Escola d'Enginyeria de Telecomunicació i Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

MASTER THESIS

TITLE: Implementation of a Wireless Sensor Network for agricultural monitoring for internet of things (IoT)

MASTER DEGREE: Master's degree in Applied Telecommunications and Engineering Management (MASTEAM)

AUTHOR: Juan Ernesto Granda Rodríguez

ADVISOR: José Polo Cantero

DATE: July, 2nd 2019

Title: Implementation of a Wireless Sensor Networking for agricultural monitoring for internet of things (IoT)

Author: Juan Ernesto Granda

Advisor: José Polo Cantero

Date: July 2, 2019

Abstract

The agricultural sector is the first industry and the most impacted by the climate changes. The delicate environment that must manage some type of crops required the constant monitoring and maintenance of the greenhouse.

The Internet of Things (IoT) gives a new alternative for real time environmental monitoring of variables such as temperature, humidity and solar irradiation that can contribute for the healthy growth of the crop, and also impact for the plagues and sickness presence.
The objective of this thesis is built a Wireless Sensor Network using radiofrequency devices and environmental sensors.

The limitations of this master thesis are: the location of the sensor node, the external conditions that will not impact the network, the simulation, test and pilots that are deployed in a controlled space.
The wireless sensor network proposed employs the Zolertia Motes using IEEE802.15.4 standard. This device allows low power consumption, as the nodes must be located in places where it may handle several weeks without change depending on the autonomy of their batteries cell.

The network protocol managed works over low consumption, as same as the transmitted and received packets of data. The standard used on this project is the 6LowPAN.
The network configured works over the stack protocol IPv6 so that all the devices handled UDP and manage this internet package.
The Raspberry Pi 3 B will work as border router between the sensor nodes and the exterior considered as Internet using the IPv4 standard internet router protocol.
The framework used for the network implementation is ContikiOS installed on the gateway and tested using one mote located in the lab. The data managed in this experiment has low data rate as this measurement do not require a permanent monitoring and high speed. The atmospherically changes are not variant enough to be observed constantly. The sample rate will be 1 package each 10 minutes.

This project aims to develop a full network implementation since the mote until the dashboard.

CONTENT

INTRODUCTION **1**

CHAPTER 1. STATE OF THE ART **2**

1.1. Technology Evolution **2**
1.1.1. Communication 2
1.1.2. Devices 3

1.2. Internet of Things **4**
1.2.1. Short range network 6
1.2.2. Long range network 11

1.3. Wireless Sensor Networks **14**

1.4. Communication protocols **15**
1.4.1. 6LoWPAN 15
1.4.2. IPv6 16
1.4.3. UDP/TCP 17
1.4.4. RPL 17
1.4.5. MQTT 18

1.5. Related projects **19**

CHAPTER 2. TECHNOLOGY SELECTION **20**

2.1 Preliminary agricultural information **20**

2.2 Selection features and devices **21**
2.2.1 Single Board Computer 21
2.2.2 Network protocol 22
2.2.3 Transceiver modules 23
2.2.4 Sensor 24

CHAPTER 3. SYSTEM IMPLEMENTATION **26**

3.1 Operative System **26**

3.2 Installing and configuration **27**

CHAPTER 4. EVALUATION AND RESULTS **33**

4.1 Simulated scenario **33**

4.2 Laboratory scenario **34**

4.3 Errors and solutions **35**

4.4 Results **36**
4.4.1 Agropolis results 37
4.4.2 Laboratory results 39

CHAPTER 5. CONCLUSION AND FUTURE WORK **41**

5.1. Conclusions 41

5.2. Future work 42

REFERENCES 43

ANNEXES 47

INTRODUCTION

The main purpose of this document is to provide a guideline to implement an IoT network for the agriculture industry. In addition, the network installed will help gathering environmental information about crops in the greenhouse or controlled areas. The farmers will be able to monitor the weather impact on the crop, acquiring relevant variables as humidity and temperature, indicating the vapor saturation of a specific sector.
The system has the possibility to support several nodes at a time. The maximum value that this wireless network can handle is less than 1000 nodes, more than enough for a complete wireless ecosystem. At this point, the results of this thesis will implement a point to point network with the capability to extend the number of nodes. The content of this document will follow the next structure:

1. The technology impact on the Internet of Things and Wireless Sensor Networks. Telecommunication background in the last few years with the implementation of radio connectivity and electronic evolution, shows the network's protocol overview with advantages and limitations of each type of technology regarding the final application. The last section of this chapter will explain how communication protocols were selected in relation to their direct impact on this project, and the order of the index aims to follow from lower to high level the OSI model.

2. Devices and technology selection to implement the monitoring. The relevant features took into account the design and selection of the parts, in general bounded by the laboratory device's availability. The key aspects to consider for the implementation of a Wireless Sensor Network are the electrical characteristic of the elements, the basic device information and fabricant recommendations. The communication between motes are standardized by the IEEE and IETF [1](Internet Engineering Task Force).

3. The operative system configuration and mandatory software requirement to create the net. The transceivers used in this project have been selected previously regarding the laboratory availability and disposition of the devices based on previous projects implementation with those motes[2][3].

4. The nodes assume a CSMA/CA configuration, to avoid collision between the simulation scenario and the real application. After that, the implementation will be execute with one mote regarding the DHT22 sensor availability. The Zolertia RE-MOTE uses a module CC2538[4] transceiver provided by Texas Instrument and equipped with a monopole antenna and debugging system in order to control the transceiver.
The gateway as border router will interact with the MQTT broker. The Raspberry will route the package using the 6LoWPAN protocol through a tunnel connected in a bridge to the Ethernet board. The data frame will be visualized in the dashboard to the farmer. At the end of this chapter, the results will show the behavior between the laboratory and greenhouse data.
The last section will focus on conclusions and future work showing the possible business impact into the agricultural market and the reliability of the system.

CHAPTER 1. STATE OF THE ART

1.1. Technology Evolution

1.1.1. Communication

In the last ten years, the presence of the communication sector has been increasing in the industry on personal devices. The market impact depends on the criteria and requirements of the customers. The development of new protocols for emerging communication products, follows the business expansion regarding companies multiple necessities, therefore all kind of industries adopted new technological trends (see Figure1*)*. The evolution consists in a tradeoff between the user necessities and the technology advance.

