, automated production lines, and in production plants. With the continuous adoption of the Industry 4.0 concept, IoT resources will be employed to allow factories to support more flexible production processes, enabling remote monitoring and control, as well as providing mechanisms to collect and analyze the created data. IoT applications are evolving and growing in fields which include healthcare, mining production, transports and logistics, firefighting, and support for the food supply chain.

To verify if the requirements are being fulfilled in an industrial/home automation environment we proposed modeling the WSN enabling IoT for these scenarios. The devices in this model will be based on IEEE 802.15.4 specification, thus networks that cover up to tens of meters (i.e. ZigBee, Z-Wave, BLE, Thread) will

also be modeled and implemented. Ancillary equipment (i.e. routers, access points) to enable WSN to connect to the conventional Internet will also be included in the models, for a closer match to the real world.

2.5. Case Study B: Smart Metering, Smart Cities, and Smart Builds

In Smart Metering/Smart Grids scenarios, both wireline and wireless networks are used to enable communication. These scenarios can be organized in three sub-networks, one related with the customer (i.e. WPAN), other related to sending/receiving measurements, the distribution (i.e. WLAN and WNAN), and finally one related with transmission/generation (i.e. WWAN) [12]. Smart Building scenarios are designed to monitor and maintain the structural health of buildings. In the scope of Smart Cities (e.g. waste management, air quality monitoring) can increase quality and enhance services delivered to the citizens, while at the same time reducing operational costs to the city administration [2].

Those applications of IoT require long-range coverage, with thousands of devices communicating in the same network, but usually exchanging few messages with reduced payload size, since they need to focus on energy efficiency to achieve years of battery duration. By modeling the long-range networks such LPWAs, TV White Space (TVWS) based, and 3GPP solutions, it may be possible to find out if these networks can offer support to the identified needs. Since these technologies will generally be applied in outside environments, with several noise and interruption sources, those should be added to the model and considered upon data analysis.

Chapter 3 Communication Protocols

The technologies there are used above the independent network layer in the TCP/IP model are discussed in this chapter. Many of these technologies are based on standard solutions applied in conventional networks and many of them are modified to attend the demands of the IoT.

3.1. Current Software Solutions

All the discussion in this chapter and in the next one is also described in a concise approach in ANNEX 1 - Application Protocols and Wireless Communication for IoT: A Simulation Case Study Proposal. It was presented at the 11th International Symposium on Communication Systems, Networks, and Digital Signal Processing (CSNDSP 2018), where we receive a travel grant award.

This chapter and the next one explains in detail the functionalities and behavior of the IoT technologies, for those who are not interested in too many details, it is recommendable the reading of annex 1 and then skip to Development chapter.

There are standardized protocols, interfaces, services and architectures that devices must use when they are communicating on networks.

The most used protocol for internet communication in constrained devices is the Constrained Application Protocol protocol, it basically brings Hypertext Transfer Protocol (HTTP) functionalities to a constrained environment.

In a publish/subscribe perspective are presented the DDS protocol, the Message Queuing Telemetry Transport for Sensor Network (MQTT-SN) protocol, Advanced Messaging Queuing Protocol (AMQP), and Extensible Messaging and Presence Protocol (XMPP) Protocol. All these protocols implement publish/subscribe communication, with varies levels of Quality of Service (QoS), network requirements, and efficiency.

The need to find sources is provided by Multicast Domain Name Server (m-DNS) and The Domain Name Server Service Discovery (DNS-SD), these services have DNS as the base and make modifications in it, focusing on providing a solid solution using an already available recourse.

The RPL protocol is the standard routing protocol, using a distance-vector approach, with multiples virtual routing topologies, constructed based on metrics and constraints.

Table 1 offers a short overview of the principal applications protocols. A full explanation of each protocol features, capability, and applications is present in the following sections.

Table 1: Comparison between applications protocols.

Protocol	**Transport**	**QoS options**	**Architecture**	**Security**
CoAP	UDP	✓	Request/Response	DTLS
MQTT-SN	TCP	✓	Publish/Subscribe	TLS/SSL
XMPP	TCP	✗	Request/Response Publish/Subscribe	TLS/SSL
REST	HTTP	✗	Request/Response	HTTPS
AMQP	TCP	✓	Publish/Subscribe	TLS/SSL
DDS	TCP	✓	Publish/Subscribe	TLS/SSL

3.2. Protocols

Protocols are the piece of code that enables good operability of the devices.

3.2.1. Constrained Application Protocol

The CoAP is a web transfer protocol, specialized for constrained nodes and networks, structured for machine-to-machine communication [13].

Some of the principal features that CoAP provides, include request/response interaction model, built-in discovery of services and resources, Uniform Resource Identifier (URIs) and Internet media types, easily interface with HTTP, very low overhead and Datagram Transport Layer Security (DTLS) [13].

The integration between CoAP and HTTP is provided by proxies that can convert single HTTP request into multiple CoAP requests for the nodes in the WSN and vice-versa. Besides, the mapping of the of the URIs is quite straightforward, having nodes that understand HTTP in one side and CoAP in the other [14].

In a machine-to-machine environment, devices must be able to find other ones in the same network. To resolve this requirement CoAP servers are encouraged to provide a resource description, via the well-known URI (/.well-known/), with this URI a device than obtaining access to the clients description with a GET request [14].

HTTP always requires a client-initiated request and if the client wants to stay updated about the service state, it must do the request again and again, which brings problems to the power constraints related with the nodes in the WSN [14].

To fulfill the need of stay updated with the service and attend the power constraint, CoAP introduces an observer design pattern, the node tells to the server that it wants to follow the service state and every time that this state change, the node is notified [14].

3.2.2. Extensible Messaging and Presence Protocol

The XMPP enables a near-real-time exchange of structured yet extensible data between any two or more network entities, using the Extensible Markup Language (XML) [15].

According [16], XML is a subset of the Standard Generalized Markup Language (SGML) and can be defined as documents that contain entities, containing either parsed or unparsed data, thus offering a design and a structure for these documents.

This near-real-time exchange of data is provided by the exchanging of the XML stanzas over the network, wherein an entity needs to connect to a server to enter in the network, for being allowed to exchange stanzas with other entities in the network [15].

XMPP consists in an asynchronous, end-to-end exchanged of Stanzas among a distributed network of globally addressable, presence-aware clients and servers, being similar to email's architecture, with useful modifications to allow the communication in close to real time [15].

The stanzas exchange process can occur in a client-to-client or server-to-server way in the inner or outer network and is defined in [15] as follows:

- Typically based on the resolution of the fully qualified domain name, determines the IP address and port at which to connect;
- Open TCP connection and XML stream;
- Negotiate a Transport Layer Security (TLS) and authenticate with Simple Authentication and Security Layer (SASL);
- Exchange Stanzas close the stream and connection;

3.2.3. Data Distribution Service

DDS is a Data-Centric Publish-Subscribe (DCPS) service for distributed application communication and integration [17] and is based in a broker-less

architecture, using multicast to bring high QoS and reliability, that suits for the real-time constraints for IoT [1].

Its architecture is defined by two layers: DCPS, that is responsible for delivering the information to subscribers and Data-Local Reconstruction Layer (DLRL), responsible for providing interfaces to the DCPS functionalities, although this layer is optional [1].

The abstraction that represents the network and the applications that can communicate with each other is called DomainParticipant and inside it must have five entities: Publisher and the DataWriter, on the sending side; Subscriber and the DataReader on the receiving side; Topic stay between publications and subscriptions [17], relating DataWriters with DataReaders [1] .

The Publisher is explained in [17] as being the entity responsible for the distributions of different types of data and uses a DataWritter to describe that specific typed data.

The Subscriber is responsible for receives published data and make it available to the application and the application will use a specific typed DataReader, to access it [17].

A Topic entity associates a unique name in the domain with a type and a QoS, controlling the behavior of the of the Publisher and the Subscriber [17].

3.2.4. Message Queuing Telemetry Transport for Sensor Network

Using the traditional model of having the device as central in the network does not work well in WSN, many factors can contribute to it, like: devices can change their addresses in function of the time, new devices can be integrated to the network, links are likable to failure, etc. What is important in these cases is the data and with a data-centric communication approach, these problems can be overcome [18][19].

Message Queuing Telemetry Transport (MQTT) is a Publish/Subscribe protocol and a good example of a data-centric communication, but it requires a point-to-point, session-oriented, auto-segmenting data transport service, such TCP/IP, which is not applicable for WSN [19].

In [18] is provided MQTT-SN, also called MQTT-S [19], a protocol that extends MQTT for operation in low-cost and low-power networks.

This protocol aims to connect embedded devices and networks with applications and middleware, it has one-to-one, one-to-many, many-to-many connection mechanism, being considered an optimal connection protocol for IoT and Machine to Machine (M2M) [1].

The protocol groups the devices in the network as follows [18]:

- Subscriber: is interested in receiving the information;
- Publishers: produces the information;
- Broker: coordinates the exchange between subscribers and publishers.

Gateways are a very important piece of the protocol because constrained devices connect to the gateways and then the gateways connect to the Broker, making the mapping of MQTT-SN to MQTT [6] [7] like is illustrated in Figure 3.

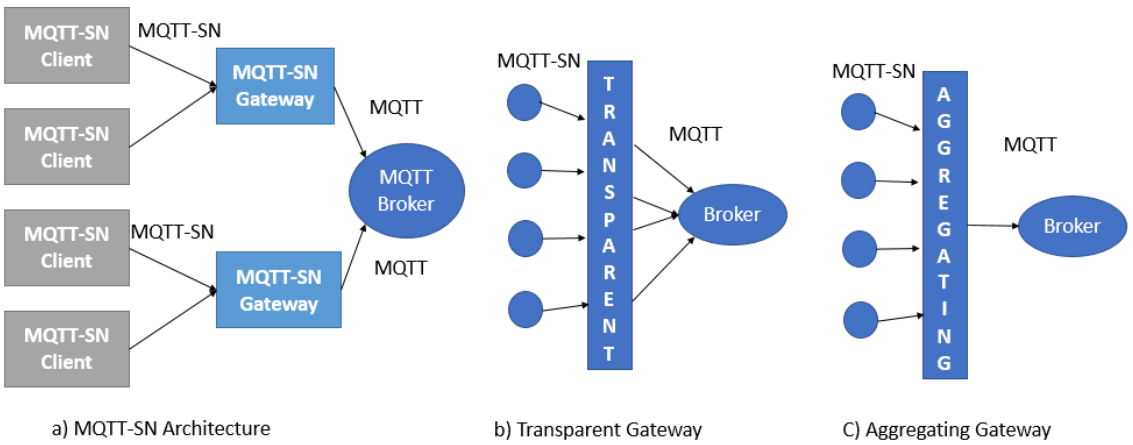

Figure 3: a) MQTT-SN Architecture. b) Transparent Gateway. C) Aggregating Gateway. (Adapted from Stanford-Clark & Truong, 2013).

Two types of gateways are defined for MQTT-SN protocol and according to [18], they can be transparent or aggregating. Transparent gateways offer an end-to-

end connection, performing a translation between client and server; Aggregating gateways have only one connection between many clients and the server [6] [7]. Both are represented in Figure 3.

In [18] is also described the three levels of QoS that MQTT-SN protocol provides:

- Level 0: best-effort, the message is delivered or not, no retransmission or acknowledgment is defined;
- Level 1: the message is retransmitted until the receives acknowledge;
- Level 2: received and only one time;

3.2.5. Advanced Messaging Queuing Protocol

AMQP is an open protocol for business messages, be structured in layers that define type systems and encoding, as a transport layer, message formats layer, interactions layer, defining the behavior between the processes, and the security necessary.

Focusing on a message-oriented environment, it supplies delivery guaranties primitives including at-most-once, at-least-once and exactly once, only requiring reliable protocol, such as TCP, to exchange messages [1].

The lowest level of the protocol defines the type systems and the encoding, these features are needed to support an interoperable data representation, defining a standard set of primitive types, composed types, collections or data structs [20].

In the transport layer is defined frame-oriented connections, sessions, and links, operating between the nodes, abstractions responsible for the safe store and/or delivery of the messages, that exist within a container [20].

Message layer standardizes the message format, delivery state for the messages, state for messages that are stored at a distribution node, between other definition [20].

The transaction layer mostly defines controllers for the communication and the security layer, using TLS or SALS supply an authenticated/encrypted transport for the frames [20].

AMQP besides supplying a peer-to-peer method also gives a publish/subscribe model to communication, as can be seen in Figure 4. A publisher sends to an Exchange, that will be in charge to create a routing path to deliver the content to the right queue [1].

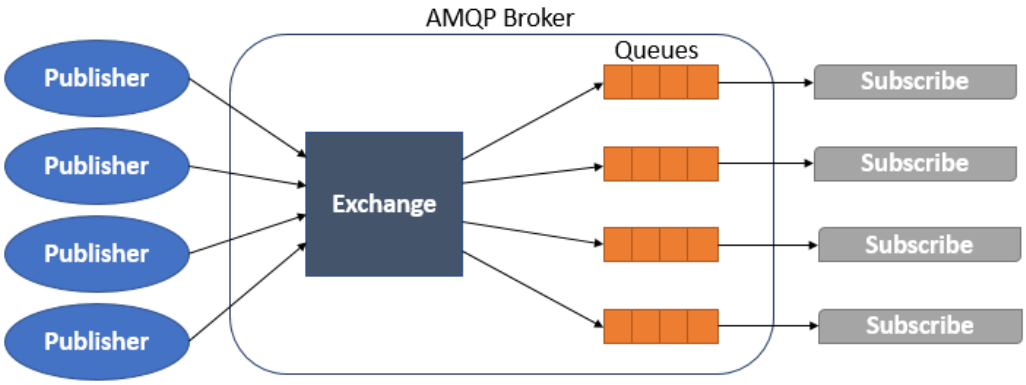

Figure 4: AMQP publish/subscribe method (Adapted from A. Al-Fuqaha et al., 2015).

3.2.6. Hypertext Transfer Protocol

HTTP/2 is the latest version of HTTP protocol, it was formalized by the Internet Engineering Task Force (IETF), the full specification can be accessed by the RFC 7540 and it was released in May 2015.

It was built to enable the more efficient use of network resources and reduce the perception of latency, introducing the header field compression. It also supports multiple concurrent exchanges on the same connection and introduces unsolicited push of representation from servers to clients [21].

HTTP/2 doesn't invalidate HTTP/1.x, that continues to run in the Web. But if a client makes a request without prior knowledge about support for HTTP/2, the next hop uses the HTTP Upgrade mechanism [21].

If the server doesn't support HTTP/2 it will respond the request with no Upgrade header field, however, if it supports, the responding request will be 101 (switch protocols) [21].

As [21] explains HTTP/2 stream have several important characteristics:

- A single HTTP/2 connection contain multiple concurrently open stream, with either endpoint interleaving frames from multiple streams;
- Streams can be established and used unilaterally or shared by either the client or server;
- Streams can be closed by either endpoint;
- The order in which frames are sent on a stream is significant because the recipient process frames in the order they are received;

3.3. Services

The following services implement discovery and registration of resources in a self-configurated, dynamic, and efficient way.

3.3.1. Multicast Domain Name Server

According to [22] mDNS provides the ability to perform DNS-like operations on the local link in the absence of any conventional Unicast DNS server and designs a portion of the namespace to local use without any fee.

The benefits of using this approach of DNS are: little or no configuration to get the network working; networks work with no infrastructure present; Will still work during infrastructure failures [22].

Sets of records can be grouped into two distinct groups [22] as:

- Shared: multiple DNS responders may have records with the same name, type, and class, thus several responders can respond to a single request;

- Unique: the control of a set of records is under control of a single responder, so is expected at least one response.

The portion of the namespace that can be used for any user must have the ".local." suffix and it is a special domain, being only recognized and meaningful in the link-local [22].

The server that implements mDNS must recognize two kinds of queries One-shot and Continuous, their principal differences are explained in [22] as follows:

- One-shot: The regular DNS query, the first answer is listened to and the rest is ignored.
- Continuous: The life-cycle of the request don't end whit the first answer, but only when the client doesn't require it anymore.

3.3.2. Domain Name Server Service Discovery

DNS-SD as described in [23] generally given a service and a domain, it allows clients to discover a list of named instances of that service.

It uses DNS which is a consolidated technology, with many developers and a vast quantity of descriptions about how it works and how to implement it, being able to extend it and offer a good service discovery with minor effort [23].

Also in [23] is defined which abilities a good service discovery must have:

- Service Instance Enumeration: Ability to query services in a logical domain;
- Service Instance Resolution: efficiently resolve the instance name to the information that client needs;
- Should be relatively persistent: the information must last a certain period if the client needs it in this period the service must be there without performing the service instance enumeration again;

To implement the service discovery, according to [23] is required to make some changes in the fields SRV and TEXT of the record, to save information about the

service instance that the client is requesting, but this doesn't affect the other functions of the DNS.

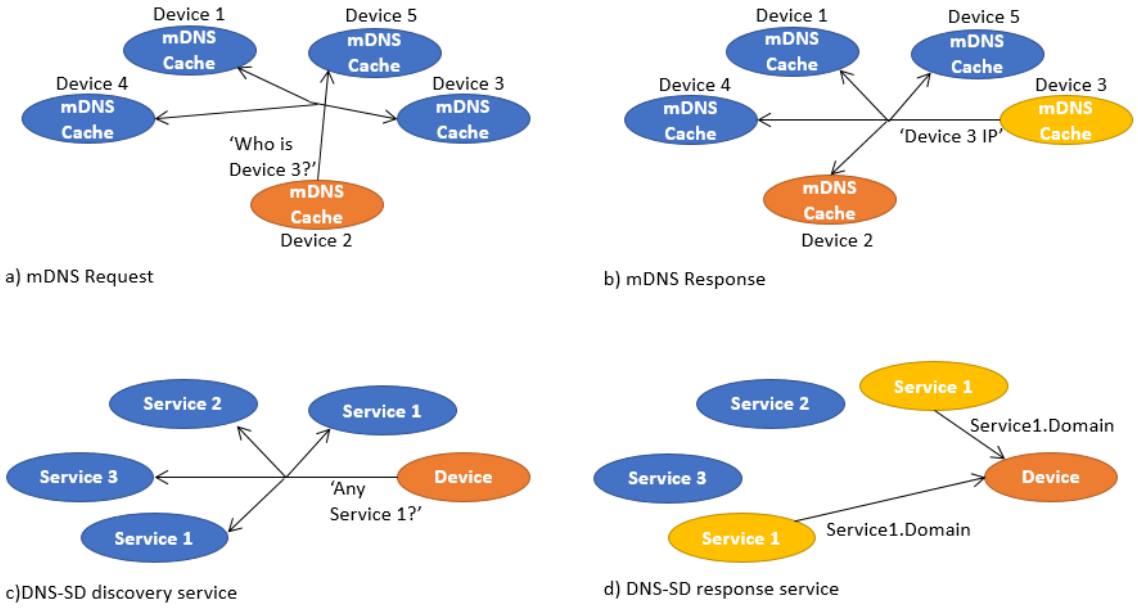

Figure 5: Multicast DNS and DNS Discovery Service request/response.

3.3.3. RPL Routing Protocol

It is a distance vector IPv6 routing protocol for LLN, such as WSN, that specifies how to build a Destination Oriented Directed Acyclic Graph (DODAG), using a set of metrics and constraints [9].

As [9] defines, a DODAG is a logical topology built over the physical network to meet some specific metrics and constraints, also is possible to have many of these in a single physical network, each one to attend some criteria.

Figure 6 illustrates this concept of multiple logical topologies on the same physical network, implemented by RPL.

RPL also tries to avoid loops in the network, which could cause packet drops and link congestion, it has two rules to achieve this loop avoidance: I- A node can't select a neighboring as parent if this neighboring is deeper (has a higher rank); II- a node is not allowed to move deeper in the graph to have more parents [9].

Nodes and link failures can be repaired in two distinct forms: global and local. When a node loses a connection and has no way “up” direction in the graph, it runs a local repair and tries to find another parent, whereas the global repair reboots the whole graph and it can only be done by the root node [9].

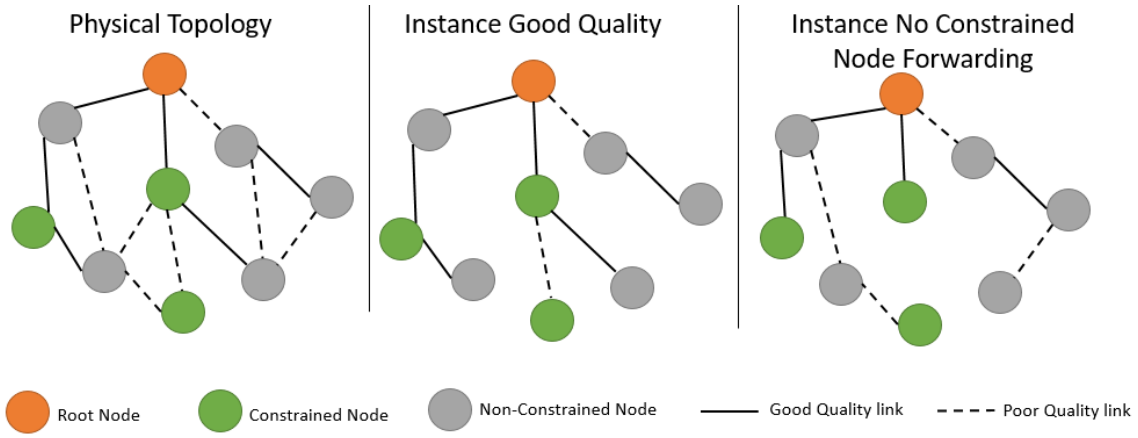

Figure 6: Multiples DODAGs for the same physical topology.

Timer management uses a mechanism called “trickle timer”, where this timer is increased when the network is a stable state, avoiding useless message exchanges; when a node detects an inconsistency this timer is reset and the node warn the others about it [9].

3.4. Representational State Transfer Architecture

Representational State Transfer (REST) is a well establish an architectural style for web applications focusing in large-scale distributed systems, which is based in four principals: Resource Identification through URI; Uniform Interface; Self-descriptive messages; Stateful interactions through hyperlinks, that says that every interaction whit a resource is stateless.

By exposing a set of resources which are used to the interaction with clients, these resources are defined by URI, this way creating the resource identification principal [24].

Uniform interface consists of a set of well-known operations and by decoupling the representation of the resource are possible to access them in many forms, that is the self-descriptive principal.

Its principal strengths are in the fact that all technologies for its implementation (i.e, HTTP, XML, URI) and infrastructure is already being pervasive [24].

3.5. Transport Layer

The transport layer is a conceptual division in the network stack, being responsible for offering host-to-host communication and services for applications, including connection-oriented data stream support, flow control, and multiplexing.

Many protocols can be found in this Layer, such as UDP-Lite, Datagram Congestion Control Protocol (DCCP), Stream Control Transmission Protocol (SCTP), Stream Control Transmission Protocol Partial Reliability Extension (SCTP-PR), but the two most used are UDP and TCP, briefly described below.

3.5.1. User Datagram Packet

UDP is defined to make available a datagram mode of packet-switched communication in a network [25] and provides a minimal, unreliable, best-effort, message-passing transport to applications [26].

This protocol has a set of unique characteristics as not establishing end-to-end communication and not providing inherent congestion control mechanisms [26].

Assuming IP as the underlying protocol [25], each UDP datagram is carried in a single IP packet and its payload is limited to a maximum 65,507 bytes or 65,527 bytes in IPv4 and IPv6, respectively [26].

3.5.2. Transport Control Portocol

TCP is a connection-oriented, end-to-end protocol designed to support multi-network applications, making very few assumptions, mostly requiring a simple IP packet from the lower layer [27].

It also implements flux control, congestion control and has mechanisms to assure a reliable delivery. Used widely in the Web and the protocols that support it, such as HTTP.

3.6. Internet Layer

This layer is responsible for controlling the network operation, it has many responsibilities including routes datagrams between networks and selecting the next-hop host.

In this layer is also attributed an IP-address for the datagrams, specifying the source and the destination. The methods for this addresses assignment is explained next.

In constrained environments, the principal concern of 6LoWPAN is to compress IP and UDP headers, but it also provides packet fragmentation.

3.6.1. IPv4 and IPv6

The IP protocol is the base of the internet, its principal concern being to move datagrams between distinct networks until it finds the destination network.

Addressing and fragmentation are the principal functions presented by this protocol. In the version 4, the addressing process was realized using 32 bits, the most significant bits representing the network, the exact quantity is defined in the network mask, followed by the local address [28]. IP version 6 uses 128 bits for addressing [29]. Fragmentation is used to transport datagrams between networks with different size limitations according to the lower layers restrictions.

Header format simplifications, support for extensions and options, flow labeling, authentication, and privacy capabilities are examples of the improvements made in version 6 [29].

3.6.2. IPv6 over Low power Wireless Personal Area Networks

The constraints presented for the Local Wireless Personal Area Networks LoWPAN created many proprietary protocols and link-only, presuming that TCP/IP family needed too much memory and bandwidth for them to scale it down as necessary [10].

The adaptation layer between network and link layers, defined by 6LoWPAN, has 3 principles, as stated in [10]:

- header compression: performed using link-level or context assumptions, by using the HC1 compression method to compress the IPv6 and UDP headers, where the size of the fully compressed headers are 2 bytes for IPv6 and 4 bytes for UDP;
- fragmentation: multiple datagram transmissions, to accommodate the IPv6 minimum Maximum Transmission Unit (MTU);
- layer-two forwarding: carries the link-level addresses to the end of an IP hop;

As [10] explains, 6LoWPAN protocol puts the information that is needed in other layers in three compressed subheaders, as follows:

- Mesh addressing: used to provide information about the two-layer forwarding;
- Fragmentation: holds the information about MTU requirement;
- Dispatcher: identifies the subheader types;

Chapter 4 Network Communication

In the TCP/IP protocol architecture, the network layer is proposed to be independent of the network access method, frame format, and medium. It is designed to connect different network types. It encompasses the Physical (PHY) and MAC of the OSI model.

Figure 7 illustrates the range and the data rate of the principal network technologies for the internet of things.

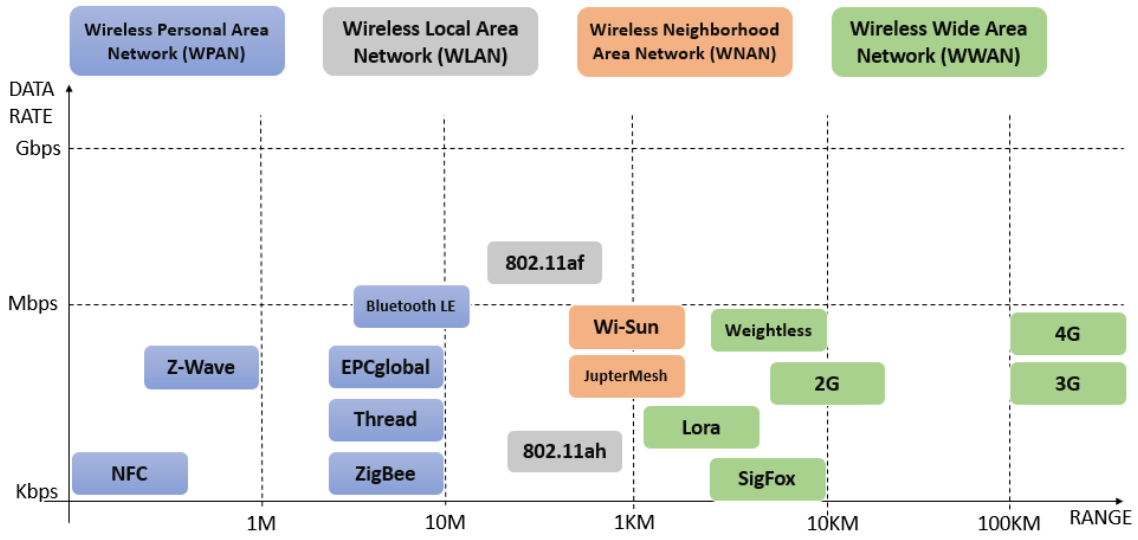

Figure 7: Networks and their data rate and ranges.

4.1. Wireless Personal Area Networks

Wireless Personal Area Networks (WPAN) for the Internet of Things works with a small range, at most 10 meters, and low data rates. Table 2 shows an overview of WPAN for IoT.

Table 2: Personal area networks comparison

Technology	**Band (MHz)**	**Data Rate (Kbps)**	**Scalability**	**Holder**	**MAX Range (m)**	**Standard**
IEEE 802.15.4	868/915/2400	20/40/250	65k nodes	IEEE	10	✓
BLE	2400	1024	5917 slaves	Bluetooth SIG	10	✓
EPCGlobal	860-960	4-640	-	EPCglobal	10	✓
ZigBee	915/2400	250	65k nodes	ZigBee Alliance	10	✓
Z-Wave	868/908/2400	40	232 nodes	Z-Wave Alliance	1	✗
NFC	13.56	424	-	NFC Forum	0,01	✓
Thread	2400	250	300	Thread Group Alliance	30	✗

4.1.1. IEEE 802.15.4

IEEE 804.15.4 is the most prominent standard for Low-Rate Wireless Personal Area Networks (LR-WPAN), it defines both the PHY and MAC layers of the OSI MODEL [30], and delivers a good solution in energy-efficiency, range and data rate [31].

The first release of the protocol was in 2003 and it has been periodically revised (2006, 2011, 2015), being the IEEE 804.15.4-2015 the active standard and the 802.15.4v-2017 the most recent amendment. Table 3 offers an overview of the content of these modifications and their respective years.

In the PHY layer two services are offered: I - PHY data service, responsible for transmitting and receiving Physical Protocol Data Unit (PPDU); II - PHY management service, which manages the requirements from the MAC layer [32].

The standard defines 16 frequency channels, between 2.405 GHz and 2.480 GHz, separated 5 MHz from each other, using a worldwide unlicensed band [31].

These channels are orthogonal, thus one specific channel does not have interference in the other two that are close to it, and can achieve 2Mbps physical data rate [31].

In this kind of networks, the devices are typed as Full-Functional Device (FFD) or Reduced Function Device (RFD).

FFD is a device that is capable of been coordinator for the Personal Area Network [32] or just as a normal node. The coordinator is responsible for the creation, control, and maintenance of the network, they can store a routing table and implement full MAC [1].

RFDs are very limited and do not need to send much information, being associated with a single FFD at each time [32].

Two topologies are defined in IEEE 804.15.4 LR-WPAN: Star or peer-to-peer.

A star topology is characterized when the communication is realized between devices and a single central controller (i.e. Personal Area Network (PAN) Coordinator), is it the primary controller of the PAN [32].

Peer-to-peer topology enables any FFD to communicate with other, as long as they are in the communicational range and an RFD device still only communicate with the PAN coordinator [32].

Table 3: Amendments to the IEEE 802.15.4 standard.

Year	Amendment	Content
2006	IEEE 802.15.4b	Resolve ambiguities, reduce complexity, increase flexibility, newly available frequencies, etc.
2007	IEEE 802.15.4a	Add two additional PHYs using Ultra-WideBand (UWB) and Chirp Spread Spectrum (CSS).
2009	IEEE 802.15.4c	Add new band frequencies for PAN in China
2009	IEEE 802.15.4d	Make modifications in the PHY and MAC layers to support the use of frequency specified in Japan.
2011	IEEE 802.15.4m	PHY/MAC alterations for TV White Space
2011	IEEE 802.15.4p	PHY/MAC alterations for Positive Train Control (PTC)
2011	IEEE 802.15.4e	A new MAC protocol for attending industrial needs.
2012	IEEE 802.15.4f	Make changes in the MKS and LRP UWB PHYs to support autonomous active RFID.
2012	IEEE 802.15.4g	Changes in the PHY to support Smart Utility Networks
2013	IEEE 802.15.4j	Medical Body Area Networks
2013	IEEE 802.15.4k	PHY for Low Energy Critical Infrastructure Monitoring
2015	IEEE 802.15.4q	PHY for Ultra Low Power
2016	IEEE 802.15.4n	PHY/MAC alterations for China medical Band
2016	IEEE 802.15.4u	PHY for India Band
2017	IEEE 802.15.4v	Add multiples Bands in multiples regions of the world.

The first one is the unslotted Carrier-Sense Multiple Access with Collision Avoidance (CSMA-CA) for nonbeacon-enable networks, which consists in random periods for waiting and backoff for the device that wishes to transmit, and the second for beacon-enabled networks is used slotted CSMA-CA, that aligns the backoff period to the PAN coordinator and the start of the beacon transmission [32].

Time Synchronized Channel Hopping (TSCH) is standard MAC protocol since 2010, it is based in the Time Synchronized Mesh Protocol (TSMP) [33], they share the underlying technology and differ in: The packet format and the communication with the higher-layers [31].

TSCH bases are in the slotframe structure, which is a group of slots that repeat over time. For a given slot, the protocol can operate three actions receive, transmit or sleep [31].

This TSCH is a proven technology and it is run in tens of thousands of networks, that combined with IPv6 upper layers (6LoWPAN) is able to fulfill the stringent of IoT industrial networks: low latency, ultra-low jitter and high reliability [34].

The requirements from MAC layer are sent packets with at most 128 bytes, with the first byte indicating how many bytes follow, the others 127 bytes can be arbitrary [31].

4.1.2. ZigBee

Focused on low-data-rate and short-range application ZigBee Alliance proposed the ZigBee Standard. It is composed by four main layers: the PHY layer, MAC layer, Network (NWK) layer and Application (APL) layer, been the PHY and MAC based on IEEE 804.15.4 and a security functionality across NWK and APL layers is also provided [35][36][37].

In a ZigBee network, devices are grouped as coordinator, router, and end-device [36].

The addressing scheme of these devices is based on a random assignment 16-bit address, with conflict resolution mechanism based on IEEE MAC layer [37].

As explained in [35] there are three topologies that are supported in the NWK layer: Star, Tree, and Mesh. In the Star topology all the network is controlled by the ZigBee coordinator (i.e., PAN Coordinator in IEEE 804.15.4 [36]) and all end-devices must connect only to this coordinator. For a tree topology hierarchical routing strategies are applied to move data and control messages and to allow peer-to-peer communication, is defined the mesh topology. Both Tree and Mesh are created by the ZigBee coordinator and can the extended using ZigBee routers.

The NWK layer is composed of the Network Layer Data Entity (NLDE), which is responsible for transport application protocol data units (NPDU), and the Network Layer Management Entity, that allow an application interacts with the rest of the stack [35].

The application layer has many sub-elements that deliver different services. To specify a simple interface with NWK layer the application support sub-layer is used. The Application Framework is the environment where the application objects are stored, to make the interface between the application objects and the device is used the ZigBee Device Object (ZDO), which supply general configuration and initialization [35]. The stack representation is presented in Figure 8.

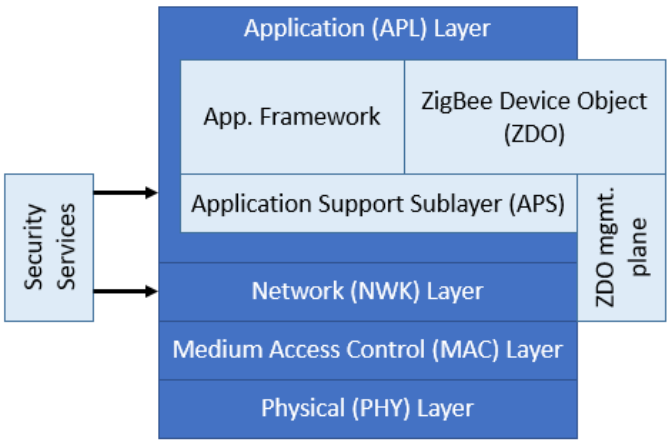

Figure 8: ZigBee Protocol Stack (Adapted from Gomez & Paradells, 2010)

4.1.3. Z-Wave

Z-Wave is a wireless protocol architecture developed by Sigma Designs. Its main propose is allowing reliable transmission of a short message from a control unit to one or more nodes in the network [36] and it should not be used to transfer a large amount of data, streaming or timing critical data [38].

Also, devices in a Z-Wave network can have two types: controllers, that initiates and sends the commands; and the slaves, which can execute or forward these commands [38].

These devices operate in the 900MHz or 2,4GHz band, having 40Kb/s or 200Kb/s data rate, respectively. It uses the Binary Frequency Shift Keying (BFSK) for modulation [36].

The MAC Layer is independent of the RF media (i.e., Radio transmitter), but it requires access to the frame (Z-wave) and implements a collision avoidance method, allowing the transmission, if the channel is free, or making the transmission wait for a random period [36].

The transmission of the data between two nodes is taken care by the Transmission Layer, providing: retransmission, checksum check and acknowledgments [38].

Routing layer has provided by the controllers, which maintain a routing table of the full topology of the network, and try to reach the destination using a direct transmission, if it is not available, the controller put the full path in the frame and send it to the slave nodes to complete the transmission [36].

4.1.4. Near Field Communication

Near Field Communication (NFC) was developed in 2002, by Sony and Philips, been a short-range half-duplex communication protocol, providing easy and secure communication [39].

The ISO 14443, ISO 18092 and FeliCa standard defining the High Frequency (HF) operation band at 13.56MHz of the NFC, supporting 424kbps of data rate, up to 10 centimeters [40].

The NFC protocol works with an active and a passive mode. In the active communication, both devices use its own energy to transmit data, while in the passive communication only the device that initiated the communication use its energy [39].

Devices that support NFC communication can be characterized: NFC mobile, NFC tag and NFC reader [39]. These devices can interact in a peer-to-peer from, using at least two mobiles, or in an active/passive way, using a mobile or a reader (active) and a tag (passive).

Data is transferred between two devices using an Near Field Communication Data Exchange Format (NDEF) message, which is composed of an unlimited number of NDEF records, each one of this records contains a length and a type, to describe its function [40].

NFC also has a mechanism to alternate the wireless technology used to transfer the data, using its touch-to-connect mechanism it can encode the necessary information about the devices and create a secure link between them, using Wi-Fi or Bluetooth, for instance [40].

4.1.5. Bluetooth Low Energy

Bluetooth Low Energy (BLE) is a derivation of classic Bluetooth, focusing on low-power control and monitoring applications, is also developed by the Bluetooth SIG for short-range communications.

Differing from Z-Wave and ZigBee, BLE provides a solution in scenarios where a single-hop solution is required and with implantation similarities from the Bluetooth, that have been widely used, is expected that BLE will be used in billions of devices [41].

The BLE stack resembles the classical Bluetooth, being composed of two main parts. The first part is the Controller, that includes PHY and Link layer, and the second one is the Host, that includes the upper layer functionalities [41]. The BLE stack can be seen in Figure 9.

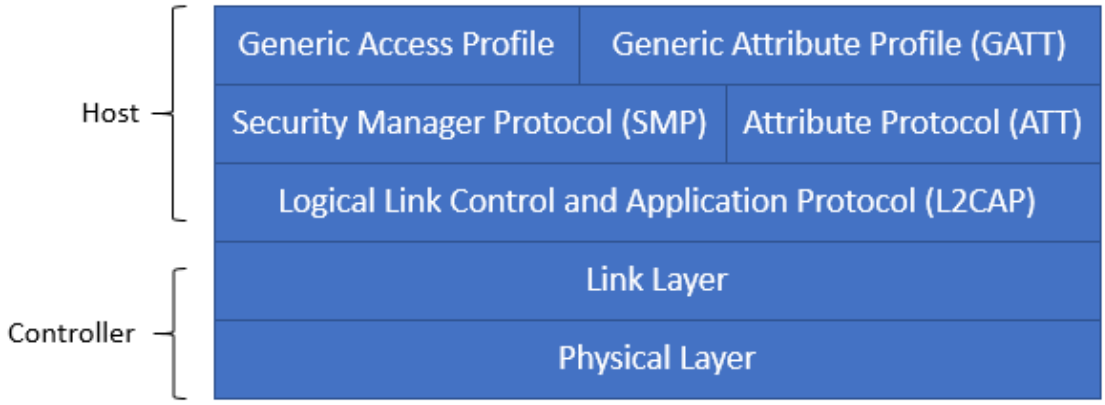

Figure 9: Bluetooth Low Energy protocol stack (Adapted from Gomez, Oller, Paradells, 2012).

The PHY operates in the 2.4GHz ISM and defines 40 RF channels with 2 MHz channel spacing. These channels can be categorized into advertising channels, used for service discovery, broadcast transmissions, and data control, and the data channel, that is used for bidirectional communications [41].

The PHY operates over 1Mbps of data-rate, with Bit Error Rate (BER) of 10^{-3} and all channels use Gaussian Frequency Shift Keying (GFSK) for modulation [41].

In the link layer to establish a connection, a device sends advertising packets, in intervals of time, through the advertising channels, telling it is a connectable device. After, some other devices can listen these adverting and send back a connection request, creating the point-to-point connection [41].

In the connection moment, the devices that send the adverting are called slaves the ones that request the connection are the masters. Masters can handle multiple simultaneous connections with different slaves, however, a slave can have only one connection with one master [41].

To accomplish the energy constraint requirements, BLE implements Time Division Multiple Access (TDMA) protocol, which tells to slaves when they must turn on its radios and listen to the master [41].

When the connection is established the device PHY channels start to work in function of time units called connection events. The connection event is started by the master, being the slave obliged to answer, and all packets are sent at the same channel during this event. Exists a minimum period of 150us between the end of an event and the start of a next one [41].

As a mechanism to flow control is the stop-and-wait, which is implemented with a field in the header, used for holds the identification of the next packaged to be received [41].

The Logical Link Control and Adaptation Protocol (L2CAP) is used for optimization of the resources to/from the above layers, handling the data in a best-effort approach which implies in not use retransmission of flow control [41].

Attribute Protocol (ATT) stores attribute for communication having devices playing server role and client role, Generic Access Profile (GATT) uses ATT to discovery services and exchange of device’s characteristics [41].

As concern to security BLE implements two modes whit different levels of requirements, as can be seen in Table 4.

GAP is the most up level in the stack and defines four behaviors for a BLE device, that can be: I) Broadcaster, only broadcast data and does not support connections with others; II) Observer, receives the data from the Broadcaster; III) Central, initiate and handle multiple connections; IV) Peripheral, have only one connection with the Central [41].

Table 4: Securities modes on BLE

Mode	Layer	Level	Integrity	Pairing	Encryption
1	Link	1	✗	No	✗
		2	✓	Unauthenticated	✓
		3	✓	Authenticated	✓
2	ATT	1	✓	Unauthenticated	✗
		2	✓	Authenticated	✗

4.1.6. Thread

Thread standard is based on IEEE 804.15.4-2006 PHY and MAC layers and provides reliable, cost-effective, low power, wireless device-to-device communication [42].

Devices in this network, are categorized in [42] as:

- Border router: realize the gateway between Thread and others network (Wi-Fi, Ethernet, etc.);
- Routers: Realize the routing service and provide security services for devices trying to enter the network;
- Router-Eligible End Devices: Can become a router if the network needs;
- Sleepy End Devices are host devices, can communicate only with their parent.

In the security context, the standard uses TSL and DTLS with wide know keys and handshakes.

4.1.7. Electronic Product Code

Electronic Product Code (EPC) is a replacement for barcode because it can carry a larger amount of information and can be electronic read over distances up 10 m, even when not visible [43].

This technology was created in the Auto-ID Center at the Massachusetts Institute of Technology (MIT) in 1999 and in 2003 the EPCglobal was created to commercialize it [43].

The EPC protocol defines physical and logical layers for a passive-backscatter, interrogator-talks-first (ITF), RFID system, that operates in the 860MHz – 920MHz frequency range [44].

This system is implemented with two distinct class of devices: The system compromises Interrogator (or reader) and Tags (Labels or Transponders).

The Reader is responsible for transmits the information and operative energy to the Tags using an RF signal in 860MHz – 920 MHz frequency range and for receiving information it must send a Continuous Wave (CW) RF signal in the same frequency [44].

Tags are responsible for modulation, the reflection coefficient of its antenna and backscattering the information for the Readers [44].

The communication process is illustrated in Figure 10.

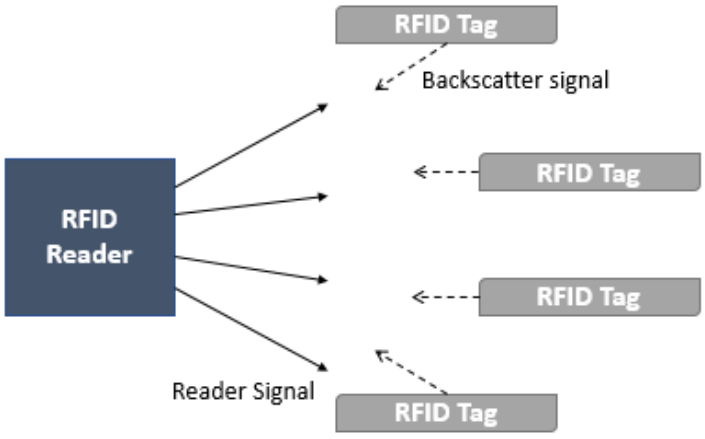

Figure 10: Communication process of EPC (Adapted from Tanenbaum & Wetherall, 2011).

4.2. Wireless Local and Neighborhood Area Networks

Many local, neighborhood and wide area networks use the sub 1GHz frequency spectrum, this frequency has different regulation in each country, some key regions sub 1GHz frequency are presented in Figure 11.