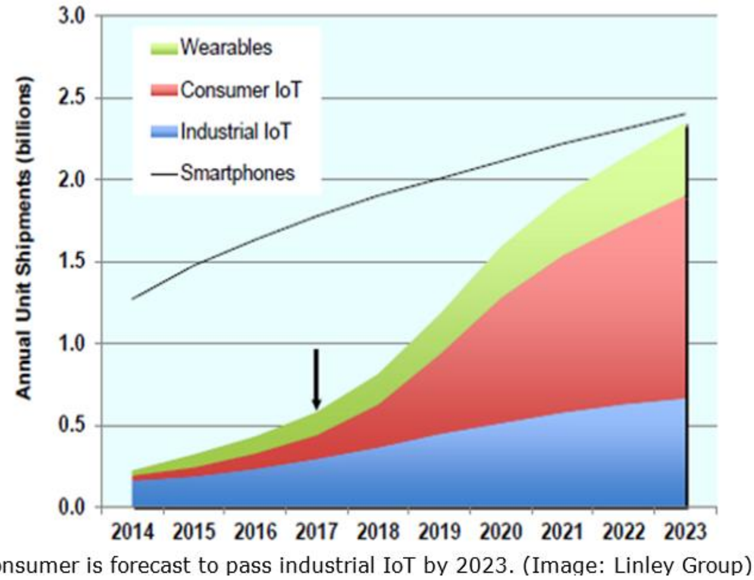


Figure 1. IoT market tendency

The information's age and mass market management required big data acquisition. The input obtained is highly rentable in terms of future business movements. The networks are configured regarding the final application given by the customer or final user of the system. The telecommunication technology has two main physical channels to transport the data, wired and wireless connections. The wired connections require a physical channel creating a real link between two or more devices (e.g. Ethernet, optical fiber, etc.).

The wireless connections are systems that use radiofrequency waves as communication channel to establish a link between the devices.
This type of communication give mobility advantages and freedom depending on the capacity of the channel. The wired connections are faster and reliable over a tangible channel, but the cost of installation and material require resources that the customer cannot achieve, neither leased nor own. An example of a cellular network as C-RAN or 4G requires the installation of fix access points, remote radio head and baseband unit, otherwise the connection must be done by Ethernet and Optical fiber.
On the other hand, mobile applications and difficult access location make the wireless communication a good option. The wireless devices have a low cost

compared to the fixed ones, and the freedom they provide is higher even though the performance and reliability decrease substantially.
The Figure 2 shows the interconnection between devices, better known as M2M (Machine to Machine), predicted to grow in a big rate in comparison of the population growth. It means that one person can manage more than one device.

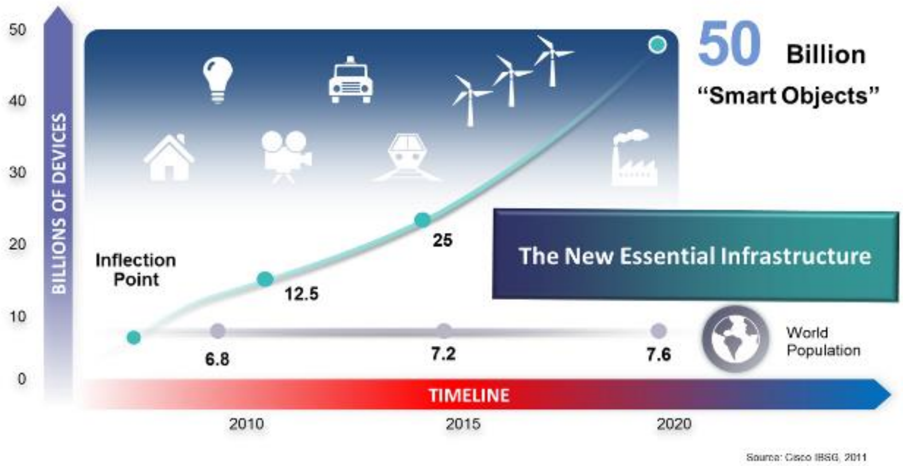


Figure 2. Prognostic of interconnected devices in the world (source: [5]**)**

1.1.2. Devices

The technological advances come with the introduction of new solutions conceived by design's engineers and researchers. The electronic devices increase the commutation process using transistors while the scale of the junctions is reducing. The miniaturization follow the Moore's law which consist in reducing the transistors size about 50% each year and a half. Consequently the quantity of silicon parts per chip increase in a logarithmical scale[6] showed in Figure 3.

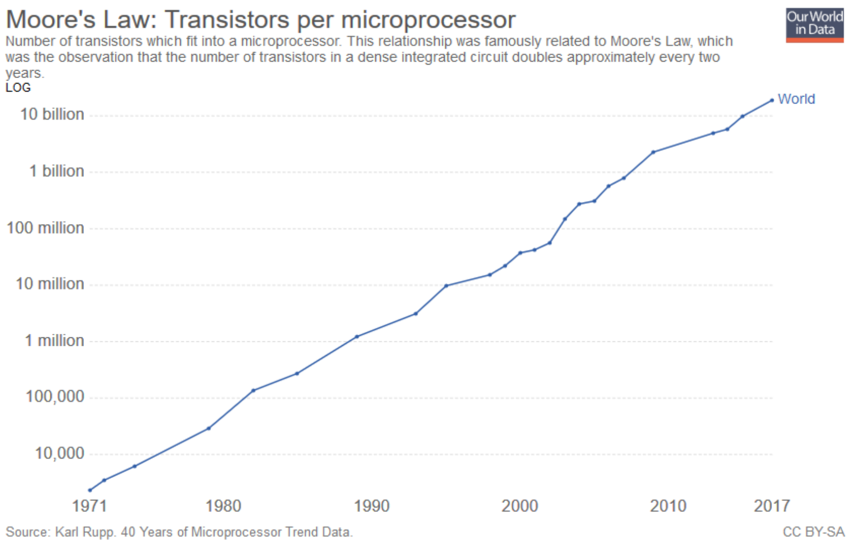


Figure 3. Number of transistors into a microprocessor in logarithm scale.

(Source : https://ourworldindata.org/technological-progress)

The microcontroller based on ARM architecture has a low cost of production, heat and power consumption, making this configuration rentable for battery supply[7]. Concerning the Smart Objects for the Internet of Things, they require low power consumption and high performance regarding the data rate managed in the device. The project will focus on portable electronics and the performance on the field. This kind of applications require power management control (PMC) and the range of action where the network will perform.

The power consumption of the devices depends on the instructions per second (IPS) that the microcontroller or microprocessor manages. In the wireless systems the energy is a critical aspect, and the autonomy of a network is managed by the MAC protocol programmed on the chip.

High performance of the devices represents a major consumption. Nowadays the transistor design has arrived until a maximum miniaturization where the current leakage on the devices can represent a main problem of power consumption. New transistors design had been proposed to confront the leakage.