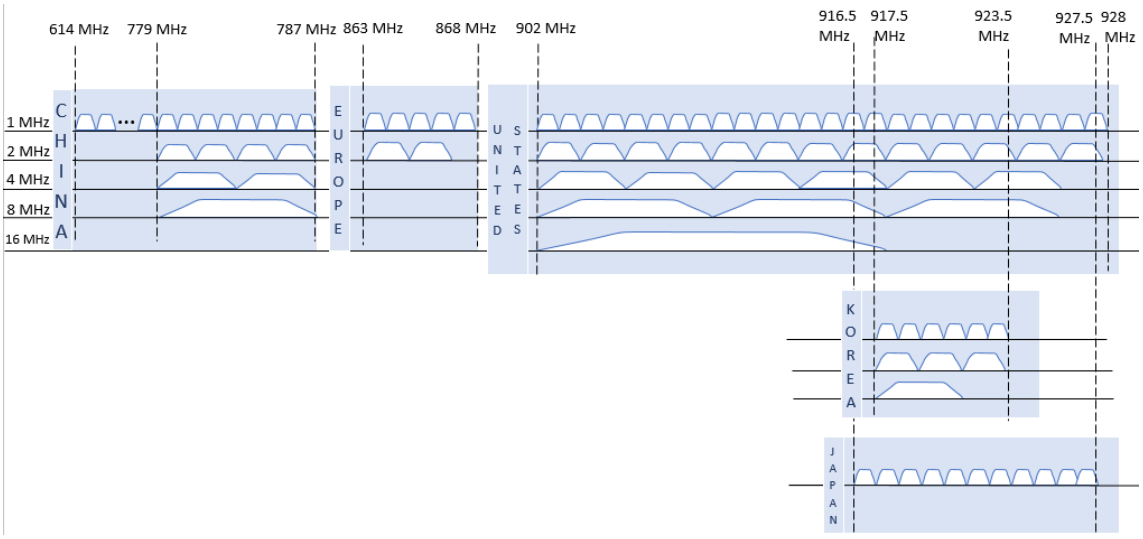

Figure 11: Sub 1GHz frequency spectrum in key regions (Adapted from Park, 2015)

4.2.1. IEEE 802.11af

The IEEE 802.11af is an amendment of the IEEE 802.11 and enables it to share the TV white space spectrum in the very high frequency (VHF) and the ultra-high frequency (UHF) bands between 54 MHz and 790 MHz [45]. The technical aspects of the PHY and MAC are given in Table 5.

The attractive characteristics of TVWS are in the ability to penetrate through walls and other obstacles much more effectively than 2.4 and 5.7 ISM bands, also by an increase in the coverage range and less power to transmit it [46].

Table 5: Technical parameters for MAC and PHY in IEEE802.11af (Adapted from Hu et. al., 2015)

Parameters	**Values**
Modulation Mechanism	Orthogonal Frequency Division Multiplexing (OFDM)
Sampling rate	5MHZ
Fast Fourier Transform (FFT) size	64
Subcarrier Spacing	78.125 kHz
FFT period	12.8 μs
Guard Interval	3.2
Modulation	Binary Phase Shift Keying (BPSK), QPSK, 16-QAM, 64-QAM
Data Rate (Mbps)	1.8, 3.6, 5.5, 7.3, 10.9, 14.5, 16.3, 18.1
Coding Rate	1/2, 2/3, 3/4, 5/6
Receiver Sensitivity (dBm)	-88, -85, -83, -80, -76, -72, -71, -70
Slot Time	21 μs
Short Interframe Space (SIFS)	64 μs
Contention Windows (CW)	[15, 1023]

The architecture of the standard is defined as a set of components, that can be seen in Figure 12, is described in [47] as follows:

- A Geolocation Database (GDB) is the primary element of the standard, and it stores, by geographic location, the permissible frequencies and operating parameters for White Space Device (WSD);
- The Registered Location Secure Server (RLSS) operates as a local database that contains the geographic location and operating parameters for a small number of basic service sets, distributing the permitted operation parameters to the access points and the stations;

- Geolocation-Database-Dependent (GDD) Entities are operated under the control of an authorized GDB and are categorized into two categories:
    - GDD-Enabling Station (i.e. Access Point) requests a white space map to the GDB and with this information, it has authority to enable and control dependent stations;
    - GDD-Dependent Station is controlled by a GDD-Enabling Station, that passes the parameters of a valid whitespace map.

The communication protocol in this network is provided by the Registered Location Query Protocol (RLQP), enabling the stations to select spectrum, power, and bandwidth allowed by their regulation domain [47].

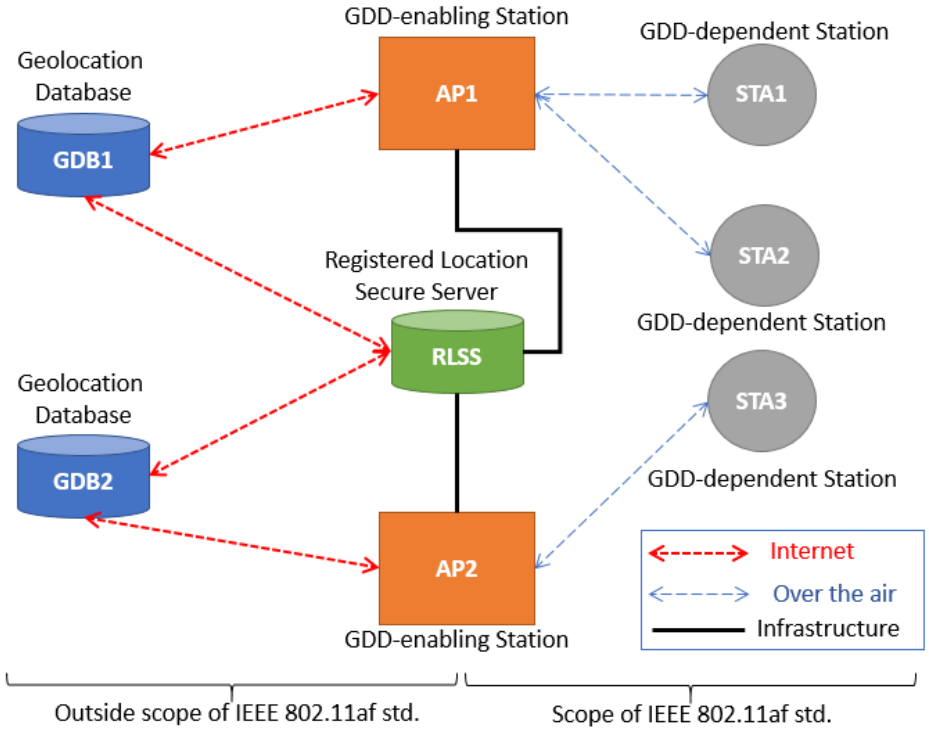

Figure 12: TVWS network including all architecture entities from IEEE802.11af (Adapted from Flores et.al., 2013)

4.2.2. IEEE 802.11ah

In 2010 IEEE formed a group to extend the IEEE 802.11 standard focusing tree use cases: sensor and meters, backhaul sensor and meter data, and extended range WiFi [48] [49].

It operates in the sub 1 GHz band and provides efficient power save strategies, minimum 100 kbps of data rate with up to 1 km of range in outdoor areas [46].

Table 6: PHY and MAC features in IEEE 802.11ah (Adapted from Park, 2015)

Layer	Feature	Description
PHY	Channel bandwidths	1 MHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz
	Modulation schemes	BPSK, QPSK, 16-QAM, 64-QAM, 256-QAM
	Code rates	1/2 with 2 times repetition, 1/2, 2/3, 3/4, 5/6 in either convolutional or low-density parity check (LDPC)
	Maximum number of spatial streams	Four spatial streams
	Data rates	150 kb/s to 347 Mb/s
MAC	A large number of devices	RAW, synchronization frame, hierarchical TIM
	Support for energy-efficient communications	Target Wake Time (TWT), Null Data Packet (NDP), short MAC frame, increased sleep time

4.2.3. Wireless Smart Utility Network

The Wireless Smart Utility Network (Wi-SUN) Alliance is an industrial group that aims to establish and promote standards based on IEEE 802.15.4

The Wi-SUN standard uses IEEE 802.15.4g specification for PHY and IEEE 802.15.4e for MAC, above those layers, are used different protocols depending on the equipment and the application field [50], Figure 13 shows the Wi-SUN standard stack.

Focusing in smart meters, it provides many application profiles such ECHONET Profile (i.e. ECHONET Lite Protocol), JUTA Profile (i.e. U-Bus Air standard), FAN Profile (i.e. multihop communication for N and S American Smart electricity),

RLMM Profile (i.e. communications profiles for sensors networks, and healthcare) [50].

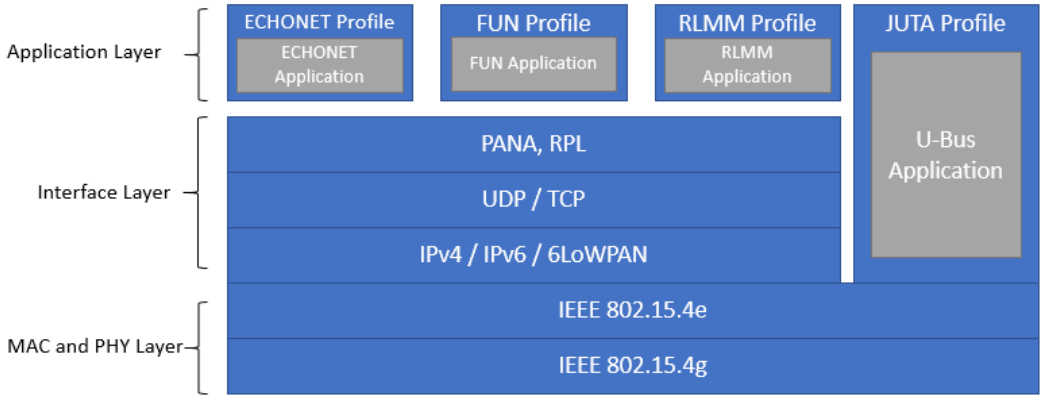

Figure 13: Wi-SUN Standard Stack (Adapted from Kato, 2016)

4.2.4. JupiterMesh

JupiterMesh is a robust, low-power wireless mesh network for neighborhood area communications, provided by the ZigBee Alliance. it is built upon IEEE and IETF standards having advanced technologies such as IPv6, frequency hopping, multi-band operation, authentication, encryption and many others [51].

JupiterMesh is focused on smart metering and smart grid and smart cities fields, having high data rates, up to 800Kbps.

The protocol is based on the IEEE 802.15.4g and IEEE 802.15.4e for PHY and MAC, respectively. For the network layer, it implements IPv6, 6LoWPAN, UDP or TCP, RPL, COAP and many other technologies for multicast and mesh features.

It operates over the sub-1GHz and 2.4GHz ISM bands, having FSK, O-QPSK, and OFMD modulation types to ensure global regulatory compliance and acceptance.

4.3. Wireless Wide Area Networks

These networks are mainly projected to support a large number of devices, communicating in low data rates, with large time spaces between communications, and efficient power management. Table 7 provides an overview of the most used solutions in wide area networks for IoT.

Table 7: Comparison between wide area networks.

Technology	Band (MHz)	DataRate (Kbps)	Scalability	Holder	Classification	Standard
LTE-A	Varies	1G	-	3GPP	WWAN	✓
SigFox	868/915	0.1	100	SIGFOX Network Operators	LPWA	✗
Lora	868/915	0.37-27	6	Semtech Corporation	LPWA	✗
Weightless – N	Sub-GHz	30-100	-	WEIGHTLESS SIG	LPWA	✗
Weightless - W	Sub-GHz	0.2-100	-	WEIGHTLESS SIG	LPWA	✗
Weightless – P	470~790	1-10000	-	WEIGHTLESS SIG	LPWA	✓
Ingenu	2400	0.06-30	1000	Ingenu	LPWA	✗

4.3.1. LTE-A

4G technologies (i.e. LTE and LTE-A) are interesting to IoT communications, meeting demand and scalability requirements, but the costs involving a 4G device still is a problem [34].

An LTE-A network is formed by two parts: the core network, which has overall control of devices and IP packet flow; the second part is the radio access network, this one is responsible for wireless communication and radio access and consists in based stations, referred as evolved Node-Bs (eNBs) [52].

Communication between the eNBs are made via X2 interfaces and the connection with the core network is given through S1 interface [52].

For IoT applications, 3GPP defines new categories of user equipment (UE): CAT-0, CAT-M, and CAT-N. These new categories all have just one antenna, implement half-duplex mode, reduce the complexity of the UE, and promote battery longevity [53]. Other specification can be seen in Table 8.

Table 8: UE in LTE-A for IoT

Specification	**CAT-0**	**CAT-M**	**CAT-N**
Downlink peak rate (Mbps)	1	1	0.2
Uplink peak rate (Mbps)	1	1	0.144
Receive Bandwidth (MHz)	23	23	20

4.3.2. LoRaWAN

LoRa is a PHY proprietary scheme developed by Semtech Corporation and uses CSS type modulation and allows multiple data rates, having the bandwidth and spreading factor to be configurable [54]. Its MAC and above layers are open source and is standardized by LoRa Alliance Consortium[55], under the LoRaWAN Specification.

In summary, LoRaWAN is an LPWAN that offers a fully bidirectional symmetrical link between endpoint and gateway, where this endpoint are projected to operate up 10 years in a regional, national or global deployments [55].

LoRaWAN defines two types of networks, the private and the public, with tree classes of devices, that can co-exist and are defined in [56] as:

- Class A: These devices only enter in the reception mode after they already have transmitted, for optimizing the power consumption, obligatory for all devices;
- Class B: Gateways that broadcast a synchronized beacon providing a time reference for the end-devices open their reception mode;
- Class C: Devices that are mainly powered and are listening to the network all the time.

Topology for a LoRaWAN is defined by [57] as a “star of stars”, meaning that groups of end-devices are connected to gateways via LoRa network and the gateways are connected to a remote server via IP network.

Lora implements the Adaptive Data Rate (ADR) scheme and can manage end-devices rates, like bandwidth occupation, spreading factor and RF output power [58], aiming to maximize the battery lifetime and permitting reduce the time occupancy [59].

This standard uses symmetric-key cryptography to end devices authentication, to protect and to provide privacy to the data in the network [54].

4.3.3. Ingenu

Ingenu consists of a proprietary LPWAN technology, that works in the ISM 2.4 GHz band, it modulation is based on the Random Phase Multi Access (RPMA). This protocol delivers higher speed than others used in LRWAN, 624Kbps in uplink and 156Kbps downlink, being this it principal feature [60], having until 1000 simultaneous users [59].

RPMA is based on the Direct-Sequence Spread Spectrum (DSSS) and has uplink and downlink performed in a half-duplex way, with intercalated periods of 2s between them, allowing the dynamic adaptation of the spreading factor observing the channels conditions [59].

4.3.4. Weightless SIG

Weightless Special Interest Group proposed three open LPWA standards, all operating in license-free or in licensed spectrum, offering different features, the power consumption of the devices and range of the network.

Weightless-W uses the TV white spaces (470 – 790 MHz) to give a solution that poses bidirectional communication, based on FDMA [60], several modulation scheme (16-QAM, DBPSK) and a vast range of spreading factors, adapting dynamically the rate and the range to the network needs [59], achieving data rate between 1Kbps and 1Mbps [54], having battery autonomy between 3 and 5 years.

Weightless-N is a project is designed to expand the range Weightless-W and reduce power consumption, up to 10 years of battery autonomy. Being a UNB standard for only one-way communication from end devices to a base station [54], using DBPSK digital modulation, performing transmission in the sub-GHz ISM bands, achieving until 5kms of range with a data rate between 30 Kbps and 100Kbps [59].

Weightless-P offers two non-proprietary PHY with two-way connectivity. It permits to enhance the reliability by using acknowledgment protocols [59], also uses a well-known scheme to modulate the signals GMSK and QPSK with non-proprietary chipset and data rate between 0.2 Kbps and 100 Kbps [54].

4.3.5. SigFox

SigFox is a proprietary UNB solution, operating in 869MHz (Europe) and 915MHz (North America) bands. It is based on Random Frequency and Time Division Multiple Access (RFTDMA) and achieves 100 bps for uplink, 12 bytes of payload and at devices can’t exceed 140 messages in a day [60].

PHY is based on BPSK modulation and by using UNB achieving efficiently use of the bandwidth, having low noise levels which result in ultra-low power consumption, high receiver sensitivity and inexpensive antenna design [54].

MAC is provided by RFTDMA protocol, that is an Alora-based protocol without sensing the channel occupancy and it has some benefits, as described in [59]: No energy consumption for medium sensing, no need for time synchronization and no constraint on the oscillator precision.

The network architecture is based on end-devices communicating to a Sigfox Network Operator, who is proprietary and are equipped with cognitive software-defined radios, connected with back-end server via an IP connection [54].

Chapter 5 Simulation Models and tools for IoT

A Discrete Event Simulation (DES) is one of the many simulation paradigms that have been proposed. It consists of a model, that Is composed by variables that represent the state of the simulation, and the evolution, that is the sequence of events that are processed in a chronological order [61].

In a monolithic simulation approach, a single Physical Execution Unit (PEU) is responsible for manage the whole simulation, that means process all the events in the correct order, ordinating new events, and updating dependencies [3].

This approach has one big problem, its low scalable. For instance, in a simulation that requires thousands of elements in the model, a single PEU take too much time to process all the events.

The Parallel Discrete Event Simulation (PDES) tries to resolve this scalability problem partitioning the simulation model, processing each part in interconnect PEUs, increasing the model complexity and initial setup. In this approach is required synchronization algorithms to ensure the correct order of the events [3].

Parallel and Distributed Simulation (PADS) is a method where the simulation runs in different processors or hosts, that share memory or are connected by a network.

This method gives many benefits including higher execution speed, scalability, interoperability. But, it also creates some drawbacks such: complex partitioning, synchronizations problems and required data distribution management [61].

5.1. Simulation Tools

Are presented tools that enable simulation in software.

5.2. OMNeT++ - A Discrete Event Simulator

OMNeT++ is an object-oriented discrete events network simulator [5][4] and it is based on the C++ language and totally open source.

It can be used for a wide variety of purposes such as modeling wired and wireless communication networks, queueing networks, multiprocessor, and distributed hardware systems, protocols; validate hardware architectures; evaluate the performance of complex software systems. In general, it's useful for any problem that could be modeled by discrete events and that can be mapped in entities that exchange messages for communication [5].

OMNeT++ has a framework approach, this means that it doesn't provide simulation components of any kind, instead, it provides the basic tools for writing the discrete simulations [62].

The base for the infrastructure provide by the OMNeT++ are modules that communicate by passing messages and there are two types of modules: simple and compound. A simple module is a C++ class that implements specifics task and a group of two or more simple modules is a compound module [62].

Modules can have parameters, mainly for configuration representation or topology [5].

A compound module can have in its inside others compound modules, making in this way, an unlimited level hierarchy. The modules, simple or compound, can only communicate with others in the same level, owing to constraints implemented in the simulation kernel [5].

These modules can communicate with each other by a gate or directly to the destination module. A gate is the input or output interface for the module, used for getting or sending messages, respectively. The communication between two modules occurs only by messages, which can contain data [62].

Those gates interfaces share a connection, that represents the physical way that these modules are going to communicate. Also, is possible to add some specifics parameters in this connection, like delay or speed. A connection with parameters is called channel and can be reused in several other places [5].

Figure 14 illustrates the basic components for a simulation in OMNeT++. Where the thick border boxes represent simple modules and the compound is represented by the thin border. The small blue boxes represent the gates and the connection is represented by the arrows and the lines.

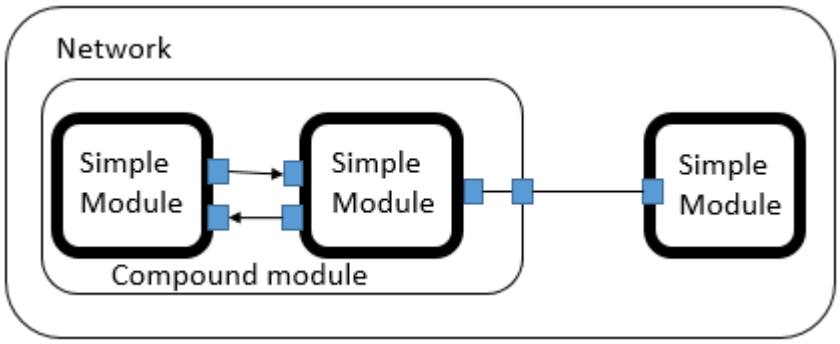

Figure 14: Representation of OMNeT++ components (Adapted from Vargas & Hornig, 2008).

5.2.1. The Separation of Model, Implementation, and Inputs

OMNeT++ works with the concept that every simulation can be divided into three distinct parts:

- Model: This part is represented by the network description language (NED), that specifies the topology and the parameters of it [62];
- Behavior: The actions that the modules will have during the simulation and those actions are implemented with C++ in the simple modules [62];

- Parameters: The input parameters needed for the simulation start to run. As the goal of a simulation is to get results changing the parameters, all they are listed in the INI files. So, it's possible to have many INI files, with many combinations of the parameters as possible [62].

5.2.2. Architecture and Parallelism

The Model Component Library consists of the compiled code from simple and compound modules. When the modules are instantiated, the concrete model is created by the simulation kernel at the beginning of the simulation [62].

After building the simulation, it's executed in the user interface libraries (ENVIR, CMDENV, TKENV). In any of this environment, is defined: where input data come from; where the results will be stored; it controls the debugging output from the simulation and it controls the simulation executions [62].

It's also possible to remove the user interface libraries and customize the environment where the simulation runs, and even embed the OMNeT++ into a larger application, as Figure 15 presents.

This feature is possible because the Simulation Kernel has generic interfaces for communication with the user interface libraries [62].

For very large simulations the OMNeT++ has a parallel execution support, which is useful for speedup or for distributing memory requirements [62].

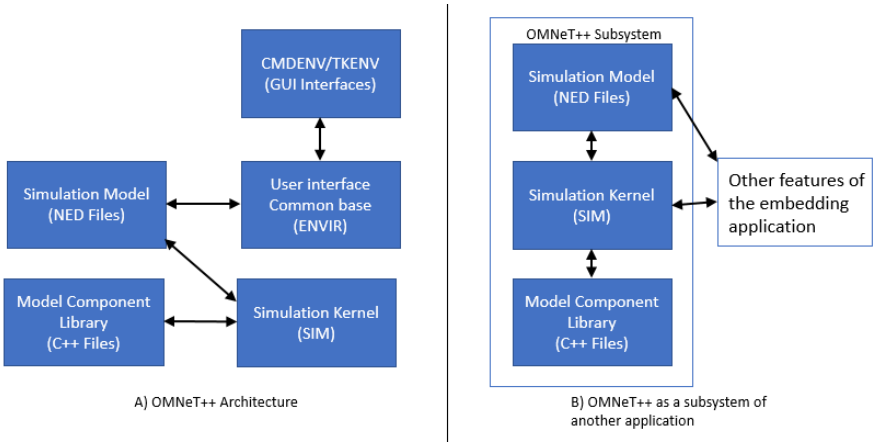

Figure 15: Architecture and embedding applications (adapted from Vargas & Hornig, 2008).

5.2.3. Network Description Language – NED

The network description language (NED) is the way that the topologies and the models are described in the OMNeT++. The language has many features to improve the development of a simulation, such as Inheritance; Interfaces; Packets; Inner types and Metadata annotations [62].

The inheritance, interfaces and inner types are powerful tools of the language that allows more generic and reusable definitions of networks topologies, parameters, and configurations [62].

Packets are ways to simplify and avoid errors caused by classes, with the same name, that is in many different directories. The implementation of the packets in the language is based on the JAVA programming language [62] [-1].

For representing information that could be useful to other tools, to the runtime environment and even for the modules itself, the language allows the metadata annotations.

5.2.4. Integrated Development Environment – IDE

OMNeT++ has an IDE that is a Graphical User Interface (GUI) tool which works with NED files [4].