1.2. Internet of Things

Internet of Things can be classified into different types of spectrum, the licensed and the exempted of charges. The objective is to interconnect smart devices depending on the technology and configuration used.
The network protocol selection depends on three parameters: Power consumption, data rate and range of the link. In agricultural applications, for example, the power consumption is a main issue due to the low maintenance and energy availability on the crop, and therefore the battery autonomy must be extended as long as possible.

In Figure 4, the relationship between the data rate and the range for each particular protocol is shown. The main overview of the classical protocols are used in the industrial and consumer sector. Forward, in this section will appreciate the advantages and disadvantages of the protocol regarding the end user.

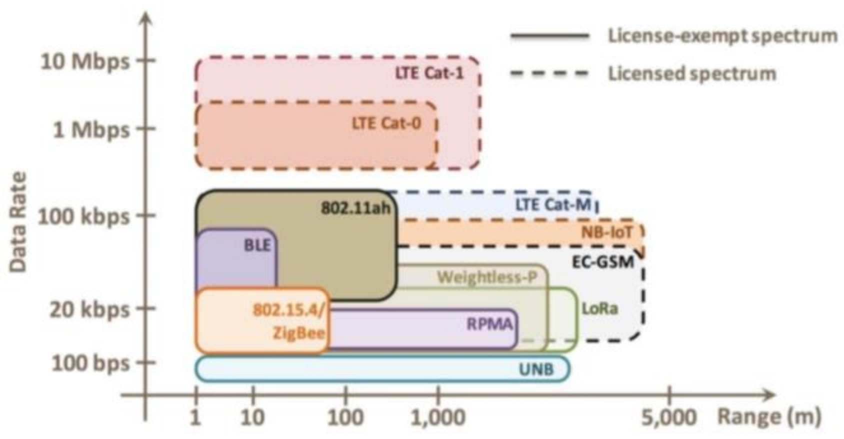


Figure 4. Relation between data rate and range in meters of LPWAN protocols[8][15]

In Short Range, from centimeters to hundreds of meters.

- Zigbee
- Thread
- 6LoWPAN
- Bluetooth / Bluetooth Low Energy
- Wi-Fi

In Long range, from hundred to tens of kilometers.

- LoRaWAN
- SigFox
- Cellular Networks (2G, 3G & 4G)
- NB - IoT

The expected quantity of devices connected can be observed in the Figure 5 where the short range has a substantial growth in the market.

Figure 5. Growth of the connected devices [9]

The mobiles phones industry is expected to slightly grow in comparison to the short-range communication as shown in the Figure 5; this trend is due to rising data consumption in Mbps (Megabits per second) that is affected negatively when the distance between devices increase.
Figure 6 shows the technologies grouped by standards, and interfaces that each network could managed giving to the designer a global view of the structures available in the market.

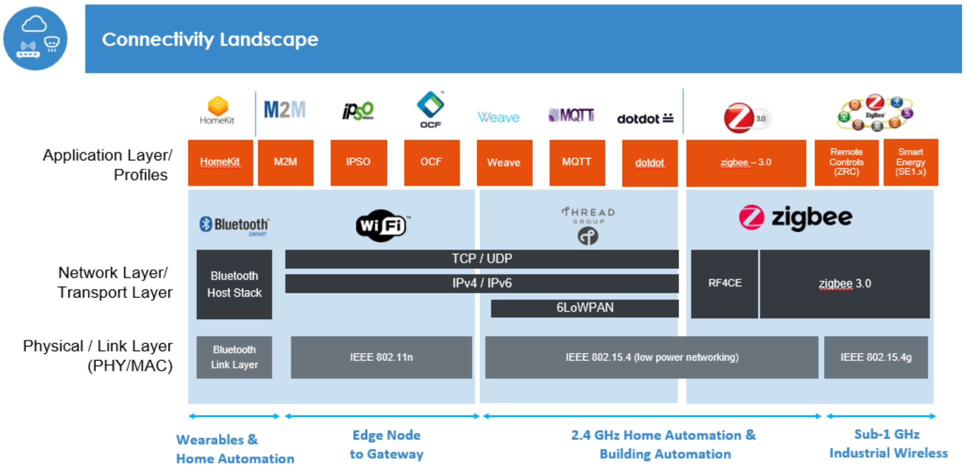


Figure 6. Connectivity technologies and stack protocols[57]

The connectivity landscape explored the protocols available recently on the market and the technology required to be implemented.
Most of the technologies which operate with the standard 802.15.4 work over the 2.4 GHz license exempt, otherwise the governmental entity authorization is required. In the case of the LoRaWAN and Sigfox license exempt that work in the ISM (Industrial, Scientific and Medical) band, the cellular networks and Narrow-Banda IoT are protocols that required a license in order to use them in the market for long-range technologies. However, Sigfox requires to set a key provided by the company and that must be paid.

1.2.1. Short range network

This section will focus on four main networks segments defined in the Figure 6, remarking that the Thread division is a similar network platform as Contiki OS used in this master thesis.

1.2.1.1 Bluetooth / Bluetooth Low Energy (BLE)

The physical layer of this wireless personal area network is property of Bluetooth holder by Bluetooth SIG (Special Interest Group).[10]
This technology evolved from the standard Bluetooth to Bluetooth low energy growing from 2004 to 2010, with the first versions 2 and 2.1 through 3 until version 4 and 4.1 in 2013, beginning to be deprecated. The new step for this technology are the versions 4.2 and 5. This last evolution of Bluetooth has been release to the market in 2017 and provide bigger advantage in comparison to their antecessors.[11][12]
For the Internet of Things, the Bluetooth version 5 released the new features as low power consumption, mesh communications stack and distance range.
The topologies shown in Figure 7 are usually used in these kinds of networks, where all the devices must communicate with a coordinator. Other topologies available are: point to point, broadcast and mesh [13].

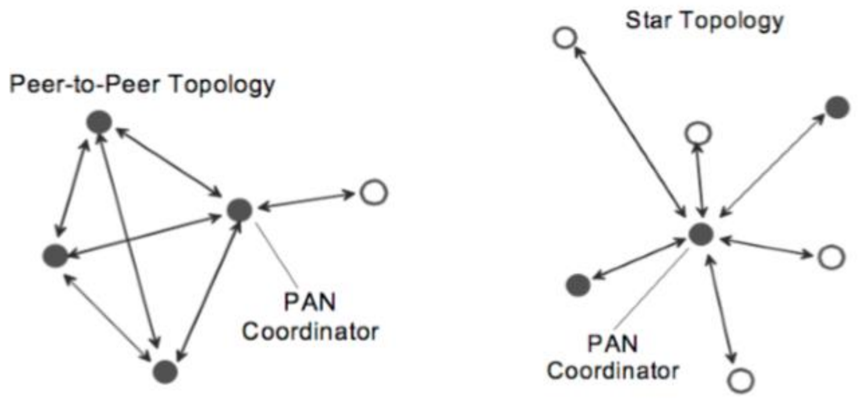


Figure 7. Common Bluetooth network structure[15]

The initiator and terminator of the communication is the coordinator. The PAN (Personal Area Network) coordinator consumes a lot of energy with respect to his constant data exchange among the nodes and the master node.

The Bluetooth Low Energy version 5.0 has improved the network control layer, the consumption power between transmission and idle periods considering its first implementation in Bluetooth 4.0, among other features. Improving the battery's life (autonomy) 10 to 20 times over the classic Bluetooth. In terms of privacy and protection, Bluetooth older versions were sending their address publicly over the spectrum, increasing the risk to be tracked. On the other hand, the recent version enhance the private communication of addresses using ECDH to Exchange encryption keys avoiding that the addresses synchronization .
The Bluetooth Low energy has many features, but can only be selected once at a time. Some significant features are: the enhanced power consumption, mesh stack, or increase the data rate influencing directly the coverage range. In details, it is possible to remark the next information[14]:

- Higher bit rate up to 2Mbps.
- Long range mode with more sensitivity for two new lower bit rates, 500kbps and 125kbps.
- Improving by eight times the broadcast capability with advertising extensions.
- Improved channel selection algorithm that enables to enhance the channel coordination and coexistence efficiency with other Bluetooth and non-Bluetooth traffic. This means that the algorithm CSA #2 allows the device to transmit the data without interfering with another types of communication or oldest version of Bluetooth.

1.2.1.2 Wi-Fi

The Wi-Fi networks are the most known regarding their daily use coming from the frequently connection at home to connect personal devices to internet. The main reason for this, is the large data rate managed by Wi-Fi generation in their wireless local area network (WLAN). The standard used in these devices is the IEEE 802.11 including the physical layer. The Wi-Fi alliance is the group in charge of the evolution and updates in the generational network of this technology.
The technology evolution in the Wi-Fi networks showed a higher increase of the data rate between users connected to the access point. Applications as video streaming, gaming and massive file transfer through wireless means require a faster and reliable data transmission.

The simpler network is an Ad hoc connection, allowing the direct communication between mobile stations (MS). In this case, the Ad hoc acts as an Access Point (AP) that will detect all the MS in his coverage range. Regarding these connections, a tree topology is considered if the mobile stations are in range of others MS, but not from the AP. In addition, the star topology shown in the right side of Figure 7, where all the devices are in range of the access point and the router manage the package deliver, avoiding possible collisions.

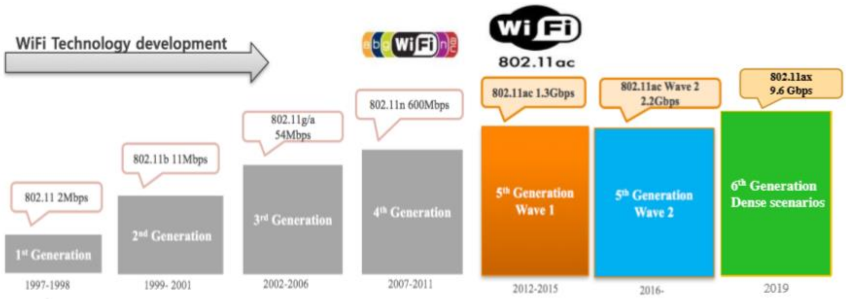


Figure 8. Wi-Fi technological evolution[15]

The main inconvenient of this type of network are the power consumption, attribute to the radio, listening constantly the channel and keeping the link awake. In the comparison of the technology evolution from the Figure 8 and the power consumption in the Figure 9, the WIFI is a network designed for local areas that require speed bit rate.

For applications where the energy is a precarious source and require batteries with low maintenance, the Wi-Fi is not recommendable for low power portable devices.

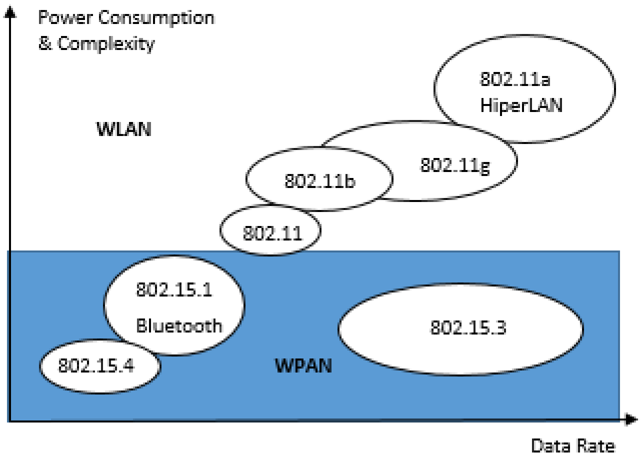


Figure 9. Power consumption & complexity relationship vs data rate in LPWAN[15]

In contrast with the Figure 8, the Figure 9 exploits the complexity and power consumption. In the case of Wi-Fi networks, each new generation increase their power consumption regarding the high bit rate managed.

The rest of the networks are considered personal areas networks in relation to their flexibility and the possibility to be implemented for wearable applications. On the other hand, the new introduction of the 802.11ax standard Wi-Fi (WiFi6), could be applicable for wearable devices with large batteries considering their lifetime extension. 802.11ax is expected to work with the next generation of mobile network 5G, and has been designed for the increasing user demand,improving their peak rates, the throughput and network efficiency[16]. For more information about the new features see Annex 16.

The 802.15.3 standard is the wireless personal area network (WPAN) known as Ultra-wideband (UWB). This standard will not be considered in personal applications and the market impact is not highlighted, or only for specific case. The UWB is frequency related for VANET (Vehicular Arena Network) systems with a data rate going from 11 to 55 Mpbs.

1.2.1.3 Thread

Thread is a networking protocol developed for the implementation of IoT. The main manufactures in charge of this project are NXP, Qualcomm, and Analog Devices among others.
The standard used in this network protocol is based on ipv6 (this topic will be explained in the chapter 2), with low latency and runs over 802.15.4.
The protocol has been released and supported by the Thread Group[17], and the devices can be certified with the stack modules after passing several tests. The open source OS (operate system) is known as OpenThread[18][17]. In comparison with ContikiOS[19] the interface has similar aspects and points aiming at the same purpose: “assimilate an standard for low-power wireless network ”.