This IDE offer, with the help of the NED language, the opportunity of creating parametrical and hand-written topologies [62].

5.2.5. Running and Visualizing the Simulation

With the model and the implementation of the modules ready, the simulation can be run. To graphically see the modules and the exchange of messages between them, OMNeT++ have Tkenv.

Tkenv is divided into three methods: automatic animation, module output windows and object inspector [62].

The method of automatic animation is responsible for represent the flow of the exchanged messages between the modules and inform the state change of them [62].

To facilitate the process of debugging the simulation, the method module output window shows debug output messages, showing the trace line of the simulation [62].

It is also possible to follow and modify the state of every object used in the running simulation, without writing any additional code, thanks to the object inspector [62].

5.2.6. Analyzing the Results of the Simulation

The software also offers a graphical analyzer tool for the data resulted from the simulation process. The result of the simulation is stored in several files, which allows many forms of visualization of these data [62].

The analyzer also accepts parameters, for a better understanding of the simulation, creating a filter. Therefore, is possible to select data from certain modules, a determined period of the simulation, hand-written parameters in the modules, etc... [5].

It's also possible to save a "recipe" for the filter, meaning that is not necessary to redo the configuration every time that the simulation is started and the results are changed [62].

Plotting the results is an easy task, with or without filters, there are built-in tools that make this. It is also possible to configure a chart to be generated, like subtitles, forms, colors, background color, name and much more.

5.3. INET Framework

To minimize the amount of effort for build networks in the OMNeT++, it was created an extension to the software that contains many useful components and protocols, including IPv4, IPv6, TPC, UDP, SCTP. This is the INET Framework.

Its structure is the same as the OMNeT++, being based on simple and compound modules and being modeled by the NED Language.

The base components in the INET are host and router models, defined in [63] as formed by:

- Interface Table: Represents the network physical interface abstraction, like Eth0 or Wlan0;
- Routing Table: Encapsulates the behavior for an IPv4 routing, for instance, it adds, removes, enumerates routes; finds the best matching route for a given destination using the Network Layer module. It's also ready for IPv6 routing mode;
- Mobility module: Responsible for implement mobility factors in the simulation, like some random walk. This module is needed in wireless simulation, even if the target is stationary;
- Network Interface Controller (NIC): Represents the controller hardware, it's composed of a queue and a MAC module or a radio module, for wired and wireless simulations, respectively;
- Network Layer: Has the protocols for the network layer, such IPv4, Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP);
- Transport Layer Protocols: Are represented by modules connect to the network layer and has the UDP, TCP and SCTP protocols implemented;
- Applications: Modules that connects to TCP or/and UDP, used for model the behavior of the user application. These applications can be configurated in the initialization files of the OMNeT++;
- Routing Protocols: Modules that implements routing protocols like Open Shortest Path Frist (OSPF) or Border Gateway Protocol (BGP);

- Multiprotocol Label Switching: Needed for simulations that use labeled packets;

The framework extends the NED language for his purposes, adding two new property tags, @node and @label [63].

Compound modules that implement network devices (host, routers, switches, access points, etc.) conventionally have the @node property [63].

The @label property can be added only to modules and gates. This will improve the graphical editor, provide better editing experience [63].

5.4. Castalia Simulator

Castalia Simulator extends the functionalities of the OMNeT++ focused on WSN and BAN environments.

Chapter 6 Development

In this chapter, the steps needed to create the testbed are described, and the drawbacks found in this process. The role of each tool in the simulation, and its configuration, parameters, and outputs are also described.

6.1. Proposed Conceptual Topology

The topology in Figure 16 is proposed to provide a reasonable representation of the environment where the IoT technologies will be applied.

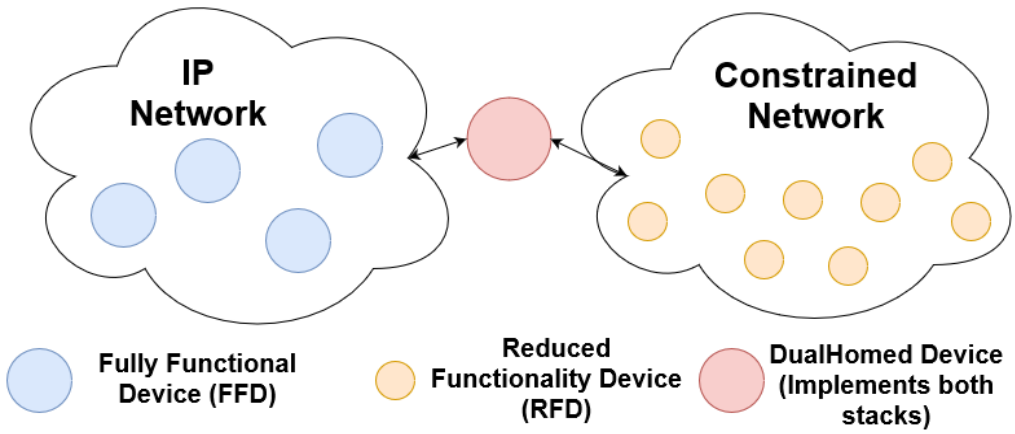

Figure 16: Proposed Network Topology

The IP Network cloud represents the network infrastructure available for general devices (e.g. Wifi, Ethernet, the Internet) and the Constrained Network cloud represents the network specific to IoT devices (with reduced functionality / additional constraints).

As those networks are conceptually different, and no standard solution can readily be applied to support both, is necessary to create a bridge device, which can communicate on both networks and therefore provide an interface.

In this model some simplifications should be made, aiming to not over complicate models and implementations that are already complex. The information that must be exchanged between these networks is displayed in Figure 17, being:

- Simulation Time: a double value that represents the clocks of each simulation (and is used to synchronize them);
- Source and Destination: Representation of the device's address;
- Payload: Data that is being transported.

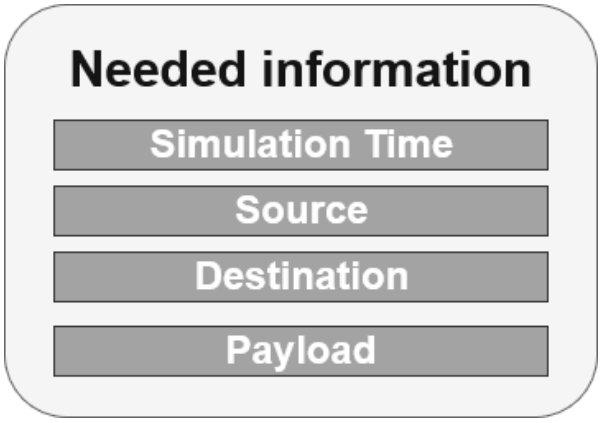

Figure 17: Information exchange between the networks.

6.2. The First Idea: The Standalone IEEE802.15.4 Solution

When searching for models of IEEE802.15.4 already implemented, one of the first found was the implementation of Michael Kirsche[1], and his solution gives a good model of the features present in the IEEE802.15.4 standard.

However, when analyzing the code, it was to perceive that It did not offer a complex simulation environment and data collection features were also absent.

[1] https://github.com/michaelkirsche/IEEE802154INET-Standalone

6.3. First Methodology: One Project Extending both INET and Castalia Features

The last available version of the Castalia Simulator is supported by OMNeT 4.6, which was released early in 2014. The INET 2.6 framework, which was available in early 2015 also was installed in the development environment, since it is compatible both with OMNET and Castalia.

Having in mind the flexibility of the proposed simulation testbed, a new project was created which would include both Castalia and INET as linked frameworks. The goal of this approach was leaving both implementations untouched, aiming not to incorporate into the code inconsistencies, errors or any kind of new misfunctions.

With Castalia providing all communication features to IoT environments and INET supplying IP communication, the main goal of this step was to create the Dual-Homed device. It would be responsible to provide the gateway between the frameworks and the conceptual model and can be seen in Figure 18.

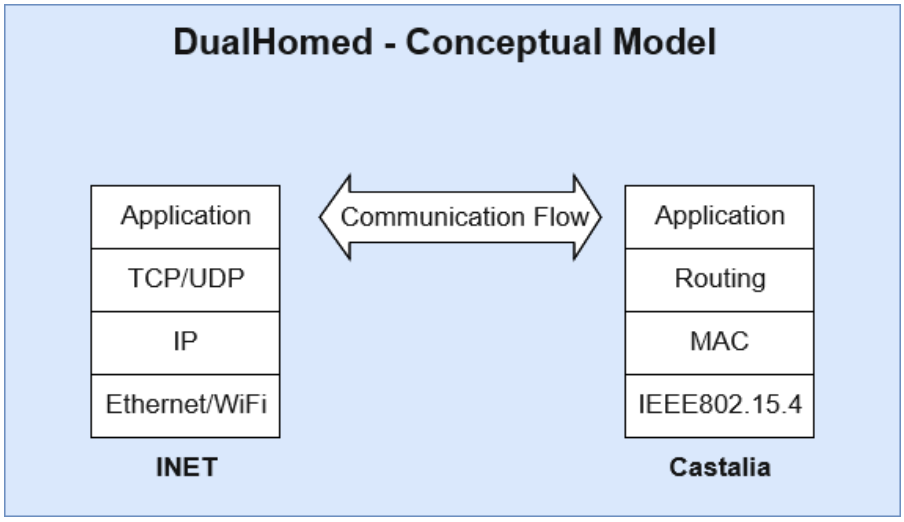

Figure 18: DualHomed - Conceptual Model

The real Implementation of the IP host provided by INET has many features and many components are used in this model, some of them includes:

- Application and Transport: implements TCP, UDP, and SCTP models;
- Internet Layer: implements IPv4, ARP, ICMP, IGMP and error handling models;
- Network Layer: implements Loopback, Radio, and Ethernet Interface models.

It also provides several configurable parameters, routing and interface tables, mobility settings and battery options. A full vision of the WirelessHost model that was chosen to integrate the DualHomed device is given in Figure 19.

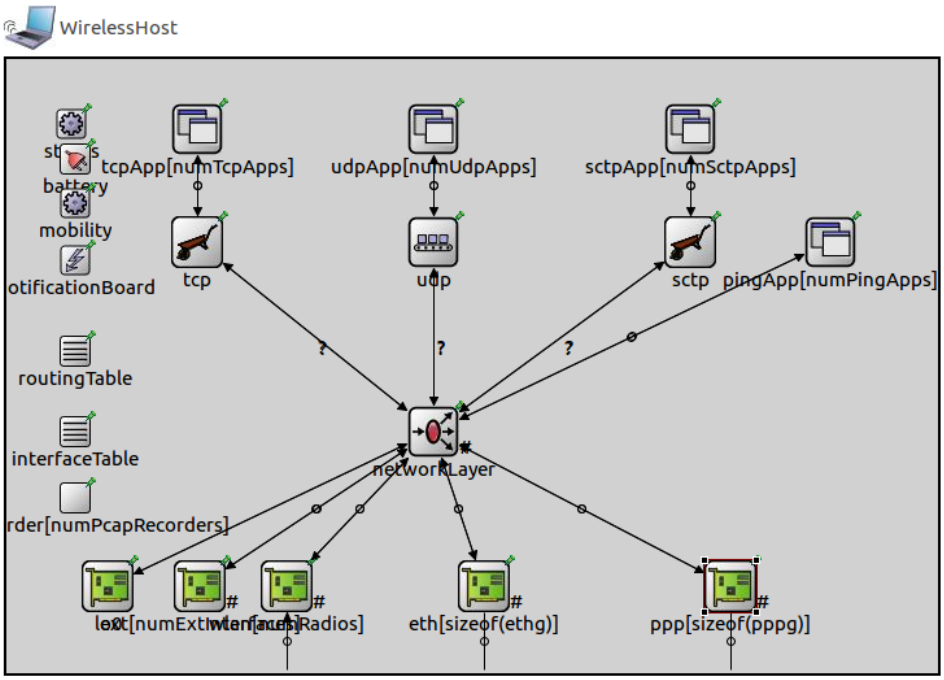

Figure 19: Wireless host model.

Castalia supplies only one kind of node and contains two managers: one that handles communication with the sensor (that is acquiring information) and another that is responsible for the resources such memory, battery and CPU. An application model is responsible for dealing with the data that is been produced in the sensor and the data that is coming from the network.

The network communication is supported/simulated by the communication model and contains the radio, MAC and routing process. To support mobility, it is

possible to apply 3D movements to the node during simulation and this feature is implemented by the mobility manager model.

This Castalia's Node model is simpler than the wireless node supplied by INET, its complete structure can be seen in Figure 20.

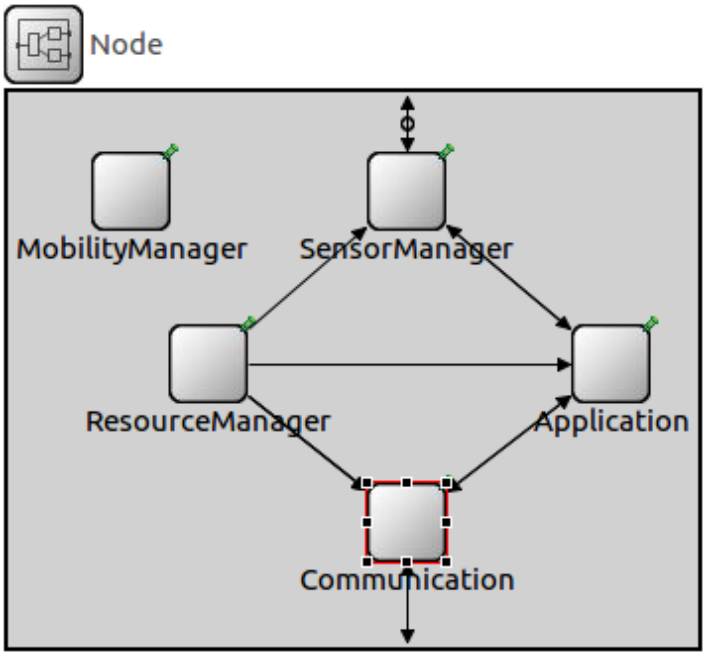

Figure 20: Node Castalia.

The developed model integrates some of the features that are available at both solutions described below.

It would basically be a Castalia Node with a personalized application component, that implements a two-way extra channel to communicate with a Bridge component, that would be responsible for "translate" the communication from INET to Castalia and vice-versa.

Also connected to the Bridge model is a wireless local area interface, its responsibility is to establish the communication with hosts that are communicating using IP that is provided by INET, configuring a hybrid device. This DualHomed device scheme is illustrated in Figure 21.

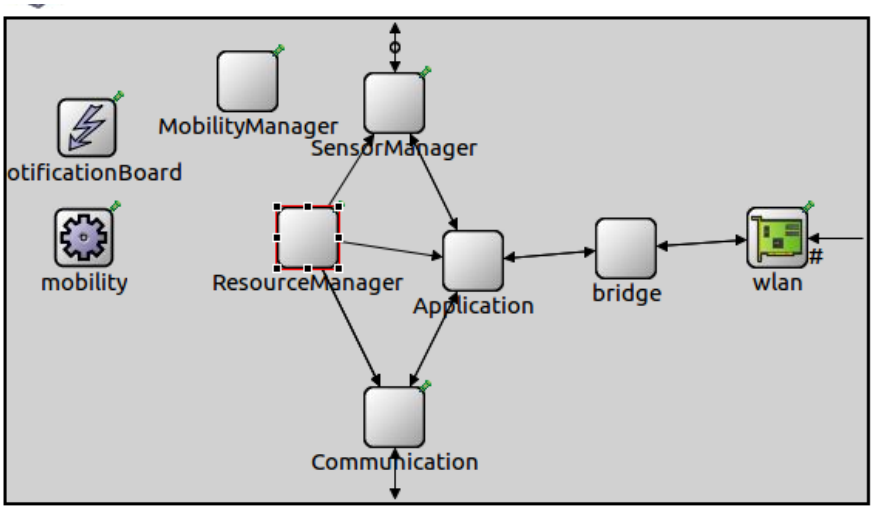

Figure 21: DualHomed device scheme.

This DualHomed device would be placed with standard Castalia's Nodes, communicating with them using a wireless channel modeled to an IoT communication scenario.

In Figure 22 is possible to observe the topology that was proposed: a configurable number of standard nodes that are communicating with each other and receiving inputs from the physical process, transmitting to a single DualHomed device.

Standard IP hosts weren't added to this topology, due to implementation problems that will be described next.

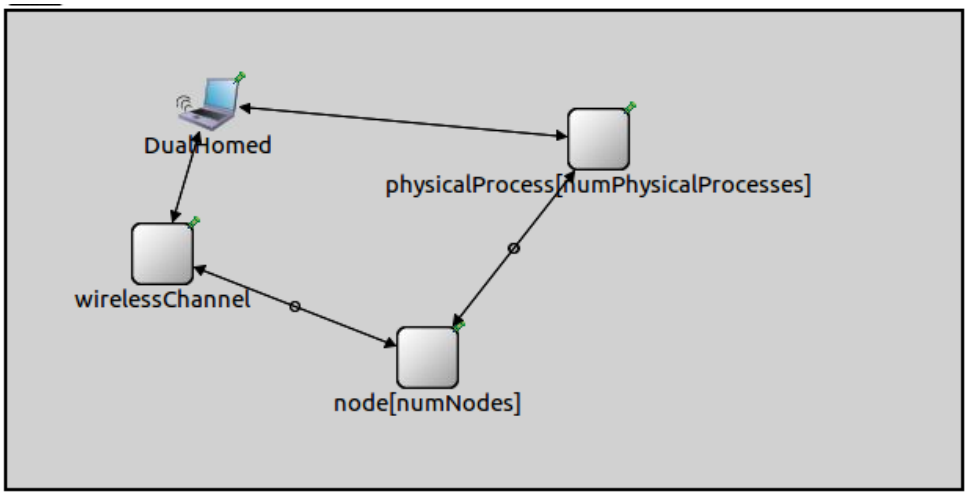

Figure 22: Frist methodology topology.

Castalia was not created to be a framework that would be integrated with others with a high level of flexibility, but a standalone simulator for WSN and BAN networks that is built upon OMNeT.

Both Castalia and INET have classes that share the same name, for instance, MacPacket. When the DualHomed project was built, it checks the linked dependencies to the project, and here things started to get complicated.

Many classes in Castalia get compiler errors because they have the same names used in the INET, creating “no such file or directory” errors. Using both the IDE and the command line interface we get the same errors as can be seen in Figure 23 and in Figure 24.

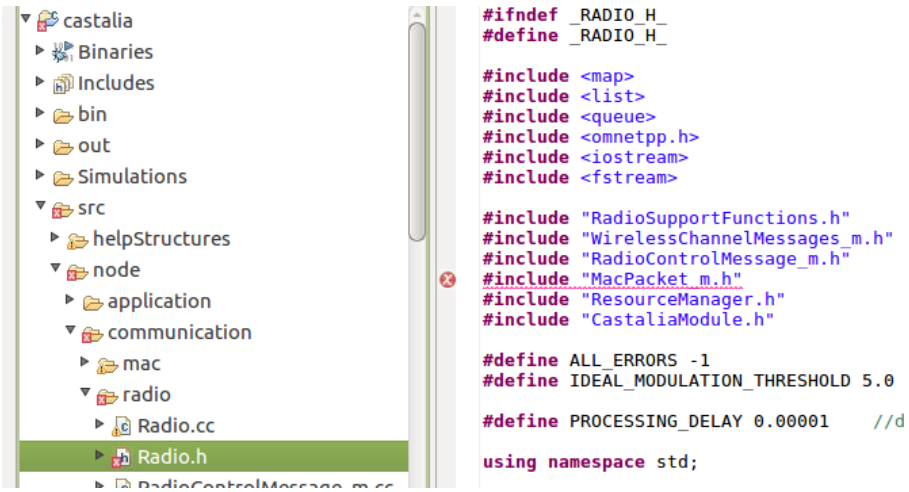

Figure 23: IDE error view.

Being a “simple” error many approaches were applied to overcome this problem, some of which are highlighted:

- Compiler configuration was changed
- The make and makemake process was analyzed
- The full path the references in the source were used

Many other solutions were tried, as well as solutions that the OMNet++ community suggested for similar problems, but it was not possible to overcome this problem.

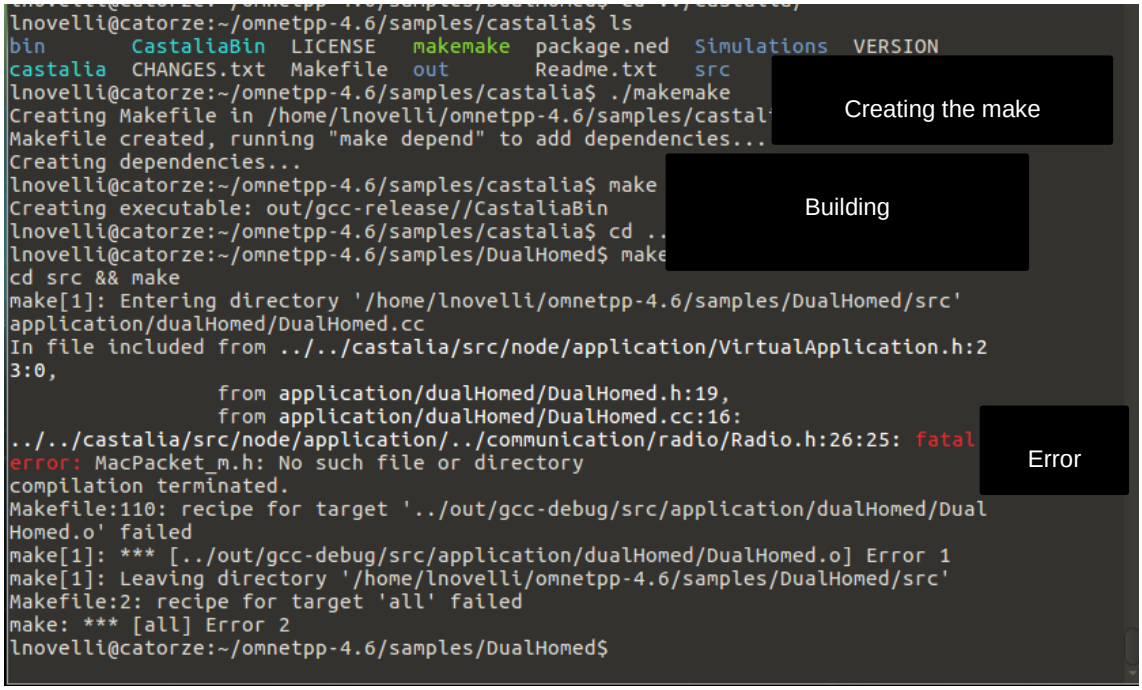

Figure 24: Command line interface error view.

Another possible explanation about this behavior lies in the Castalia architecture itself because in its manual it was not possible to find any reference to the creation of new nodes.

Castalia has virtual classes that implement many of the application behaviors, routing, MAC and mobility models using specific directories inside the file system of the simulator. To create new modules of one of these types is necessary to extend the corresponding virtual class and put the files inside of the directory. However, this behavior cannot be applied to nodes, because the Node has no virtual classes or directories.

6.4. Second Methodology: One Project for each network

This methodology would consist of two independent simulations that would communicate with each other. With this approach each tool would be separated and independent, overcoming the compilation problems that were found and described before.