This master thesis has **not** the objective to explain the difference between both network protocols.
Standards employed in this protocol are observed in the Figure 10.

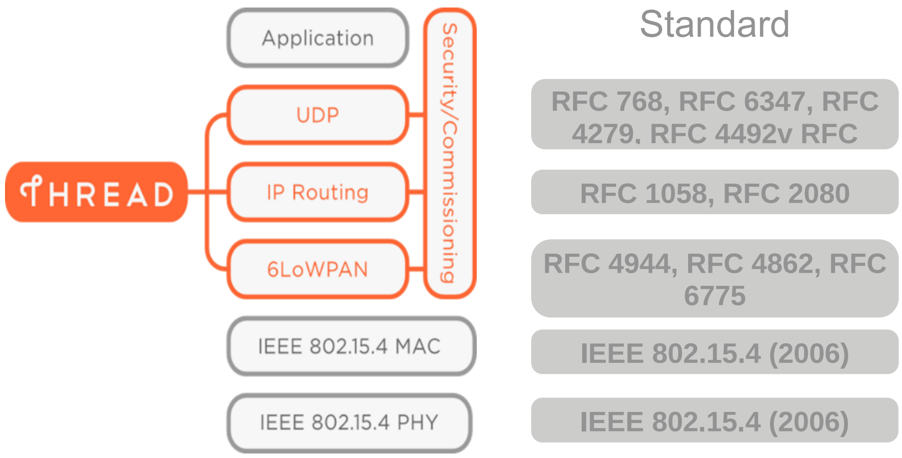


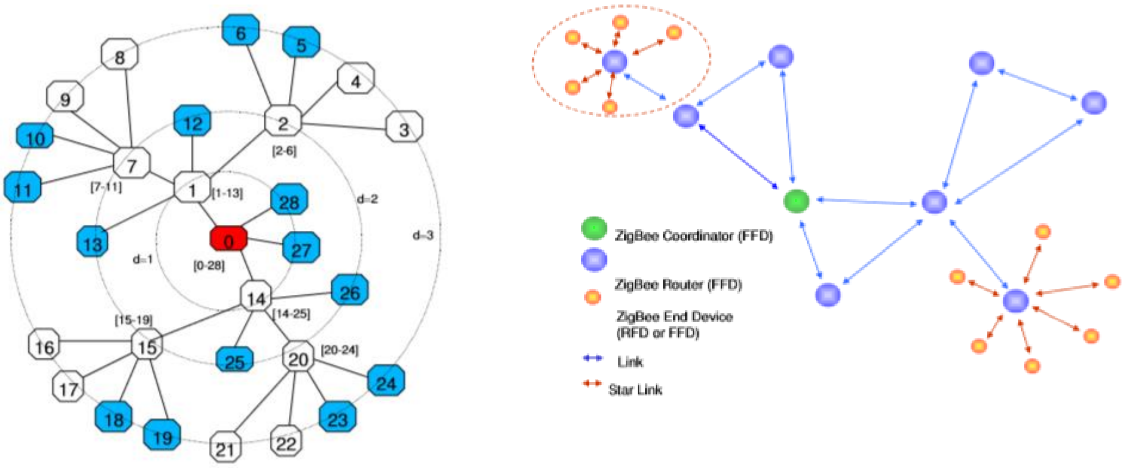
Figure 10. Stack Module of Thread[20]

Contiki manages the same stack structure and standards as OpenThread. Both of them are the OS for network protocols that works over IPv6, the main difference being the security/commissioning with their own firmware[17].
The chapter 3 will deploy the stack module and explain in more details the configuration. In this section, Contiki has been selected for the evaluation and application regarding its popularity on the market and documentation. The case of the Thread is used mainly on applications notes from devices certified. The manufacturers are registered on this alliance. The popularity extended of Contiki provided a higher possibility to find information and support available online instead of register and wait technical support on the manufacturers’ website.

The devices supported by Contiki go from the simple microcontrollers to ARM based structures of 32 bits. In addition, Thread is good supported by the manufacturers and requires a specific procedure to get more detail information from their technical support.
The implementation of network protocols is due to the necessity to improve the performance in the Physical/Link layer IEEE 802.15.4 for Internet of things.

1.2.1.4 Zigbee

The high-level protocol deployed by Zigbee alliance provides a complete solution for IoT applications regarding their networking for the smart devices that implement the 802.15.4 as physical layer. Worldwide, it is recognized as a protocol designed for IoT, making simpler the interconnection of smart objectives, combining multiple applications profiles into a collection of devices[21]. For example, dedicate devices for Lighting and Occupancy.



a) b)

Figure 11.a. Zigbee tree topology
Figure 11.b. Zigbee mesh topology[15][21]

Zigbee has the possibility to configure the topology, even though all the models must contain the coordinator (FFD) shown in figure 11.b. The tree configuration observed in figure 11.a. goes from the coordinator node (red node) to the external nodes in blue color that requires the white node hubs to achieve the coordinator.

The concepts of profiles and cluster are defined by attributes that compound the application for the end user. The attributes can be supplicated and assigned for different clusters and giving more than one end point application. For more details of the profiles' structure predesign by Zigbee alliance for end applications refer to the Annex 1.

1.2.1.5 6LoWPAN

The IPv6 over Low Power Wireless Personal Area Network is a standard network that allow the implementation of internet protocols in embedded devices based on batteries, connected by low-power and low-bandwidth[22]. Simplifying the IPv6 functionalities compacting the data-frame is showed in the Figure 12.