The conceptual model would be a little different from the previously described, adding a communication channel between the process, as it is displayed in Figure 25.

This kind of simulation could be classified as a distributed simulation but would not require changing any aspect of the current implemented models, because in this methodology they are two separated simulation running without even knowing the existence of the other.

In this implementation, the WSN network project remains with the same versions of the OMNeT and the Castalia mentioned before. The IP network project was developed with the most recent versions of the tools, specifically OMNeT 5.2.1[2] and INET 3.6.4[3].

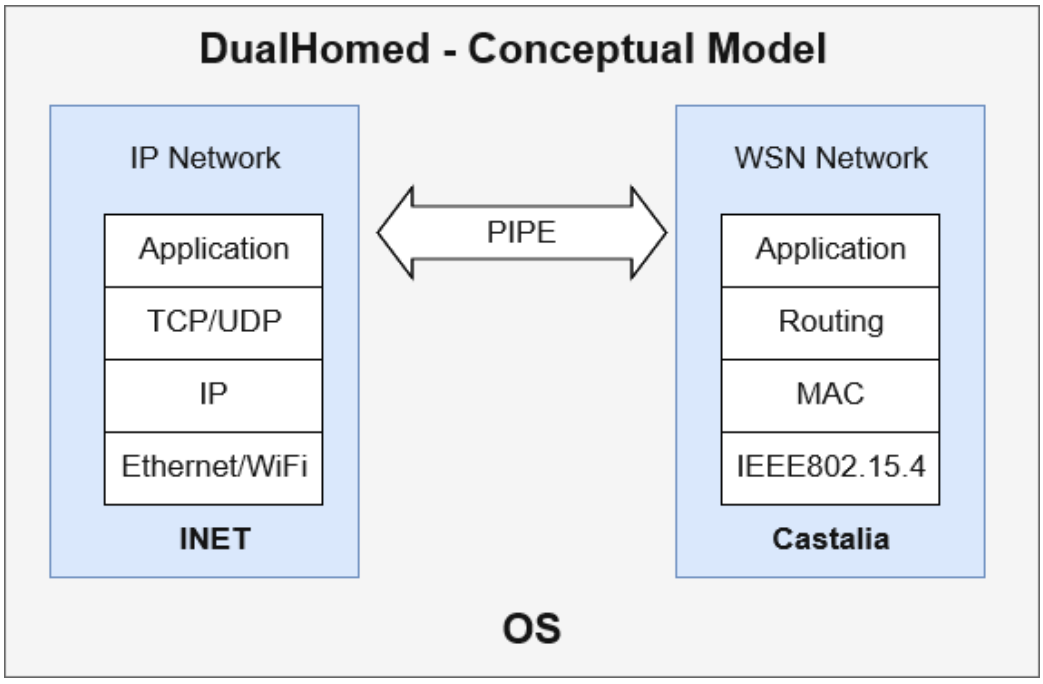

Figure 25: Second methodology model.

Those updated versions have more documentation and examples, and that was a great help in the development process, in contrast with the ones used in the first methodology, where examples and documentation were scarce.

[2] https://www.omnetpp.org/component/jdownloads/download/32-release-older-versions/2321-omnetpp-5-2-1-linux
[3] https://github.com/inet-framework/inet/blob/v3.6.4/WHATSNEW

All the communication between the two processes is handled by a PIPE that is a temporary file that implements a FIFO (first-in, first-out) behavior. This PIPE is implemented by the OS, in this case, LINUX, the simulation only must assure that both processes have the PIPE open at the same time throughout the communication.

6.4.1. IP Network Project

This project includes a host that encapsulates the information and sends it to the Castalia project, and one or more other wireless hosts that are making requests to the WSN network and receiving the information. This topology is displayed in Figure 26-A.

The wireless host that is provided by this INET version is modeled simpler than the older versions, providing:

- Wireless interface
- Lookback interface
- Network model implemented
- UDP transport layer
- UDP app

This model can be seen in Figure 26-B.

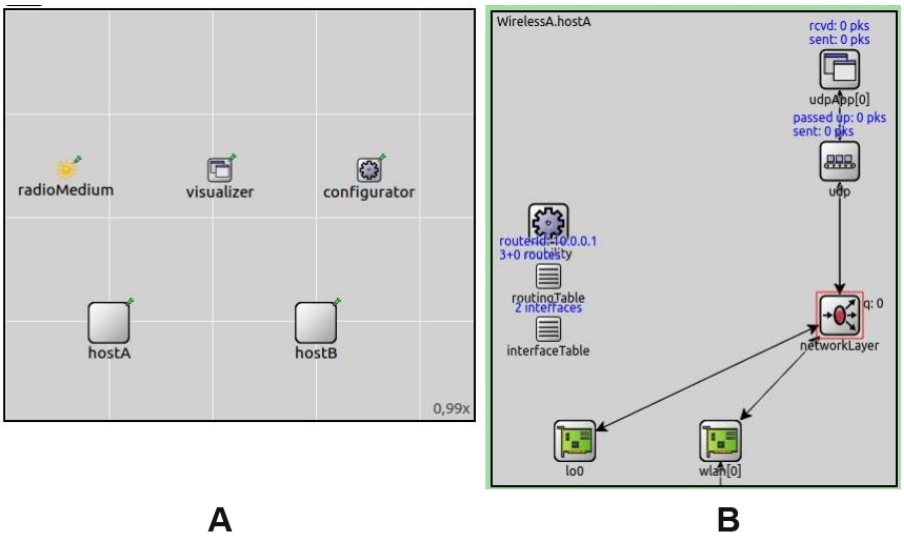

Figure 26: IP Network. A) Topology. B) Host

6.4.2. WSN Network Project

This project remained with the standard Castalia node, in order to avoid the compilation problems described in the previous sections and added at the application level the communication via PIPE, the rest remaining as specified in the first methodology.

6.4.3. The PIPE problem

The PIPE was implemented and was functional when tested with one simulation and a simple outside program that only reads the information and prints it in the standard output. In Figure 27 it is possible to see this communication happening. The WSN simulation has several nodes communicating with just one specific node, and this node is responsible for encapsulating the information being transmitted and sending it to the PIPE.

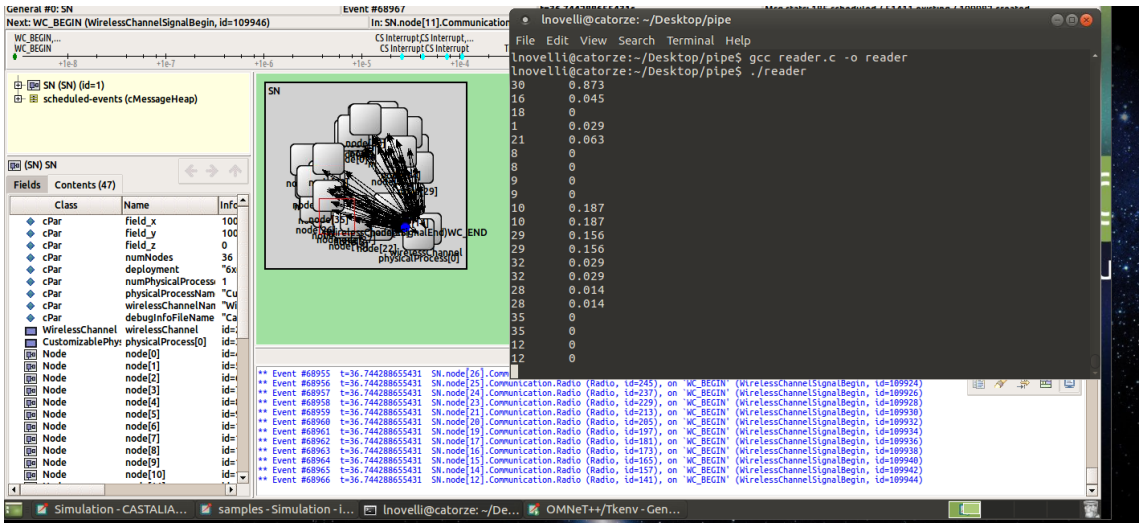

Figure 27: Castalia communication using PIPE with a simple program.

However, when running both simulations at the same the communication does not occur. The PIPE file is created, no error or warning is given in the compilation process nor any type of error occurs during the execution, but in the receiver (e.g. the process that opened the PIPE for reading) only gets empty messages from the FIFO.

The only constrained when using PIPEs is that both ways have to be open during communication. This requirement is achieved on our code because the pipes are open in the initialization method in both model (this method is called even before of the simulation starts and it is used to set variables and parameters to the models).

To overcome this problem was necessary to change the type of data storing, the simplest way was to change the pipe file to a standard text file with no locks.

6.5. Simulation Parameters

The parameters' models are extremely flexible and can be changed to generate various possible simulation scenarios, they are set at the models itself using default values or it can be done in the .ini file. This file contains all the parameters that can be set and its setting is made during the construction of the simulation.

6.5.1. IP Network

Having in mind that the focus of the simulation is to understand the behavior of an IoT environment when added an IP component, the IP simulation will not have many changeable parameters.

The INET Framework presents many parameters that can be changed and those that are being used in this project are showed in Table 9.

Table 9: IP network parameters.

Model Component	**Parameter**	**Description**
Radio	transmitter.communicationRange	Maximum range radio communication
	receiver.ignoreInterference	Ignores the interferences
Mac	fullDuplex	Is used full duplex channel

	useAck	Is used acknowledgment
Network	arpType	ARP module used
UDPApp	destAddresses	Destination address
	destPort	Destination port
	messageLength	Message Length
	sendInterval	Sending Interval
	packetName	Name of the packets

6.5.2. WSN Network

The WSN network is very configurable and the project aims to take advantage of this possibility, in Table 10 is listed the parameters that can be managed in the Castalia Simulator.

Table 10: WSN parameters.

Model Component	**Parameter**	**Description**
Radio	CC2420, CC1000	Texas Instruments radio models
	TxOutputPower	Power used in the radio transmission
	collisionModel	Collision in the radio transmission
	carrierFreq	The frequency of the radio
	bufferSize	Number of frames in the buffer
	maxPhyFrameSize	Maximum frame size in bytes
	phyFrameOverhead	Constant overload in Bytes

MAC	TunableMAC, T-MAC, S-MAC, IEEE 802.15.4	MAC model protocols
	dutyCycle	Duty cycle interval
	macMaxPacketSize	Maximum packet size
	macBufferSize	Buffer size
Routing	bypassRouting, multipathRings	Routing models
	maxNetFrameSize	The maximum size of the frame
	netBufferSize	Net buffer
	netDataFrameOverhead	Overload in bytes
Application	reportDestination	Destination of the generated packets
	isSink	A node that receives the packets
	constantDataPayload	Payload application size
	packet_rate	Number of packets per second
	packetHeaderOverhead	Overhead of the application
	priority	Priority of the packet
WirelessChannel	pathLossExponent	Signal fading
	sigma	How variable is the fading
	signalDeliveryThreshold	Threshold of delivery
	onlyStaticNodes	If there are only static nodes
Network	field_x, field_y, field_z	Dimensions of the deployment field
	numNodes	Number of nodes in the deployment field
mobilityManager	xCoorDestination, yCoorDestination, zCoorDestination	Destination of the node when it is not static

	Speed	Speed in seconds of the dislocation

6.6. Simulation Outputs

The OMNeT simulator provides all the necessary features to collect the data that is generated in the simulations. This functionality is enabled in the models by throwing signals that are collected and stored in three kinds of files: scalars, histograms, and the vectors.

The scalar file is composed by aggregate values that are obtained at the end of the simulation (e.g. maximum, minimum, average), the vector file consists in values that are recorded in function of the simulation time. The histograms are representations of the distribution from the other data files.

The data collected from the simulation models can be grouped in information that represents the communication process that occurs in both networks, meaning the IP network provided by the INET framework and the WSN network from Castalia Simulator.

6.6.1. IP Network

In Table 11 are present the description of all the information that is been collected from the simulation model grouped by Scalar, Vector, and Histogram.

Table 11: Recorded values from IP simulation.

FILE	Group	Values Recorded
Scalar	Application	Received or transmitted packets count by units and bytes and the lifetime of a packet.
	Transport	Received and transmitted packets and the ones that have been dropped by a bad checksum or wrong port.

	Network	Queueing time, dropped packets, queue length, received packets, packets treated in the MAC, transmission and received state, symbol, bit and packet error rate.
	Medium	Counting of: transmission, radio frames send, reception, interference, reception decision, listening decision. Cache hit by reception, interference, noise, snir, reception decision and result.
Vector	Application	Send Packets, throughput, an end to end delay, received packets
	Transport	Send packets, received packets
	Network	Queue length, radio mode, radio state, queueing time
Histogram	Application	End to end delay
	Network	Queueing time, symbol error rate, bit error rate, packet error rate,

6.6.2. WSN Network

The output collection mechanism that is used in the Castalia Simulator differs from the standard OMNeT data collection. The Castalia has a tool that collects and process the data from the module simulations, the explanation of those values can be seen at Table 12.

Table 12: WSN simulation data.

Module	**Output**	**Recorded Values**
MAC	Sent packets loss	Loss packets at the MAC layer
Radio	RX pkt loss	Loss packets at receiver on the Radio layer
	TXed pkts	Transmitted packets
Resource Manager	Consumed Energy	Consumed energy in each node
	Estimated network lifetime	Remaining time of the network

	Remaining Energy	Remaining Energy in each node
Simulation	Execution ratio	Simulation Time / Real time
	Execution time	Execution Time

6.7. Final setup

The simulation occurred in a period of 100hours with a warm-up period of 2 minutes. The confidence interval for the results is 95%, with 5 repeats that use different seeds to generate random values.

This simulation WSN component has 36 sensors placed randomly in a field that is 100x100 meters.

The final topology in the simulation graphical interface can be seen in Figure 28. In the left side the IP-hosts are displayed and on the other side, the sensors are displayed.

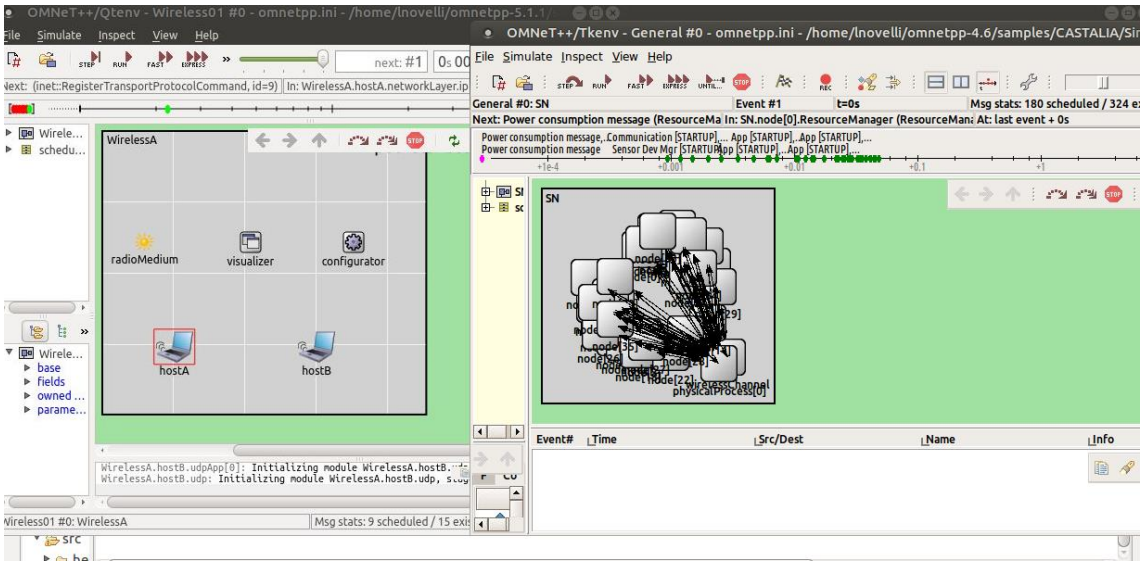

Figure 28: Final topology

Chapter 7 Simulation Results and Analysis

In this chapter the results from the simulation are analyzed and summarized.

7.1. IP Network

The complete configuration of the simulations can be found in ANNEX 2 – OMNeT++ Configuration Files.

The results of the communication in the IP network are summarized in two groups: the queue for the received messages in the host that represents the bridge between the networks and the end-to-end delay of the communication.

The queue length is an important measure because it displays the time that the waiting time to the packets been transmitted.

The simulation showed that this waiting time is close to 0, given the simplicity of the IP network topology, that has only two hosts communicating, thus, most of the time the channel is free for the bridge host.

It is important to highlight that the changes in the parameters of the WSN network have not interfered on the result achieved in this network, therefore the analysis here is valid for all scenarios in the WSN network.

Figure 29 illustrates a histogram of the time that a packet waited in the queue to be sent. It can be seen that the bar that representing almost no waiting time is

clearly where most of the packets are. It also shows that some packets waited between 0 and 10 milliseconds and few packets waited at most 20 milliseconds.

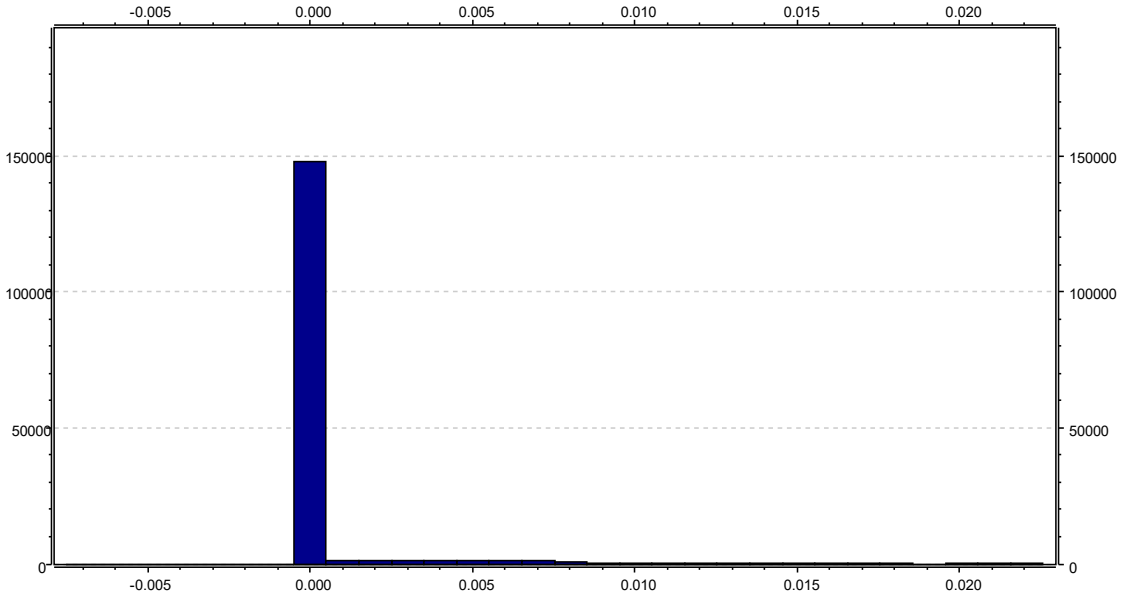

Figure 29: Queue histogram.

Figure 30 complements the previous analysis with a dispersion graph that represents each packet individual waiting time. It shows clearly that most of the packets waited less than 10 milliseconds to gain access to be transmitted.

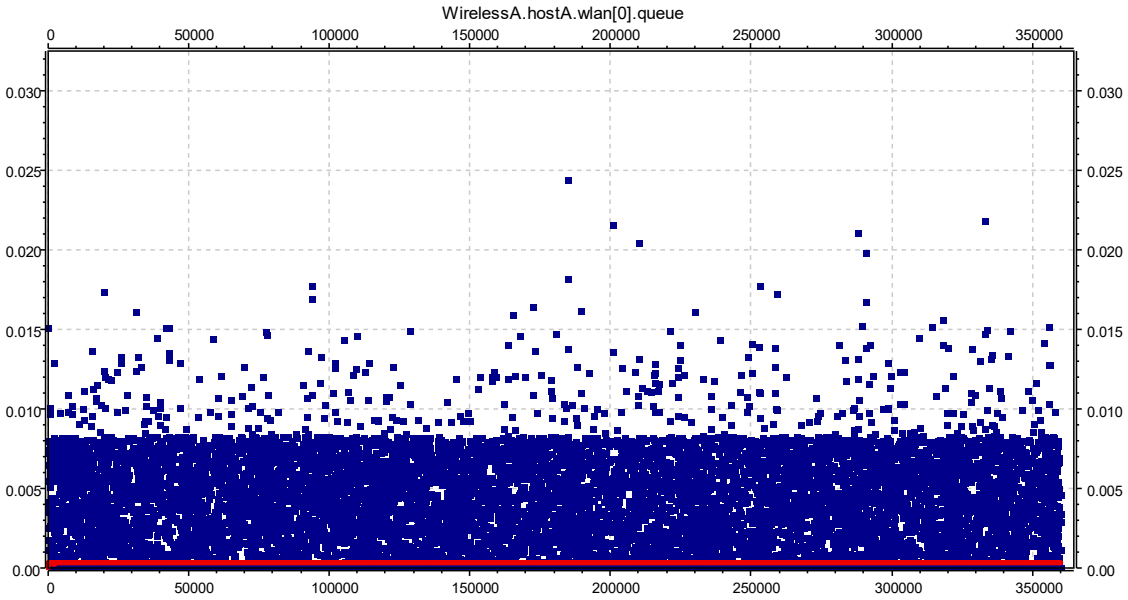

Figure 30: Dispersion graph of the queue waiting time.

The throughput achieved at host destination is also elevated, where the arguments of simplicity of the topology and channel availability can be applied here too.

In Figure 31 is possible to visualize that most of the communication process achieved 240.000 Kb/s.

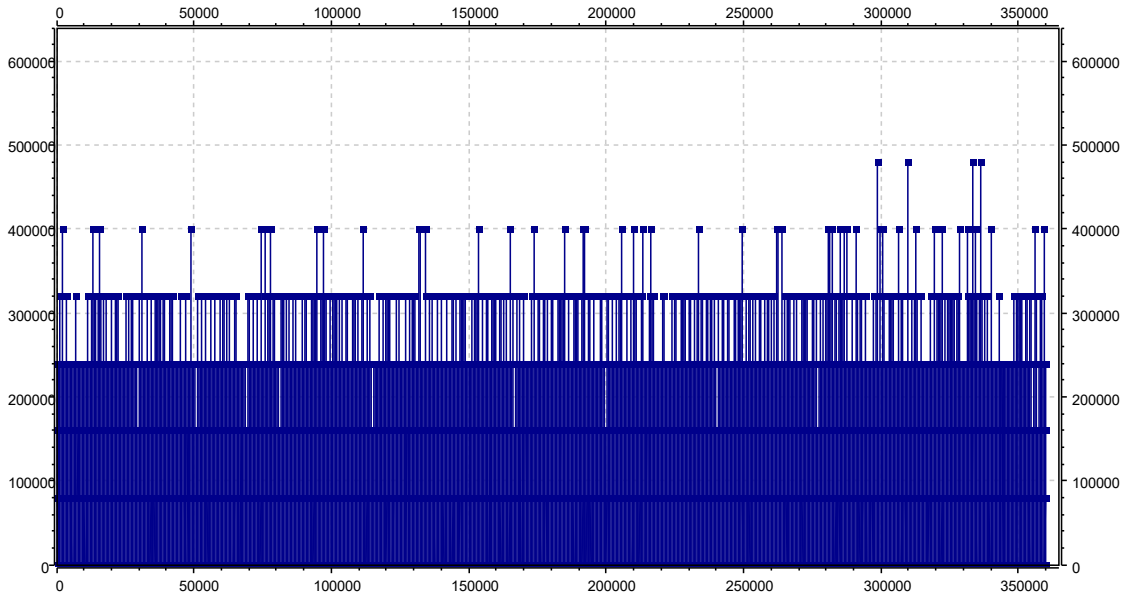

Figure 31: Destination host throughput.