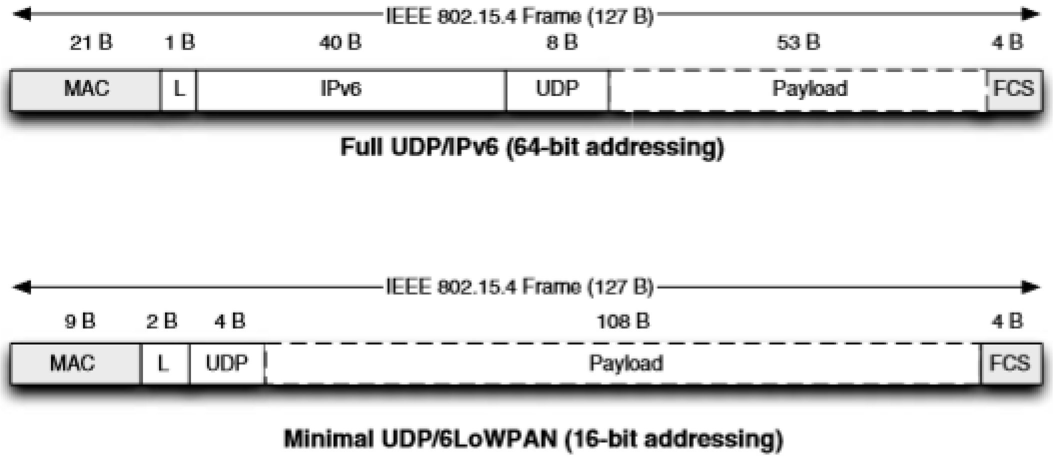


Figure 12. 6LoWPAN header compression exemple [22]

This standard will be explained more in details in the section 1.4.1 which will focus on the applicability in the field of study.

1.2.2. Long range network

The communication for long rate is deployed to reach distance from hundreds of meters to tens of kilometers.

1.2.2.1 LoRaWAN

The Long Range (LoRa) wide area network is the stack protocol for the LoRa modulation defined by the LoRa alliance. The frequency band is over the license-exempt spectrum using the ISM band. One of its advantages is the range on the suburban areas up to fifteen kilometers and urban areas up to five kilometers. The physical layer for this network protocol can use the 802.15.4g [23] that is an improved layer of the 802.15.4.

The architecture of the network protocol is a centralized system (see Figure 13) where the node can interact with the master node or LoRa server though the

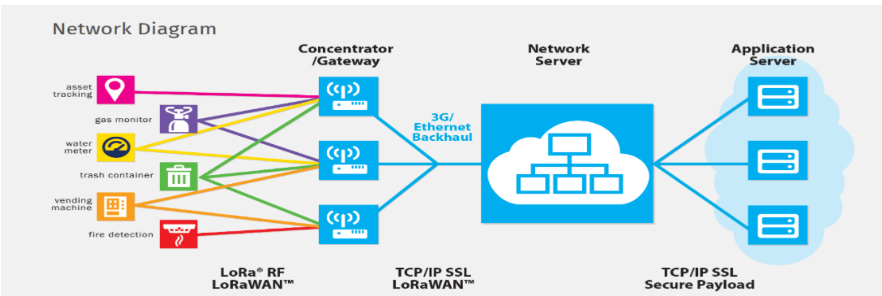


Figure 13. Network Diagram for LoRaWAN.[23]

gateways that are linked with the end nodes, existing the possibility that one device is shared with more than one gateway. To avoid collisions the MAC address is shared, and the packet is delivered to the high gain gateway when the channel is available.
Depending of the distance, the spread factor increases and the data rate is reduced substantially. The spread factor is associated with the fading on the channel given long distances and obstacles.

1.2.2.2 Sigfox

The operation range of this network protocol achieves longitude from thirty to fifty kilometers in rural areas, while in urban areas reaches distances from three to ten kilometers. This protocol was deployed by a French company and nowadays is declining their market impact due to the innovation in the narrow band protocols and their close business model license impact. In addition, the Sigfox application has been used for certain type of projects that required long distance with a previous RF infrastructure, the main advantage being the ability to connect to the public gateways in contrast to the LoRaWAN private networks.
The low power consumption and the long-distance communication maintain this network protocol striking, but cannot support voice application. Sigfox have their own centralized system developed by Sigfox alliance (see Figure 14). The bit rate managed in this network protocol is one hundred bits per second with a maximum packet size of twelve bytes, and depending on the transmission range up to one kilo bit per second.

Figure 14. Sigfox network architecture

1.2.2.3 Cellular Network

This network protocol has become the most popular paid service for long distance communications. The cellular technology received their name from a quasi-hexagonal coverage area; the evolution of this technology has increased in the last ten years for the high demand of connectivity. The number of subscriptions and data rate consumption (Figure 15), increased due to the streaming of video and data exchange between mobile devices.

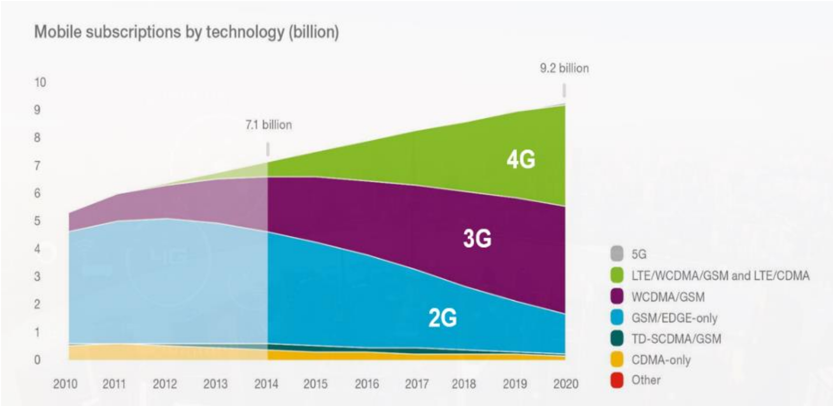


Figure 15. Number of subscribers[24]

Each generation evolution trades off the data rate for coverage, the cellular area is reduced giving the possibility to more users to use efficiently the bandwidth provided by the Internet Service Provider (ISP).

The Internet of Things in this type of networks in related with the LTE-M1 using the existing cellular structures, the long-term evolution category M was designed for IoT applications and Machine to Machine (M2M) communication, also related with the fifth mobile generation (5G). Even though, both technologies have similar aspect they cannot be consider the same.

1.2.2.4 Narrow Band IoT

This LPWAN wireless access is a standard defined in the 3rd generation partnership project (3GPP) used worldwide in the telecommunication associations. The main advantage is the low power consumption and the possibility of an extended battery life up to ten years[24].
In the figure 16, the advantages of this technology provide to the customer the possibility to employ the alternatives for long range communication due to a gain of 20dB over the conventional GSM network.

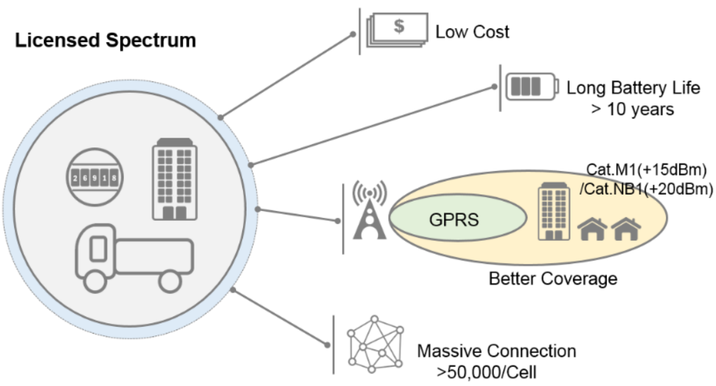


Figure 16. Narrow band (NB) IoT advantages.