The end-to-end delay represents the amount of time that each packet takes to leave its source at the WSN network and arrive in the destination host in the IP network.

Figure 32 and Figure 33 display this information, using a histogram and a dispersion graph, respectively.

Both graphs show that the delay between the networks lasts at least 8 milliseconds, and from the analysis of the queue is possible to infer that this time is mainly caused by the WSN.

Is also possible to see at the histogram that most of the packets have a delay between 8 and 9 milliseconds.

From the dispersion graph, it's possible to state that most of the packets are delayed at most by 16 milliseconds, there are some samples that overpass the 20 milliseconds and some outliers that achieve 30 milliseconds.

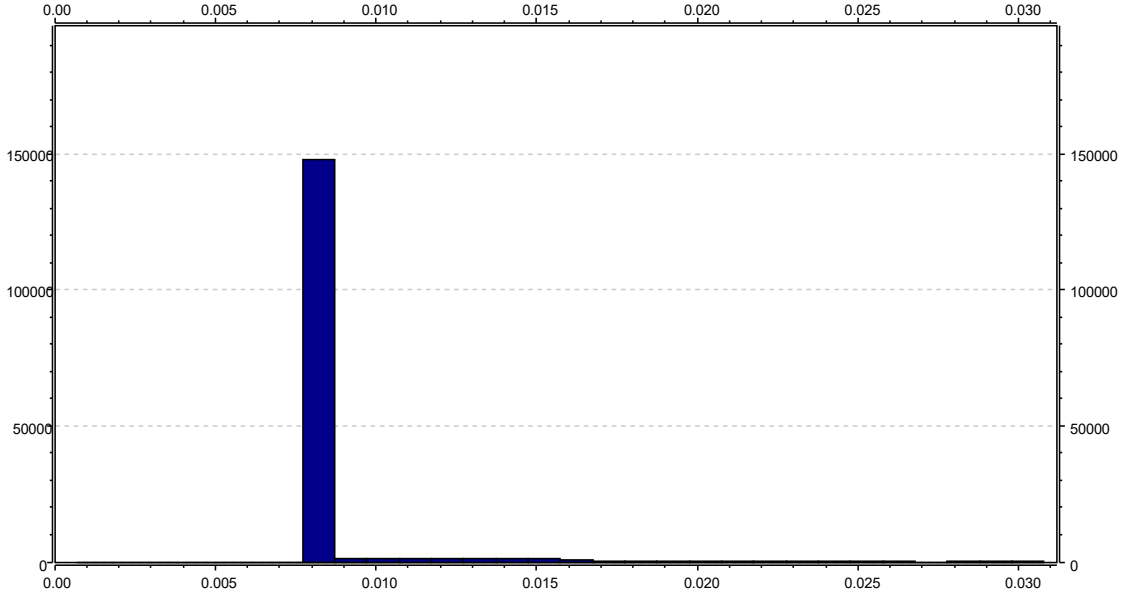

Figure 32: End-to-end delay histogram from WSN to IP.

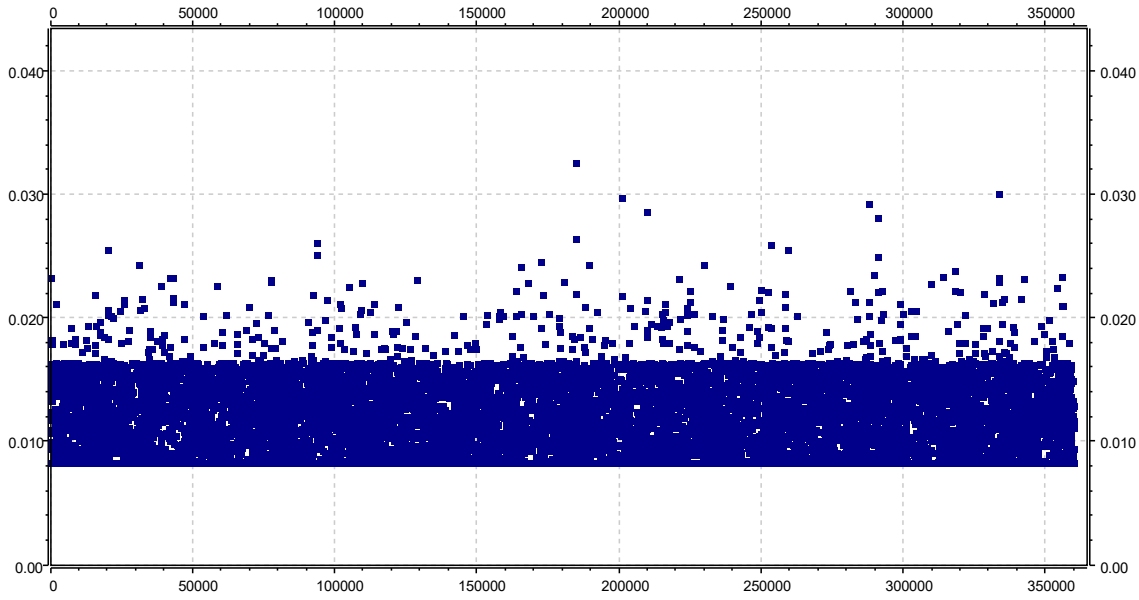

Figure 33: End-to-end delay dispersion graph.

In Table 13 gives many other summarized complementary information about the behavior of the modules present in the IP network.

Table 13: Complementary information

Measure	Value
Total packets received/ transmitted	157473
Queue maximum length	3
Queue average time	$1.2183*10^{-4}$
Total bytes sent	$1.58682*10^{8}$
Total packets received at the destination host	157453
Total bytes received at the destination host	$1.5742*10^{8}$
End-to-end delay – average	0.0085 milliseconds
End-to-end delay – standard deviation	0.0012 milliseconds

7.2. WSN Network

The simulations of the WSN network ran in two fronts, changing parameters that are relative to the software.

At the software front, it was proposed to analyze the behavior of the nodes by changing their routing protocols. The routing protocols used are the byPassRouting, which presents no routing function and the MultipathRouting that implements a system where all nodes get a number, and when transmitting it verifies if its number is lower of the previous node that transmitted this packet to transmit it again (resend it); if the number is higher, the packet is not retransmitted.

The MAC protocol used is the Timeout-MAC (TMAC) protocol, which is a variation of the Sensor-MAC protocol (SMAC). The TMAC has fixed periods for sleep and to be active, while in SMAC these periods are flexible.

Table 14 displays the items that will be recorded.

Table 14: Measures at from the simulation

Module	Value
MAC	Sent packets loss
RADIO	Receiver packets loss
	Transmitter packets loss
Resource Manager	Consumed Energy
	Estimated lifetime
	Remaining Energy

In Table 15 are displayed the total and the average for the sent packets loss for each of the routing protocols. Packet loss means that for some reason the packet can't be used for that specific level.

Is possible to notice that the BypassRouting has much fewer losss since it implements no logic for the routing process. Multipath routing has 624% more loss both in total and average.

Table 15: MAC module sent packets loss summary

Group	Protocol	Count (packets)
Total	BypassRouting	215.999
	MultipathRingsRouting	1.348.683
Average	BypassRouting	599.998
	MultipathRingsRouting	3.746.343

Figure 34 displays the packet loss for each node at the MAC level.

Is possible to see that the ByPassRouting algorithm has a constant behavior, maintaining around 6.000 packets breaks for every node. When analyzing the curve from the MultiPathRingsRouting can be very unpredictable, presenting much worse result for every node.

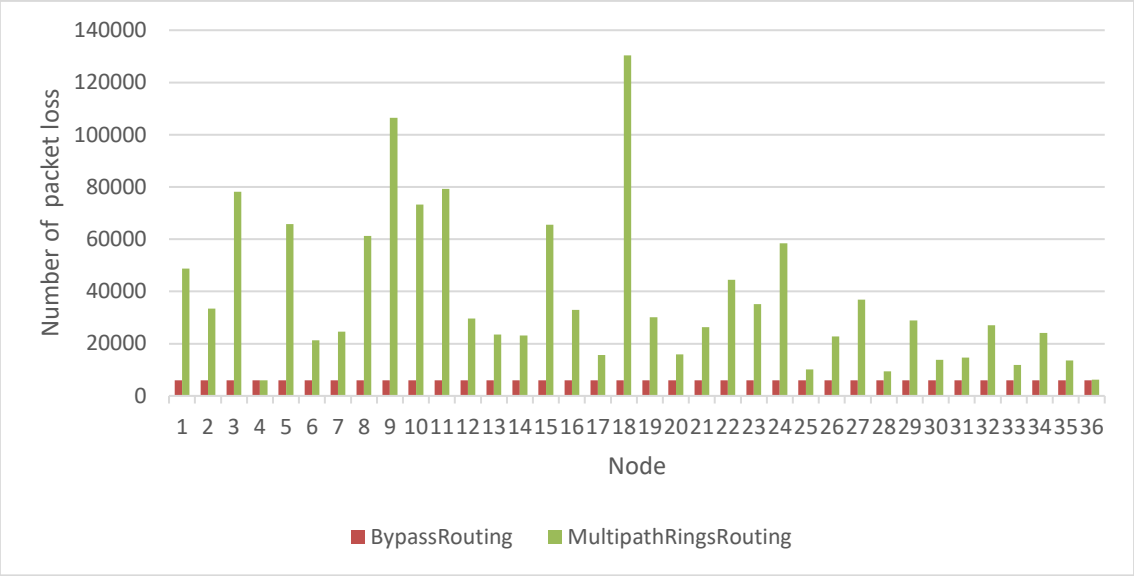

Figure 34: MAC sent loss by routing protocol.

At Radio level the messages receive a classification when they arrive, that states what happened to them during the course from the source until arriving at the destination. Figure 35 displays, for each protocol, the packets' state at the receiver and it's possible to infer the poor performance of the MulpathRingsRouting when compared with the BypassRouting, achieving a worse result for every state.

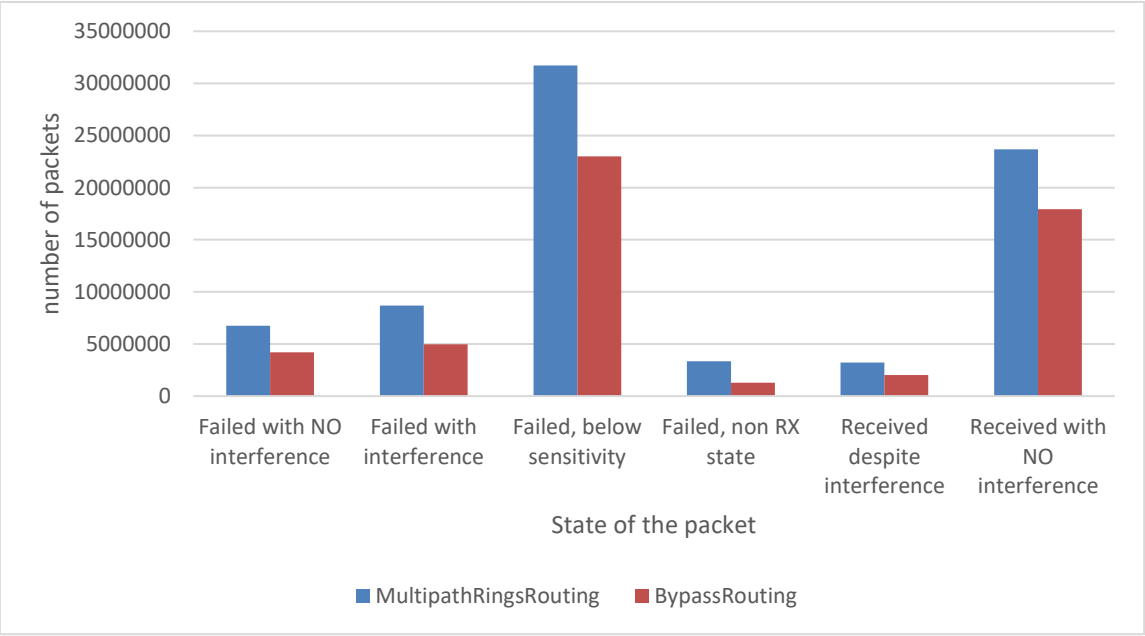

Figure 35: Radio receiver packet loss comparing both protocols.

When expanding the analysis of the radio receiver for each packet is possible to achieve more granularity in the results and try to verify if MultipathRingsRouting is better in some specific scenario.

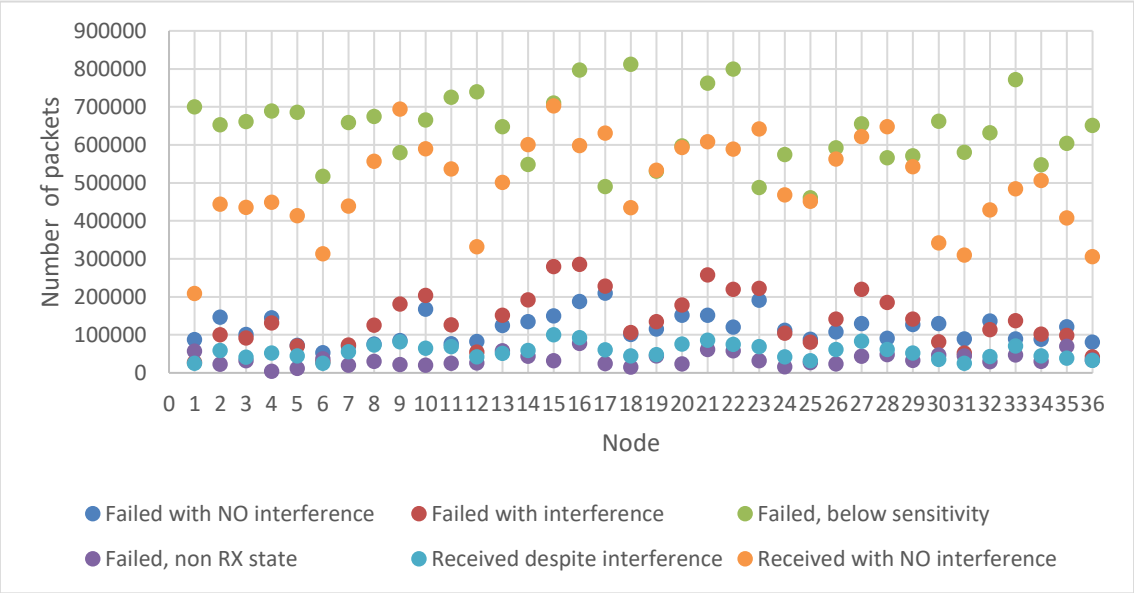

Figure 36: Radio receiver packet loss using byPassProtocol for each node.

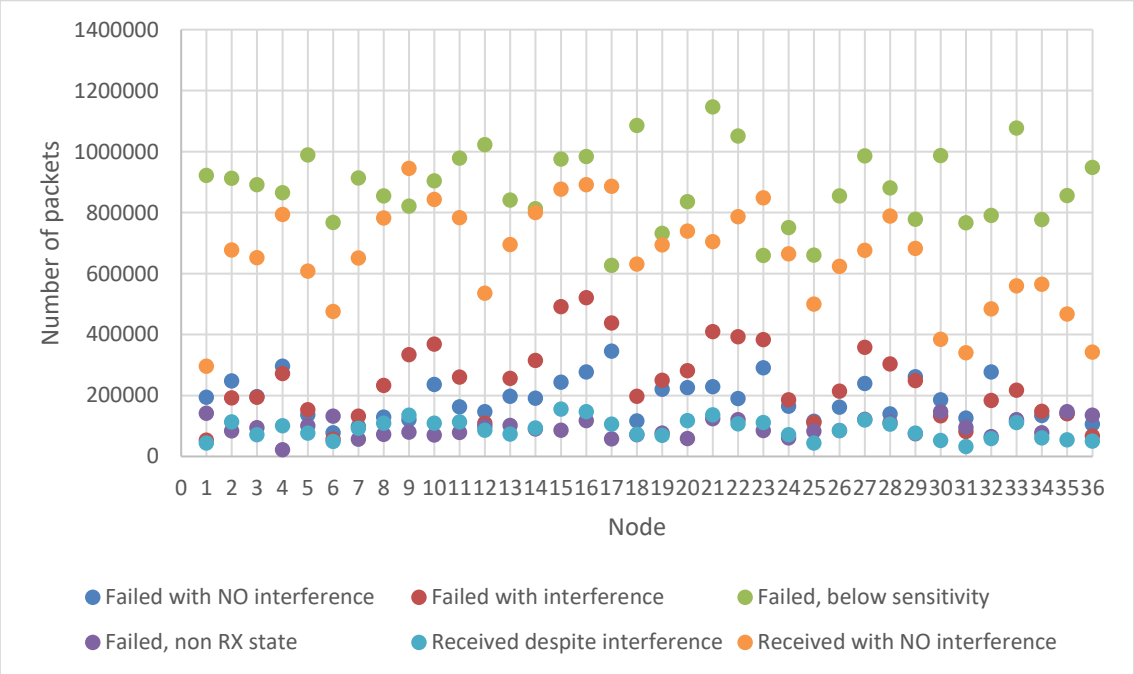

Figure 37: Radio receiver packet loss using MultiPathRing for each node.

When analyzing both Figure 36 and Figure 37, it can be seen that the gap between the orange and green dots, representing received and failure, respectively. The overall behavior stays the same, with green dots always on top.

Was also noticeable the increase of all types of failed packets when comparing the MultipathRIngsRouting with the byPassProtocol.

The red dots that represent the state “Failed with interference” are higher, meaning that this routing protocol increases the interferences in the communication process of the nodes. The remaining states maintain the same aspect when comparing both protocols.

When the node is transmitting the data, the behavior described previously is also present. BypassRouting maintains almost a continuous line near of 60.000 packets and the MultipathRingsRouting present an unpredictable behavior, having higher oscillation between the nodes, as can be seen in Figure 38.

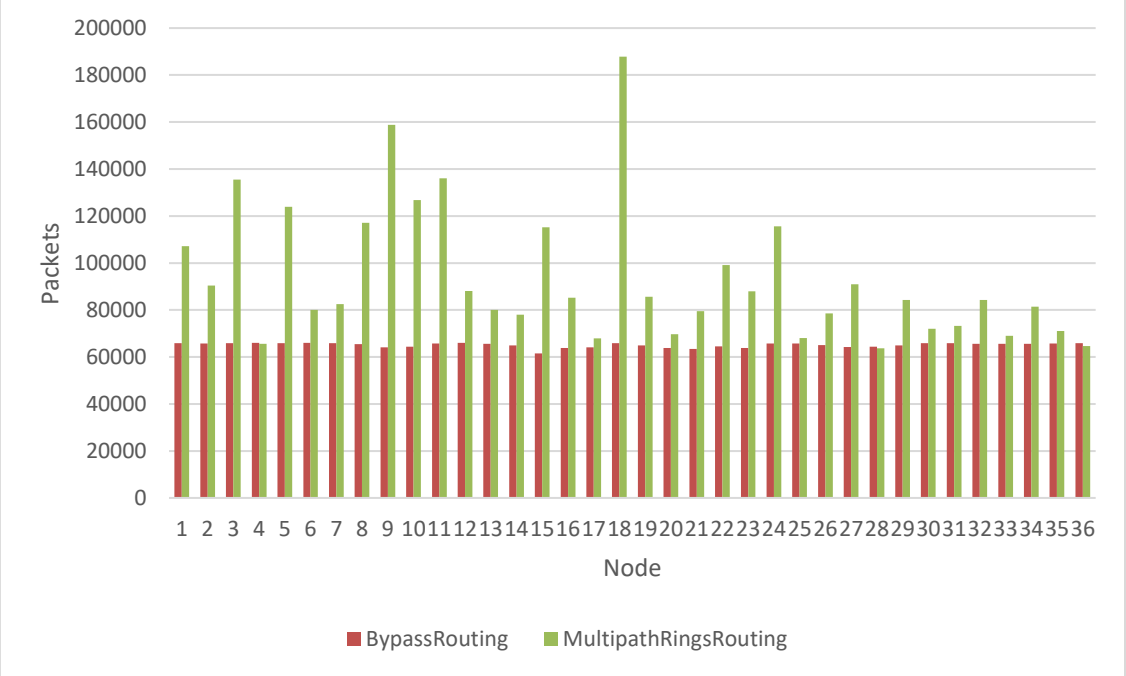

Figure 38: Transmitter radio packet loss for each node.

For an IoT environment, the energy consumption is a major factor to be considered when choosing some protocol or technology to deploy a solution in a real environment, as explained at previous chapters.

One more time the BypassRouting is more effective when compared with MultipathRingRouting protocol, Figure 39 illustrates the inefficiency of the multipathing, for some nodes it came close to the Bypass, but at the overall performance, the two lines are distant, showing that energy is being wasted with MultipathRings.

The total amount of energy was respectively 129.891 KJ and 133.906 KJ to BypassRouting and MultipathRingsRouting, a difference at the order of 3%.

For the same conditions, the estimated remaining energy would keep the network running for more 21 days when using the BypassRouting and 20 days enabling the multipathingRouting.

The estimated difference of 3% in energy efficiency would be provided a whole day in the sensor lifetime, totalizing 25 days for ByPass and 24 days for MultipathRings.

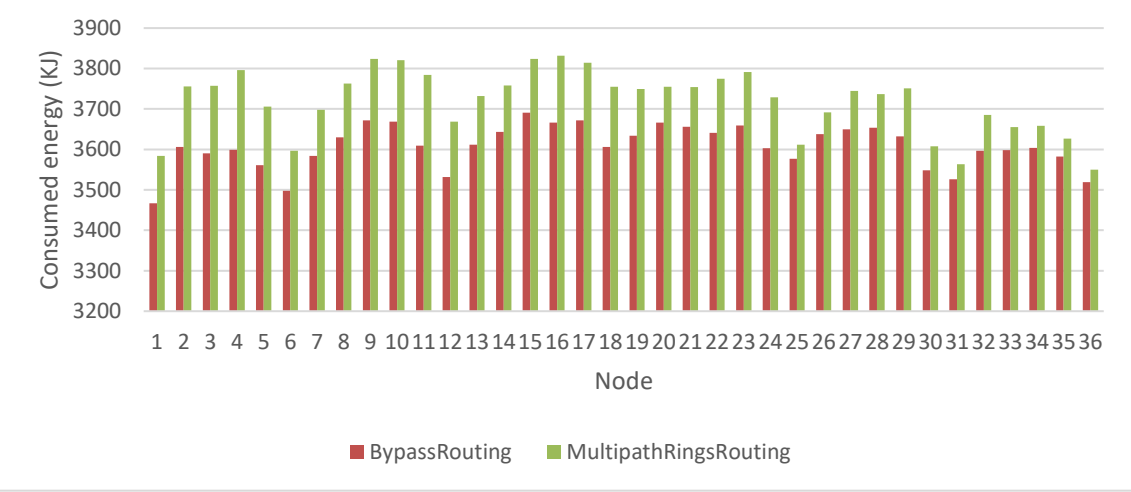

Figure 39: Consumed energy for both protocols.