The average data rate is expected to be 100kbps, for a quicker throughput. The new technology is a license networks since this communication requires the use of telecommunication network.

The poor documentation and operational uncertainty between the ISP and the modules that use this technology, give to the customer an unsecure vision of their future potential.
Nowadays Quectel is one of the manufacturers at the vanguard of this technology evolving from the NB first generation to NB2.
The main advantage of the new narrow band evolution is the possibility to handle the handover between cellular networks.
In the table 1, it is possible to observe the different features remarking a new type of localization through NPRS using the Observed Time Difference Arrival (OTDOA) measurements [25].
Therefore, providing to the customer the possibility to use mobile stations, instead of static nodes located without any change of cell.

Parameters	**Cat . NB1**	**Cat . NB2**
3GPP release	Release 13	Release 14
TX power class	20, 23 dB	14, 20, 23 dB
Maximum DL TBS	680 bits	2536 bits
Numb processes HARQ	1	2
Support for positioning of device	No	OTDOA
Multicast	No	Yes
Mobility	No	Yes cell re-selection

Table 1. Comparison between NB-IoT and NB2

1.3. Wireless Sensor Networks

The concepts of Internet of things (IoT) and Wireless Sensor Networks (WSN) have been mixed and misunderstood.
Both systems share the objective to interconnect several devices into a single point with the aim of providing certain information. IoT can be assumed as a WSN, but not all WSN can be consider as IoT. The main point of this affirmation is given by the concept of Internet of thing: *“The Internet-of-Things (IoT) main vision is to create an intelligent world where the physical, the digital and the virtual are converging to create smart environment”.*[20]
WSN is a self-configuring network of sensor nodes, using radiofrequency signals deployed to obtain data from the environment with a limited processing capability, and in order to conserve power that is a limited resource. In few words, IoT is the main ecosystem of sensors, gateways, internet connections, front haul, backhaul and final user. The WSN is the local network which provides certain “dummy” information to be processed and analyzed.
The wireless sensor node requires a border router device connected to internet which interconnects the local network.
In the section 1.2 each network protocol works over different type of standards instead of IPv4. The figure 17 shows the OSI model and the similarities regarding different physical layers, considering the IEEE 802.15.4 as the main standard network.

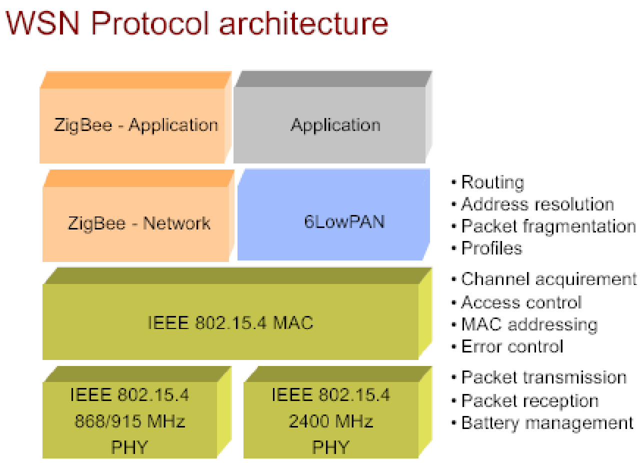


Figure 17. Wireless sensor network protocol[15]

The five principal components of this type of networks are:

1. Processor/Controller
2. Supply source
3. Internal/External Memory
4. Radio link
5. Sensors

Nowadays the micro-processor comes with a micro-controller due to the split of communication into the computational process. With a limited supply resource, the devices must run in low power consumption.
On the physical memory the software commands is allocated to control the sensor information and radio link communication. The micro controllers have dedicated process structure to manage the communication performance.
The physical layer of the IEEE 802.15.4 sub-GHz (868-915 MHz) applies for private network with the possibility of increasing the coverage area in the case of the Zigbee protocol. The smaller the frequency oscillation, the longer waveform pattern and signal penetration on the coverage area.

1.4. Communication protocols

This section will expose the protocols and standards employed in the Internet of Things (IoT), as well as the stack protocols shown in the figure 6.
Other standards and protocols will be showed and followed as in the text book. [22]

1.4.1. 6LoWPAN

This network protocol is based on IPv6 over low power, and low rate wireless personal area network (6LoWPAN), standardized by the IETF for application where the header packet is reduced and used in Machine to machine (M2M) communication. This protocol usually has for target application:

- Personal devices (health & fitness).
- Smart metering and grid infrastructures.

- Vehicular automation.
- Home automation.
- Among others.

The physical and link layer works with the IEEE802.15.4 standard, and this protocol is also known as optimize IPv6 over IEEE802.15.4. The energy optimization requires a UDP packet transmission regarding their features to avoid the acknowledgment from the receptor. This characteristic allows only the unicast transmission saving energy, instead of waiting a reply from the receptor node and assuring a good reception. The main problem of this technique is the uncertainty of the packet arrival. The Figure 18 shows the OSI model used in traditional IP protocol stack and the 6LoWPAN.

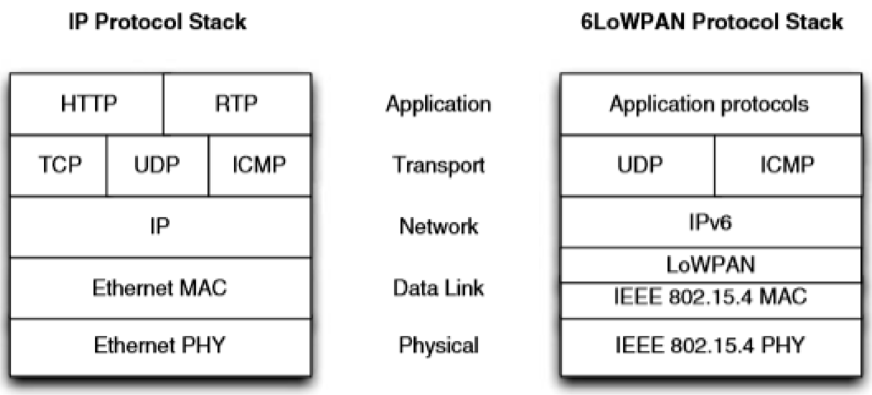


Figure 18. Difference between conventional IP and 6LoWPAN protocol stack.[22]