Facing this kind of results the research achieves an odd conclusion regarding the routing models that are present in Castalia, it's better to have no routing techniques than using the MutipathRings solution.

The protocol represents no gain in a number of packets that are accepted, it does not increase the quality of transmission, the status of the packets remains almost the same.

When analyzing the energy consumption, that is crucial for this kind of devices, the MultipathRings represents an increase at node's consumption, an order of 3%, representing a whole day in the network lifetime.

This behavior can be caused by the poor methodology that the protocol applies for routing. The packets are retransmitted based only in a number that is attributed to a node in the network initialization, regardless the traffic flow, devices' physical location, latency, delays, throughput and other factors and measures that interfere in the communication and consequently how the routing must be made.

Chapter 8 Conclusion and Future Work

This chapter summarizes the project achievements and proposes some lines to continue the simulation research.

8.1. Project overview

This research project proposed three main objectives:

- Review of the state-of-the-art literature on networks and simulation;
- Describe the general test bed;
- Implement the test bed;

Regarding the first objective, was delivered an extensive overview that results in the article present in ANNEX 1.

Two major tests beds were presented, the low coverage range networks scenario for smart-homes and industrial automation and the high coverage range networks for smart-metering, smart-cities and smart-buildings.

The implementation process was realized using the OMNeT++ discrete event simulator with the Castalia Simulator and INET Framework.

A topology for integrating an IP network with a WSN network was proposed. Two approaches were applied for implementing this integration: a complete merge of the networks, that proved to be fruitless and one where both networks were

independent and didn't know about the existing one another, that ended up being used.

An interface packet format for the communication process was established and defined for the information exchange between the networks. This interface format included: destination, source, payload and simulation time.

For the actual simulation results presented, the parameters used were:

- MAC protocol = TMAC;
- Routing Protocols = Bypass and MultiPathRings;
- Radio model = CC2420 Texas Instrument.

Both networks outputs were analyzed and were possible to infer:

- The waiting time for the radio's communication queue was near to 0;
- The end-to-end delay was at least 8 milliseconds;
- The MultiPathRings routing present more packet breaks at MAC and Radio level;
- The communication process with the MultiPathRings is not improved, the state of the packets mostly remains the same;
- The energy consumption with MultiPathRings was 3% higher and it resulted in one day less for the network lifetime.

8.2. Future work

Some possible paths for future work include:

- Create the simulation implementation for the case study B, that unfortunately could not be included in the simulation scope of this project.
- Investigate if the MultiPathRings can prove a better solution for less extreme scenarios, in scenarios where scalability is an important factor or in some case where the distance between the nodes is higher.
- Create new modules to Castalia Simulator that include actually in use industry patterns or the de-facto standards for communication.

REFERENCES

[1] A. Al-Fuqaha, M. Guizani, M. M.-… S. & Tutorials, and U. 2015, “Internet of things: A survey on enabling technologies, protocols, and applications,” *ieeexplore.ieee.org*, 2015.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, “Internet of Things for Smart Cities,” *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[3] G. D’Angelo, S. Ferretti, and V. Ghini, “Modeling the Internet of Things: a simulation perspective,” in *2017 International Conference on High Performance Computing & Simulation (HPCS)*, 2017, pp. 18–27.

[4] A. Varga, “THE OMNET++ DISCRETE EVENT SIMULATION SYSTEM.”

[5] “OMNeT++ Simulation Manual.”

[6] A. Boulis, “Castalia: A simulator for Wireless Sensor Networks and Body Area Networks - User’s Manual,” no. May, 2013.

[7] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Comput. Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[8] A. F. Molisch, K. Balakrishnan, C.-C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, U. Schuster, and K. Siwiak, “IEEE 802.15.4a channel model -final report.”

[9] J. Vasseur, C. Fellow, C. Systems, N. Agarwal, T. Leader, and J. Hui, “RPL: The IP routing protocol designed for low power and lossy networks Internet Protocol for Smart Objects (IPSO) Alliance,” 2011.

[10] J. W. Hui and D. E. Culler, “Extending IP to Low-Power, Wireless Personal Area Networks,” *IEEE Internet Comput.*, vol. 12, no. 4, pp. 37–45, Jul. 2008.

[11] B. L. Risteska Stojkoska and K. V. Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions,” *J. Clean. Prod.*, vol.

140, pp. 1454–1464, Jan. 2017.

[12] Y. Kabalci, “A survey on smart metering and smart grid communication,” *Renew. Sustain. Energy Rev.*, vol. 57, pp. 302–318, May 2016.

[13] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” 2014.

[14] C. Bormann, A. P. Castellani, and Z. Shelby, “CoAP: An Application Protocol for Billions of Tiny Internet Nodes,” *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, Mar. 2012.

[15] P. Saint-Andre, “RFC 6120 - Extensible Messaging and Presence Protocol XMPP,: Core,” 2011.

[16] World Wide Web Consortium., “Extensible Markup Language (XML) 1.0 (Fifth Edition),” 2008.

[17] OMG, “Data Distribution Service (DDS) Brief,” no. April, pp. 1–20, 2015.

[18] A. Stanford-Clark and H. L. Truong, “MQTT For Sensor Networks (MQTT-SN) Protocol Specification,” 2013.

[19] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks,” in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE ’08)*, 2008, pp. 791–798.

[20] OASIS Standard, “OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0,” 2012.

[21] M. Belshe, R. Peon, and E. M. Thomson, “Hypertext Transfer Protocol Version 2 (HTTP/2),” 2015.

[22] S. Cheshire and M. krochmal, “Multicast DNS,” 2013.

[23] S. Cheshire and M. Krochmal, “DNS-Based Service Discovery,” Feb. 2013.

[24] C. Pautasso, O. Zimmermann, and F. Leymann, “Restful web services vs. "big"" web services,” in *Proceeding of the 17th international conference on World Wide Web - WWW ’08*, 2008, p. 805.

[25] J. Postel, “RFC 768 - User Datagram Protocol.”

[26] L. Eggert, G. Fairhurst, and G. Shepherd, “RFC 8085 - UDP Usage Guidelines,” 2017.

[27] “RFC 793 - Transmission Control Protocol,” 1981.

[28] J. Postel, “Internet Protocol.”

[29] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification.”

[30] H. Zimmermann, “OSI Reference Model--The ISO Model of Architecture for Open Systems Interconnection,” *IEEE Trans. Commun.*, vol. 28, no. 4, pp. 425–432, Apr. 1980.

[31] M. R. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. A. Grieco, G. Boggia, and M. Dohler, “Standardized Protocol Stack for the Internet of (Important) Things,” *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[32] M. Standards Committee of the IEEE Computer Society, “IEEE Std 802.15.4-2011, IEEE Standard for Local and metropolitan area networks—Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs),” 2011.

[33] K. S. J. Pister and L. Doherty, “TSMP : TIME SYNCHRONIZED MESH PROTOCOL,” *EAI Endorsed Trans. Internet Things*, vol. 1, no. 1, 2015.

[34] M. R. Palattella, M. Dohler, A. Grieco, G. Rizzo, J. Torsner, T. Engel, and L. Ladid, “Internet of Things in the 5G Era: Enablers, Architecture, and Business Models,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[35] ZigBee Alliance, “Zigbee Specification,” 2012.

[36] C. Gomez and J. Paradells, “Wireless home automation networks: A survey of architectures and technologies,” *IEEE Commun. Mag.*, vol. 48, no. 6, pp. 92–101, Jun. 2010.

[37] A. Wheeler, “Commercial Applications of Wireless Sensor Networks Using

ZigBee," *IEEE Commun. Mag.*, vol. 45, no. 4, pp. 70–77, Apr. 2007.

[38] Z-Wave, "Z-Wave Protocol Overview," 2006.

[39] V. Coskun, B. Ozdenizci, and K. Ok, "A Survey on Near Field Communication (NFC) Technology," *Wirel. Pers Commun*, vol. 71, pp. 2259–2294, 2013.

[40] R. Want, "Near field communication," *IEEE Pervasive Comput.*, vol. 10, no. 3, pp. 4–7, 2011.

[41] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, vol. 12, no. 12, pp. 11734–11753, Aug. 2012.

[42] Thread Group, "Thread 1.1.1 Specification." 2017.

[43] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, Fifth. Boston: Prentice Hall, 2011.

[44] GS, "EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz."

[45] H. Hu, D. Kaleshi, A. Doufexi, and L. Li, "Performance Analysis of IEEE 802.11af Standard Based Neighbourhood Area Network for Smart Grid Applications," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015, pp. 1–5.

[46] B. Bellalta, L. Bononi, R. Bruno, and A. Kassler, "Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges," *Comput. Commun.*, vol. 75, pp. 1–25, Feb. 2016.

[47] A. B. Flores, R. E. Guerra, E. W. Knightly, P. Ecclesine, and S. Pandey, "IEEE 802.11af: a standard for TV white space spectrum sharing," *IEEE Commun. Mag.*, vol. 51, no. 10, pp. 92–100, Oct. 2013.

[48] M. Park, "IEEE 802.11ah: sub-1-GHz license-exempt operation for the internet of things," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 145–151, Sep. 2015.

[49] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, “A survey on IEEE 802.11ah: An enabling networking technology for smart cities,” *Comput. Commun.*, vol. 58, pp. 53–69, Mar. 2015.

[50] T. Kato, “Standardization and Certification Process for ‘Wi-SUN’ Wireless Communication Technology,” no. 1, 2016.

[51] “JupiterMesh FAQ | Zigbee Alliance.” [Online]. Available: http://www.zigbee.org/faq/jupitermesh-faq/. [Accessed: 26-Dec-2017].

[52] M. Hasan, E. Hossain, and D. Niyato, “Random access for machine-to-machine communication in LTE-advanced networks: issues and approaches,” *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 86–93, Jun. 2013.

[53] M. Elsaadany, A. Ali, and W. Hamouda, “Cellular LTE-A Technologies for the Future Internet-of-Things: Physical Layer Features and Challenges,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2544–2572, 2017.

[54] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low Power Wide Area Networks: An Overview,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[55] K. E. Nolan, W. Guibene, and M. Y. Kelly, “An evaluation of low power wide area network technologies for the Internet of Things,” in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 439–444.

[56] N. Varsier and J. Schwoerer, “Capacity limits of LoRaWAN technology for smart metering applications,” in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[57] D. Bankov, E. Khorov, and A. Lyakhov, “On the Limits of LoRaWAN Channel Access,” in *2016 International Conference on Engineering and Telecommunication (EnT)*, 2016, pp. 10–14.

[58] J. de Carvalho Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino, “LoRaWAN - A low power WAN protocol for Internet of Things: A review and opportunities,” *2017 2nd Int. Multidiscip. Conf. Comput. Energy Sci.*, pp. 1–6, 2017.

[59] C. Goursaud and J. M. Gorce, “Dedicated networks for IoT : PHY / MAC state of the art and challenges.”

[60] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, “Understanding the Limits of LoRaWAN,” *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.

[61] G. D’Angelo, S. Ferretti, and V. Ghini, “Simulation of the Internet of Things,” in *2016 International Conference on High Performance Computing & Simulation (HPCS)*, 2016, pp. 1–8.

[62] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” *Proc. 1st Int. Conf. Simul. tools Tech. Commun. networks Syst. Work.*, p. 60, 2008.

[63] OMNeT++, “INET Framework for OMNeT++,” p. 157, 2012.

ANNEX 1 - Application Protocols and Wireless Communication for IoT: A Simulation Case Study Proposal

Application Protocols and Wireless Communication for IoT: A Simulation Case Study Proposal

Lucas Novelli
Instituto Politécnico de Bragança and INESC Coimbra
Bragança, Portugal
a38680@alunos.ipb.pt

Paulo Melo
CeBER, University of Coimbra and INESC Coimbra
Coimbra, Portugal
pmelo@fe.uc.pt

Luisa Jorge
CeDRI, Bragança Polytechnic Institute and INESC Coimbra
Bragança, Portugal
ljorge@ipb.pt

André Koscianski
Federal University of Technology-Paraná
Ponta Grossa, Brazil
koscianski@utfpr.edu.br

***Abstract*—The current Internet of Things (IoT) solutions require support at different network layers, from higher level applications to lower level media-based support. This paper presents some of the main application requirements for IoT, characterizing architecture, Quality of Service (QoS) features, security mechanisms, discovery service resources and web integration options and the protocols that can be used to provide them (e.g. CoAP, XMPP, DDS, MQTT-SN, AMQP). As examples of lower-level requirements and protocols, several wireless network characteristics (e.g. ZigBee, Z-Wave, BLE, LoRaWAN, SigFox, IEEE 802.11af, NB-IoT) are presented. The variety of possible applications scenarios and the heterogeneity of enabling technologies combined with a large number of sensors and devices, suggests the need for simulation and modeling tactics to describe how the previous requirements can be met. As a potential solution, the creation of simulation models and the usage of the OMNET++ simulation tool to enable meaningful IoT simulation is discussed. The analysis of the behavior of IoT applications is proposed for two use cases: Wireless Sensor Networks (WSN) for home and industrial automation, and Low Power Wide Area (LPWA) networks for smart meters, smart buildings, and smart cities.**

Keywords—IoT; Application Protocols; WSN; LPWA; Simulation.

I. INTRODUCTION

The Internet of Things (IoT) enables the mix between the physical and informational world. Physical objects will be able to see, hear, think together, share information and coordinate decisions in a variety of domains, such as: transportation, healthcare, house and industrial automation, prevention to natural and man-made disasters, elderly assistance [2], intelligent energy manager and smart grids, automotive, traffic management [31], etc. Many communication protocols (e.g. 6LoWPAN, CoAP, DDS, MQTT-SN) and wireless standards (e.g. IEEE 802.15.4, BLE, LoRa, NB-IoT, SigFox) were created or modified to incorporate the IoT requirements, which include power efficiency, low throughput, and reliability. Therefore, in real-world IoT scenarios, many combinations of these networks and protocols can be applied to ensure optimal operation given the huge number and the heterogeneity of devices, many environmental factors, costs, and devices distribution.

A simulation model can evaluate whether the combination of technologies chosen is a good solution to a determined scenario, with quantitative and qualitative evaluation [7]. Two major IoT application areas are mentioned in this paper: The use of Wireless Personal Area Networks (WPAN) to enable Wireless Sensor Networks (WSN) for house and industry automation, and the use of Low Power Wide Area (LPWA) networks to construct solutions for smart cities, smart metering, and smart buildings.

The specific contributions of this paper include a summary of various technologies proposed for IoT communication many by organizations and companies. In a proposed simulation dimension, we intend to integrate within OMNet++ two different simulation platforms, the INET framework (for IP networks) with the Castalia Simulator (for WSN networks), in particular scenarios, defining topology, protocols to be implemented, and metrics to be collected.

The remainder of this paper is organized as follows. Section II describes the communication in the application level. Section III discusses the characteristics of the many network technologies and their application. A quick overview of simulation models is presented in Section IV, as well as a short description of the use cases simulation methods proposed. Finally, Section V provides some concluding remarks and future works envisioned.

II. APPLICATION COMMUNICATIONS

The protocols presented in this section are the most used solution for IoT networks and machine-to-machine (M2M) communication. They enable interaction between the devices and they try to offer solutions to IoT and M2M requirements.

A. Request/Response Approach

The Constrained Application Protocol (CoAP) is a web transfer protocol, specialized for constrained nodes and networks, structured for machine-to-machine communication, providing a request/response interaction model, built-in discovery of services and resources, based on Uniform Resource Identifier (URI) and Internet media types, easily interface with HyperText Transfer Protocol (HTTP), very low overhead and Datagram Transport Layer Security (DTLS), with asynchronous messages over User Datagram Protocol (UDP) [24].

The Extensible Messaging and Presence Protocol (XMPP) enables a near-real-time exchange of structured yet extensible data between any two or more network entities, using the Extensible Markup Language (XML). It consists in an

asynchronous, client-to-client or server-to-server exchanged of Stanzas among a distributed network of globally addressable over Transmission Control Protocol (TCP), being similar to email's architecture, with useful modifications to allow the communication in close to real time [23].

B. Publish/Subscribe Approach

This type of communication is widely used in constrained environments to decrease power consumption in the devices and provide less network traffic.

Data Distribution Service (DDS) is a Data-Centric Publish/Subscribe (DCPS) service for distributed application communication and integration and is based in a broker-less architecture, using multicast to bring high Quality of Service (QoS) and reliability, that suits for the real-time constraints for IoT [2]. Its architecture defines five entities in the protocol: (1) Publisher, responsible for the distributions of different types of data; (2) DataWriter, an interface between the Publisher and the application; (3) Subscriber, the receiver of the published data; (4) DataReader, an interface to handle the data in the Subscriber; and (5) Topic, controls their behavior [17].

Message Queuing Telemetry Transport for Sensor Network (MQTT-SN) [26], is a protocol that aims to connect embedded devices and networks with applications and middleware, it has one-to-one, one-to-many, many-to-many connection mechanism [2]. The protocol groups the devices in the network as the Subscriber, interested in receiving the information; the Publishers, produces the information; and the Broker, that coordinates the exchange between subscribers and publishers; there is also the Gateway, that intermediates the connection between the devices and the broker, offering direct or aggregated communication [26].

Advanced Message Queuing Protocol (AMQP) is an open protocol for business messages, structured in several layers: Type systems and encoding layers, applied support an interoperable data representation; The transport layer, consisting in frame-oriented connections, sessions, and links, operating between the devices; The message layer, that gives general information about the messages; There are also layers to control security, interactions, and behavior between the processes [16].

CoAP introduces an observer design pattern, in which the node asks the server to notify it about any state change [4].

C. Discovery Services

This process consists of finding, configuring and enabling communication between devices that can provide services and information, without human interface, in a dynamic topology context, in which changes occur frequently.

There are two solutions based on the Domain Name System (DNS) that can provide discovery services: the DNS Service Discovery (DNS-SD) [6] and Multicast DNS (mDNS) [5]. With DNS-SD, given a service and a domain, it allows clients to discover a list of named instances of that service, offering a good service discovery with minor effort. mDNS can perform DNS-like operations on the local link to find a service, with the benefits of requiring little or no configuration setup and being resilient to infrastructure failures [5].

CoAP servers are encouraged to provide a resource description, via the well-known URI (/.well-known/), with this URI a device can obtain access to the client description with a GET request [4]. Multicast service discovery is also supported in a CoAP network, devices have to be prepared to receive this requests, but they can ignore it if this kind of discovery is not desired [24].

In a MQTT-SN environment, gateways can announce its presence by broadcasting an ADVERTISE message, in intervals greater than 15 minutes (to avoid congestion), and the devices must update their list of active gateways. Is possible for the device to send SEARCHGW message to the network, whereupon the other devices can respond with an active gateway ID (or the gateway itself can send it) [26].

With the DDS protocol, local entities in the network exchange their properties (e.g. identifying keys, addresses, ports, QoS parameters) periodically, to maintain an updated database with this information, and when a remote entity is discovered a matching is made between local and remote properties, and this new remote entity will be allowed to communicate only if this matching process is completed and both entities agreed with the communication [17].

D. Quality of Service

WSNs are also classified as Low power and Lossy Networks (LLN) and are unreliable because the communication could be disrupted by many obstructions, noises, and interferences [29], therefore the protocols for those networks must have a built-in mechanism to ensure reliable communication.

MQTT-SN offers three levels of QoS: Level 0: best-effort, the message can be delivered or not, no retransmission or acknowledgment is defined; Level 1: the message is retransmitted until the acknowledgement is received; Level 2: the message is received one and just one time [26].

Focusing on a message-oriented environment, AMQP supplies delivery guaranties primitives, including at-most-once, at-least-once and exactly once [2].

Working over UDP, CoAP messages can be lost, duplicated or arrive out of order, and to achieve reliability the protocol implements four types of messages: confirmable, non-confirmable, acknowledgment and reset, which requires a response, not requires response, confirms of the reception, and tells that it cannot be processed, respectively [18].

Focusing on each entity, DDS provides many QoS policies, being able to control when and how the communication occurs between entities, implemented by the Real-Time Publish-Subscribe (RTPS) Protocol, is also possible to configure a deadline for communication, specify a maximum delay between creation and use of the data, determinate the quantity of time that an entity is active, and much more [17].

E. Security

AMQP, XMPP, MQTT-SN, and DDS deliver authentication and/or encryption for transporting the data communication in the network, provided by the TLS and SSL standards [17] [26] [16] [23], whereas CoAP uses DTLS [24].

F. Web integration

The requirements presented in constrained networks created many proprietary protocols and link-only connections, presuming that TCP/IP family needed too much memory and

bandwidth for them to scale as necessary. HTTP, that is widely used in conventional networks, is unsuitable for IoT solutions, given the headers overhead [31].

traffic between devices inside LLN, from a control point to a set of devices in the LLN and vice-versa. The protocol also tries to avoid loops in the network, which could cause packet

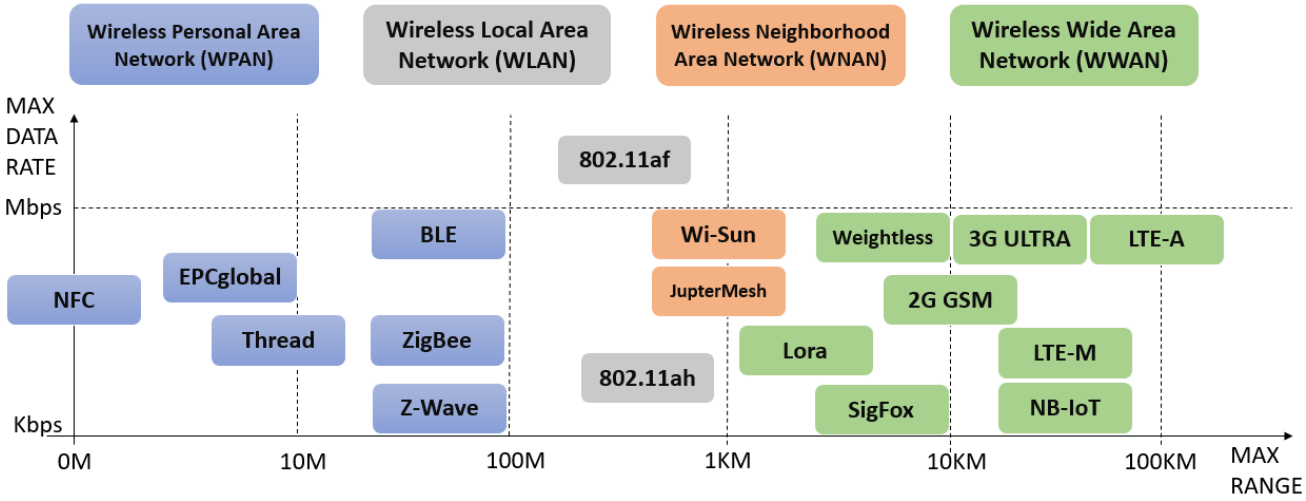

Fig. 1. Networks standards applicable at IoT solutions and their data rate and ranges.

UDP is defined to make available a datagram mode of packet-switched communication in a network and provides a minimal, unreliable, best-effort, message-passing transport to applications, which has demonstrated a good trade-off in terms of energy cost and reliability [18]. TCP is a connection-oriented, end-to-end protocol designed to support multi-network applications, it implements flux and congestion control, and has mechanisms to assure a reliable delivery, but it does not scale well on constrained devices [31] because of the amount of data required for traffic control and reliability [18].

The integration between CoAP and HTTP is provided by proxies that can convert single HTTP request into multiple CoAP requests for the nodes in the WSN and vice-versa. Besides, the mapping of the of the URIs is quite straightforward, having nodes that understand HTTP in one side and CoAP in the other [4]. Being also designed on the Representational State Transfer (REST) architecture, which relies on consumption and expose of resources defined by URI without ambiguity, the conversion between CoAP and REST is simple [2].

While the adaptation of IPv6 allows addressing a huge amount of devices, it also introduces overheads not supported in constrained solutions [31]. A solution proposed is IPv6 over Low power Wireless Personal Area Networks (6LoWPAN), that has 3 principles: I) Header compression performed by link-level or context assumptions, to compress the IPv6 to 2 bytes and UDP headers to 4 bytes; II) Fragmentation to support multiple datagram transmissions, to accommodate the IPv6 minimum MTU; III) Layer-two forwarding for carrying the link-level addresses to the end of an IP hop [12].

G. Routing

The Routing Protocol for LLN (RPL) is a distance vector routing protocol for IPv6 that uses the physical topology as a Directed Acyclic Graph (DAG) to create one or many Destination Oriented DAG (DODAG), a logical topology created using a set of metrics and constraints [29]. It supports drops and link congestion, failures can be repaired in two distinct forms: I) The global repair that reboots the whole graph and is done by the root node; II) The local repair can be realized by any node and consists in find another parent [29].

RPL implements timer management using a mechanism called “trickle timer”, where a timer is increased when the network is in a stable state, avoiding useless check messages exchange; when a node detects an inconsistency this timer is reset and the node warn the others about it [29]. Table 1 provides an overview on different application protocols properties.

III. NETWORK LEVEL COMMUNICATION

IEEE 802.15.4 is the most prominent standard for LR-WPAN, it defines both the Physical Layer (PHY) and Medium Access Control Layer (MAC) from the OSI Model, and delivers a good solution for energy-efficiency, range and data rate [18]. The first release of the protocol was in 2003 and it has been periodically updated (2006, 2011 and 2015).

TABLE I. APPLICATION PROTOCOLS SUMMARY

Protocol	Transport	QoS	Architecture	Security
CoAP	UDP	Yes	Request/Response	DTLS
MQTT-SN	TCP	Yes	Publish/Subscribe	TLS/SSL
XMPP	TCP	No	Request/Response	TLS/SSL
AMQP	TCP	Yes	Publish/Subscribe	TLS/SSL
DDS	TCP	Yes	Publish/Subscribe	TLS/SSL

In IEEE 802.15.4 networks, the devices are typed as a Fully-Function Device (FFD), that can be either the coordinator which is responsible for the creation, control, and maintenance of the network [2] or other unconstrained node, or as a Reduced-Function Device (RFD) which are constrained and must be associated with a single FFD at each time. Two topologies are defined: Star or peer-to-peer. An overview of solutions discussed in this section is presented in Figure 1.

A. WSNs for Home, and Industrial Automation

Integrating smart objects into the home and industrial plants can track resources consumption, monitor, and control to make it more efficient in fields such logistic and supply chains. In these scenarios a wireless network focused on low-data-rate, short-range, and focus on energy efficiency is widely applied.

Many companies proposed solutions for the specifications described above such ZigBee Alliance, Sigma Designs, and Thread with their standards ZigBee, Z-Wave, and Thread, respectively, all of them based on the IEEE 802.15.4 specification. ZigBee is composed of four main layers: the PHY, MAC, network layer, and application layer with a security cross layers functionality is also provided [9]. Z-Wave has the main propose of allowing reliable transmission of a short message from a control unit to one or more nodes in the network, operating in the 900MHz or 2,4GHz band, having 40Kb/s or 200Kb/s data rate, respectively [9]. It should not be used for transfers of large amount of data, streaming or timing critical data. Thread standard provides reliable, cost-effective, low power, wireless device-to-device communication [27].

Differing from Z-Wave, Thread, and ZigBee, Bluetooth Low Energy (BLE) provides a solution in scenarios where a single-hop solution is required, with implantation similarities from the Bluetooth, operating in the 2.4GHz ISM and achieves over 1Mbps of data-rate [8].

Near Field Communication (NFC) is a short-range half-duplex communication protocol, providing easy and secure communication. The ISO 14443, ISO 18092 and FeliCa standard defines the operation of the NFC at 13.56MHz band [30]. It is useful when distances are very short, at most 10 centimeters, and relatively higher data rate, supporting transmissions up 424kbps.

The Electronic Product Code (EPC) can carry a larger amount of information, being electronic read over distances up 10 m, even when not visible, and it may be widely applied for device identification and passive communication. The EPC protocol defines physical and logical layers for a passive-backscatter, interrogator-talks-first (ITF), RFID system, that operates in the 860MHz – 920MHz frequency range [11].

B. Smart Metering, Smart Cities, and Smart Builds

Smart meters can be electricity, gas or water meters that are intelligent and automatic, providing efficiency for companies and control for customers. In this context, devices will be inside houses and buildings, making the communication difficult, and thousands of devices in a small area will need to be connected to a single base station.

Options to fulfill the requirements of wireless connection in a range of few meters up to many hundreds of meters with data rate up to Mbps in regions where the TV White Space (TVWS) spectrum is free are the IEEE 802.11af and the Weightless-W, with more effective penetration through obstacles than 2.4 and 5.7 ISM bands [3]. IEEE 802.11af enables devices to share the operating between 54 MHz and 790 MHz with the Registered Location Query Protocol (RLQP), enabling the stations to select spectrum, power, and bandwidth allowed by their regulation domain. Weightless-W is provided by the Weightless SIG and uses the TVWS (470–790 MHz) to give a solution that poses bidirectional communication, adapting dynamically the rate and the range to the network needs [10], achieving data rate between 1Kbps and 1Mbps [21], with battery autonomy between 3 and 5 years.

In regions where TVWS is not available or the specifications of data rate, cover range or energy exceed the application needs, the IEEE 802.11ah standard can be used to provide extended range Wi-Fi [20] and provide low power communication operating in the sub 1 GHz band and offering minimum 100 kbps of data rate with up to 1 km of range in outdoor areas [3].

The Wireless Smart Utility Network (Wi-SUN) Alliance and the ZigBee Alliance aim to establish and promote their standards based on IEEE 802.15.4 for smart metering needs, i.e. Wi-SUN and JupiterMesh, respectively. Wi-SUN is already operating in Japan and Korea and define many different protocols depending on the equipment and the application field [14]. JupiterMesh is not released yet but it aims to operate with data rates up to 800Kbps and implement IP solutions including 6LoWPAN and RPL.

The LPWA networks possess the ability to offer low-cost connection for huge number of low-power devices distributed over large areas, with a cost per device is under $5 and connection subscription per unit as low as $1 [21], begin the main factor of the adoption of it for the proposed to let the city's infrastructure connecting and sensed. Integrate this category the LoRaWAN, the SigFox, the Ingenu, the Weightless-N, and the Weightless-P Standards applying Ultra-Narrowband (UNB) and Direct Sequence Spread Spectrum (DSSS) as modulation schemes for PHY which provides excellent coverage and MAC star topology with random access.

In environments where requirements for data rate, payload and message exchange are very low and the coverage area is up to few kilometers, SigFox suits well. It consists of a proprietary UNB solution, operating in 869MHz (Europe) and 915MHz (North America) bands. It is based on Random Frequency and Time Division Multiple Access (RFTDMA) and achieves 100 bps for uplink, 12 bytes of payload and but its devices can't exceed 140 messages in a day [1]. The network architecture is based on end-devices communicating to a SigFox Network Operator, who is proprietary and are equipped with cognitive software-defined radios, connected with back-end server via an IP connection [21].

For implementations that required unlimited message exchange and hundreds Kbps data rate is advisable to consider the LoRaWAN, Weightless-N, and Weightless-P specifications. LoRaWAN is standardized by LoRa Alliance Consortium and consists in the LoRa, which is a PHY proprietary scheme developed by Semtech Corporation, allows multiple data rates, with the bandwidth and spreading factor configurable [21]. Its MAC and above layers are open source, offering a fully bidirectional symmetrical link between endpoint and gateway, where this endpoint are projected to operate up to 10 years [15], uses symmetric-key cryptography to end devices authentication, protecting and providing privacy to the data in the network [21]. Weightless-N is a UNB standard for one-way communication from end devices to a base station [21], performing transmission in the sub-GHz ISM bands, achieving up to 5kms range with a data rate between 30 Kbps and 100Kbps [10]. Weightless-P offers two non-proprietary PHY with two-way connectivity. It permits to enhance the reliability by using acknowledgment protocols [10], reaching data rate between 200 bps and 100 Kbps [21].

Ingenu is proprietary, uses the ISM 2.4 GHz band, delivering higher speed than others used in LPWA presented above, 624Kbps uplink and 156Kbps downlink, this being its principal feature [1], supporting up to 1000 simultaneous users [10].

The Third Generation Partnership Project (3GPP) also provides a solution that uses the present infrastructure used for mobile communication to enable IoT communication, within a coverage area of tens of kilometers. 2G technologies (i.e. GSM, GPRS/EGPRS) is ideal in terms of power consumption and cost per unit, while the 3G family (i.e. UMTS, HSPA) is not power efficient, has higher cost and its network capability exceeds IoT requirements [19]. 4G technologies (i.e. LTE and LTE-A) are interesting to IoT, gathering demand and scalability requirements, but the costs involving a 4G device still is a problem [19]. To solve the problems in regular LTE, LTE-M (or enhanced MTC) and NarrowBand IoT (NB-IoT) Standards were created with new categories of user equipment (UE): CAT-1, CAT-0, CAT-M1, CAT-NB1, Further enhanced MTC (FeMTC), and Enhanced Narrow Band IoT (eNB-IoT). These new UEs have just one antenna, implement half or full-duplex mode, reducing complexity, and promoting battery longevity. With 5G, 3GPP wants to implement full M2M communication with two distinct solutions: massive machine-type communication (mMTC) and ultra-reliable machine-type communication (uMTC) which gives 5G connection for devices with constrained power use and for ultra-reliable/time efficient needs, respectively [25], but these IoT-designed solutions are expected to be released only by 2020.

IV. Simulation Models and Case Study Proposal

A simulation approach appears to be an adequate way to analyze the behavior of IoT solutions, including the many protocols that can be applied in WSN networks (the base to home and industrial automation), or in LPWA networks (an emerging technology to enable wide area communication to battery-powered devices requiring high power efficiency, enabling the smart cities and smart metering scenarios). Discrete Event Simulation (DES) provides a model, composed by variables representing the state of the simulation, but it may, however, display poor scalability for IoT scenarios [7]. The Parallel Discrete Event Simulation (PDES) and Parallel and Distributed Simulation (PADS) divides the simulation model, processing each part in different CPUs or different hosts, however these approaches bring some drawbacks including complex partitioning, synchronizations algorithms, and is required a data distribution management [7].

Another possible approach would be the physical implementation of the WSN and LPWA networks, however, this option would impose much higher costs, an inflexible solution in terms of topology, setups, and configuration. Also, the possible combinations of application protocols would be restricted to the devices capability and the time to collect and analyze all the real traffic generated in the network would be another restriction.

The simulation case studies proposed will comprise

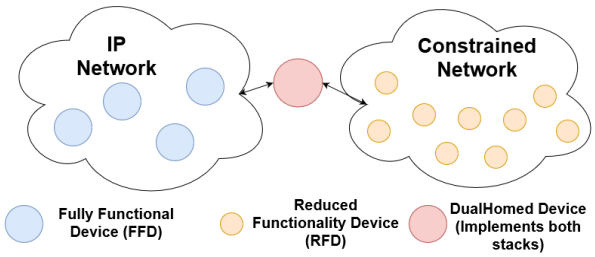

Fig. 2. Idealized topology for case studies

discovering the key measures related to each kind of application simulating using different (suitable) technologies and tying the results to the technologies used in particular settings. Some characteristics of these applications and the kind of technologies to consider in their analysis are presented below.

A. Case Study A: WSNs for Home, and Industrial Automation

WSN enables home automation in a setting where each node in the house (constrained device) interacts with a home hub with the capacity to store and process data and act as communication gateway with other networks [22].

In the industrial context, RFIDs and WSN already have been used in supply chain, automated production line, and in production plants. With the continuous adoption of the concept of Industry 4.0, IoT resources will be applied to allow factories to support more flexible production processes, enabling remote monitoring and control, as well as providing mechanisms to collect and analyze data created. IoT applications are evolving and growing in fields that include healthcare, mining production, transport and logistics, firefighting, and food supply chain.

To verify if the requirements are being fulfilled in an industrial/home automation environment is proposed the modeling of the WSN enabling IoT for these scenarios. The devices in this model will be based on IEEE 802.15.4 specification, thus networks that cover up tens of meters (i.e. ZigBee, Z-Wave, BLE, Thread) will also be modeled and implemented. Auxiliary equipment (i.e. routers, access points) to enable WSN to connects to conventional web will also be included in the models, for closer representation of the real world.

B. Case Study B: Smart Metering, Smart Cities, and Smart Builds

Smart Metering/Smart Grids scenarios wireline/less are used to enable communication. These scenarios can be organized in three sub-networks, one related with the customer (i.e. WPAN), other related to sending/receiving measurements, the distribution (i.e. WLAN and WNAN), and finally one related with transmission/generation (i.e. WWAN) [13]. Smart Building scenarios are designed to maintain the structural health of buildings. In the scope of Smart Cities (e.g. waste management, air quality monitoring), can increasing the quality and enhancing of the services for citizens and reduce operational costs to city administration [31].

Those applications of IoT require long-range coverage, with thousands of devices communicating in the same network, but usually exchange few messages with reduced payload size, since they need to focus on energy efficiency to achieve years of battery duration. By modeling the long-range networks such LPWAs, TVWS based, and 3GPP solutions, it may be possible to find out if these networks can offer support to the identified needs. Since these technologies will generally be applied in outside environments, with many noise and interruption sources, those should be added to the model and considered upon data analysis.

Finally, case studies A and B will have their network's traffic generated using protocols designed for IoT. With this simulation model we expect to obtain quantitative and qualitative data, characterizing the behavior of the IoT-specified constrained devices, other conventional equipment, and the whole network. Analyzing the collected data we intend to check the IoT requirements, in metrics which can include general network metrics (e.g. packet loss rate), network data flow, bottlenecks, battery-powered devices life cycles, or reliable communication.

C. Current Development

The implementation of the framework is being performed in OMNeT++ [28], an open source object-oriented discrete-event simulator based on the C++ language. To support both dimensions of an IoT network, the INET Framework, which provides an implementation of IP components and protocols, will be used to create the IP network, whereas the Castalia Simulator will be used to enable the constrained IoT devices (wireless) network, as can be seen in Figure 2.

We intend to support the two frameworks interconnection by creating DualHomed devices, which will provide a gateway between the two networks, requiring that both stacks are present in one device. Implementing the most prominent protocols stacks (e.g. 6LoWPAN, CoAP) for the RFD devices is also a task that will be performed. In this way, we expect to collect several different metrics throughout the full network, which can include: transference data-rate; number of packets transmitted, received and lost; and power consumption.

V. CONCLUSION

This paper briefly discusses the most common requirements for IoT applications, and the communication protocols used to provide them, focusing on their similarities, capacities, and issues in a computational constrained and power efficient environment. We briefly describe the key features that are expected from these communication service support mechanisms, including security, web integration, QoS, discovery services, architecture, and routing.

The key network technologies enabling personal, local and wide range wireless communication in IoT environments are also summarized, to offer an overview of their technical capabilities that include: data rate transmission, covered area, band frequency in use in many geographical areas, security features, and companies that standardize these solutions.

Overall boundaries for a simulation case study using WSN for home and industrial automation (where already many solid research and commercial solutions are already being applied) are proposed. A second case study is proposed for Smart Metering/Smart Grids, Smart Cities, and Smart Builds, areas that are acquiring the focus of researchers and many companies that are proposing standards to enable commercial application.

A simple description is provided of the components that will be applied to the simulation models. The simulations models aim to support different combinations, what-if scenarios, and environment changes using a wide range of application protocols and network standards being developed for IoT. The proposed implementation in OMNeT++ of the described case studies will be developed as future research.

Acknowledgement: This work has been partially supported by FCT under project grant UID/MULTI/00308/2013.

REFERENCES

[1] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the Limits of LoRaWAN," *IEEE Commun. Mag.*, vol. 55, no. 9, pp. 34–40, 2017.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutorials*, 2015.

[3] B. Bellalta, L. Bononi, R. Bruno, and A. Kassler, "Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges," *Comput. Commun.*, vol. 75, pp. 1–25, Feb. 2016.

[4] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An Application Protocol for Billions of Tiny Internet Nodes," *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, Mar. 2012.

[5] S. Cheshire and M. krochmal, "Multicast DNS," 2013.

[6] S. Cheshire and M. Krochmal, "DNS-Based Service Discovery," Feb. 2013.

[7] G. D'Angelo, S. Ferretti, and V. Ghini, "Modeling the Internet of Things: a simulation perspective," in *2017 International Conference on High Performance Computing & Simulation (HPCS)*, 2017, pp. 18–27.

[8] C. Gomez, J. Oller, and J. Paradells, "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology," *Sensors*, vol. 12, no. 12, pp. 11734–11753, Aug. 2012.

[9] C. Gomez and J. Paradells, "Wireless home automation networks: A survey of architectures and technologies," *IEEE Commun. Mag.*, vol. 48, no. 6, pp. 92–101, Jun. 2010.

[10] C. Goursaud and J. M. Gorce, "Dedicated networks for IoT : PHY / MAC state of the art and challenges."

[11] GS, "EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz."

[12] J. W. Hui and D. E. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks," *IEEE Internet Comput.*, vol. 12, no. 4, pp. 37–45, Jul. 2008.

[13] Y. Kabalci, "A survey on smart metering and smart grid communication," *Renew. Sustain. Energy Rev.*, vol. 57, pp. 302–318, May 2016.

[14] T. Kato, "Standardization and Certification Process for 'Wi-SUN' Wireless Communication Technology," no. 1, 2016.

[15] K. E. Nolan, W. Guibene, and M. Y. Kelly, "An evaluation of low power wide area network technologies for the Internet of Things," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 439–444.

[16] OASIS Standard, "OASIS Advanced Message Queuing Protocol (AMQP) Version 1.0," 2012.

[17] OMG, "Data Distribution Service (DDS) Brief," no. April, pp. 1–20, 2015.

[18] M. R. Palattella *et al.*, "Standardized Protocol Stack for the Internet of (Important) Things," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1389–1406, 2013.

[19] M. R. Palattella *et al.*, "Internet of Things in the 5G Era: Enablers, Architecture, and Business Models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[20] M. Park, "IEEE 802.11ah: sub-1-GHz license-exempt operation for the internet of things," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 145–151, Sep. 2015.

[21] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[22] B. L. Risteska Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *J. Clean. Prod.*, vol. 140, pp. 1454–1464, Jan. 2017.

[23] P. Saint-Andre, "RFC 6120 - Extensible Messaging and Presence Protocol XMPP,: Core," 2011.

[24] Z. Shelby, K. Hartke, and C. Bormann, "RFC 7252 - The Constrained Application Protocol CoAP,," 2014.

[25] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14–21, Mar. 2017.

[26] A. Stanford-Clark and H. L. Truong, "MQTT For Sensor Networks (MQTT-SN) Protocol Specification," 2013.

[27] Thread Group, "Thread 1.1.1 Specification." 2017.

[28] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," *Proc. 1st Int. Conf. Simul. tools Tech. Commun. networks Syst. Work.*, p. 60, 2008.

[29] J. Vasseur, C. Fellow, C. Systems, N. Agarwal, T. Leader, and J. Hui, "RPL: The IP routing protocol designed for low power and lossy networks Internet Protocol for Smart Objects (IPSO) Alliance," 2011.

[30] R. Want, "Near field communication," *IEEE Pervasive Comput.*, vol. 10, no. 3, pp. 4–7, 2011.

[31] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

ANNEX 2 – OMNeT++ Configuration Files

IP Network Simulation Configuration

```
[Config Wireless01]
description = Two hosts communicating wirelessly
network = WirelessA
sim-time-limit = 100h
warmup-period = 120s
*.host*.networkLayer.arpType = "GlobalARP"
*.hostA.numUdpApps = 1
*.hostA.udpApp[0].typename = "UDPBasicApp"
*.hostA.udpApp[0].destAddresses = "hostB"
*.hostA.udpApp[0].destPort = 5000
*.hostA.udpApp[0].messageLength = 1000B
*.hostA.udpApp[0].sendInterval = exponential(12ms)
*.hostA.udpApp[0].packetName = "UDPData"
*.hostB.numUdpApps = 1
*.hostB.udpApp[0].typename = "UDPSink"
*.hostB.udpApp[0].localPort = 5000
*.host*.wlan[0].typename = "IdealWirelessNic"
*.host*.wlan[0].mac.useAck = false
*.host*.wlan[0].mac.fullDuplex = false
*.host*.wlan[0].radio.transmitter.communicationRange = 500m
*.host*.wlan[0].radio.receiver.ignoreInterference = true
*.host*.**.bitrate = 1Mbps
```

WSN Network Simulation Configuration

```
include ../Parameters/Castalia.ini
sim-time-limit = 100h
warmup-period = 120s
SN.field_x = 100                                # meters
SN.field_y = 100                                # meters
SN.numNodes = 36
SN.deployment = "6x6"
SN.node[*].Communication.Radio.RadioParametersFile = "../Parameters/Radio/CC2420.txt"
SN.node[*].Communication.Routing.collectTraceInfo = true
SN.node[*].ApplicationName = "ValueReporting"
SN.node[3].Application.isSink = true
SN.node[*].Application.destination = "hostB"
[Config RoutingProtocolName]
SN.node[*].Communication.RoutingProtocolName = ${RoutingProtocolName = "MultipathRingsRouting", "BypassRouting"}
[Config MacProtocol]
SN.node[*].Communication.MACProtocolName = ${MACProtocolName = "TMAC", "Basic802154"}
[Config varyDutyCycle]
SN.node[*].Communication.MAC.dutyCycle = ${dutyCycle= 0.02, 0.05, 0.1}
[Config varyBeacon]
SN.node[*].Communication.MAC.beaconIntervalFraction = ${beaconFraction= 0.2, 0.5, 0.8}
[Config varyTxPower]
SN.node[*].Communication.Radio.TxOutputPower = ${TXpower="-1dBm","-5dBm"}
```