1.4.2. IPv6

The Internet protocol version 6 was deployed due to the increasing number of IP addresses generated daily. The IPv4 protocol follows the standard RFC791 while the IPv6 works over RFC8200.
The number of addresses in IPv6 is $7.9 \cdot 10^{28}$ times higher than IPv4, and the auto configuration protocol is related with the Annex 2.
Other important differences are given regarding the facility of the IPv6 to integrate some features as ARP, DHCP, NAT & super-netting.
The main changes are:

- Traffic class: substitute the type of service.
- Flow label: faster forwarding can be assumed as a virtual circuit, which require the same treatment to the same destination also used with resource reservation.
- Payload length: in this case the amount of bits are higher than in IPv4 and we consider the packets as jumbo gram in comparison to the datagram total size from the protocol version 4 as small packets. The jumbo grams must set this parameter to zero and use the extension header allocated after the destination IP address.
- Next header: This block is in charge of providing information about the posterior's headers incoming. The table of this parameter allows to understand the kind of packet that the router or host must compute.

- Hop limit: as in the same case for Time to Live in IPv4 this parameter shows the maximum hops that the packet can do in the net, if this value is equal to zero the router will drop the IP datagram. Each time that this datagram passes a router this value is decremented by one.

The packet restructure is shown in the figure 19 a (IPv4 header format), and figure 19 b (IPv6 header format).

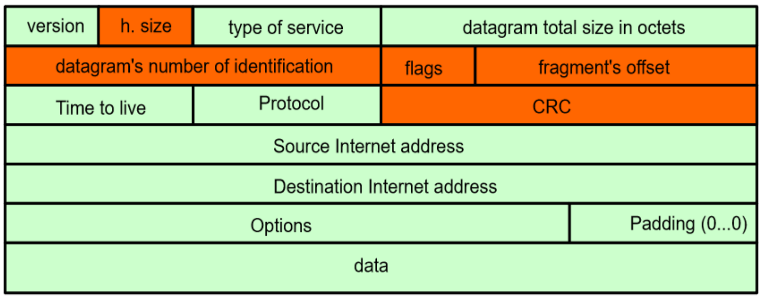


Figure 19 a. IPv4 header, showing the aim difference slots with IPv6

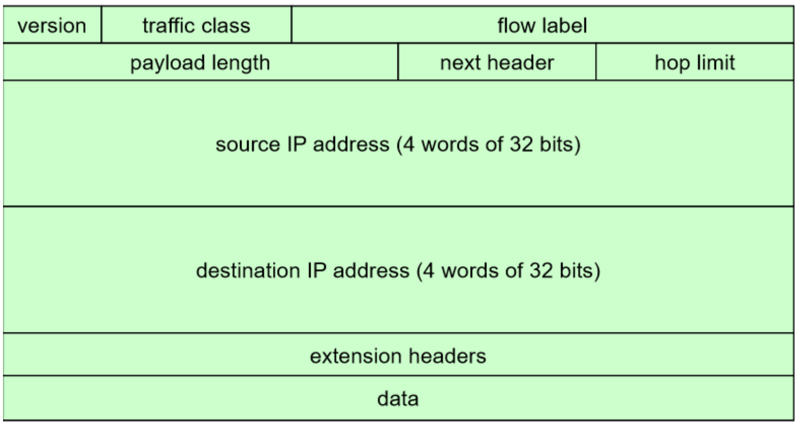


Figure 19 b. IPv6 header

1.4.3. UDP/TCP

The figure 18 at the left side shows the transport layer segment. In the IP protocol stack it exists three type of protocols used to pack and send the packet though the network. User Datagram Protocol (UDP) transmits the packet without any link establish and enough information in the header to try to achieve their destination, while the Transmission Control Protocol (TCP) is a robust transmission with a link establish allowing a really low packet loss in comparison to UDP.

1.4.4. RPL

The Routing Protocol for Low power and lossy networks (RPL) is a protocol based on IPv6 that requires a router to inform its neighbors of topology changes. It is less computation complex than IP protocol because of the Distance Vector (DV) protocol used in his configuration. DV protocol is based on calculating the next hop address with the less cost to any link of the network[26].
The standard RF4944 on the figure 20 shows the possibility to locate the routing protocol into the transport or network layers depending until where the packet can

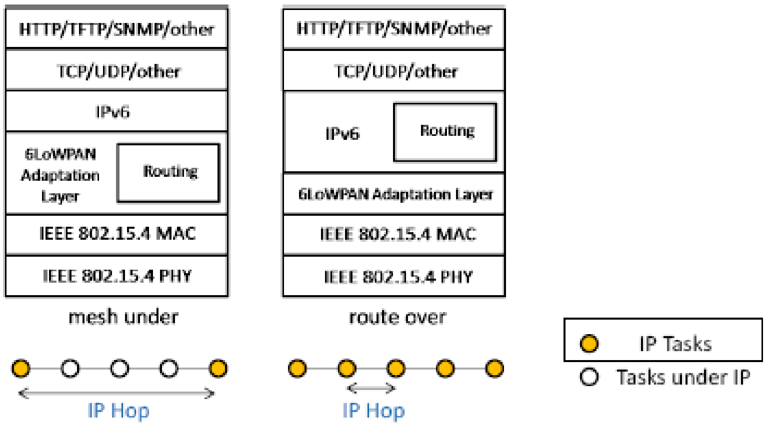


Figure 19. Mesh-under versus Router-over[15] ***(source: rfc 4944)***

access. More graphical explanation for the unzip datagram in the routing hop is available in the Annex 2.

1.4.5. MQTT

The Message Queuing Telemetry Transport (MQTT) is a protocol for the interconnectivity M2M (machine to machine) on the application layer. This protocol is favorable for the IoT regarding the main features (see Annex 15) as it was explained by IBM in[27], this protocol is light with an easy implementation on restricted hardware where the bandwidth is limited with high latency.

The MQTT overall works over TCP/IP, but for the sensor network development, it is necessary to use a branch of this protocol which works over UDP. The MQTT-S [28] has been redefined as MQTT-SN [29]due to many confusion of the abbreviation S. For this reason MQTT.org suggests the “SN” one regarding the application of the Sensor Networks working with MQTT.
The development and protocol construction requires the broker and the sensor publication/subscription to transmit and receive data. The advantage to assign a QoS (Quality of Service) will be discharge (QoS =0) for this project regarding the low power consumption between the broker and the node client. The publication and subscription model using the MQTT integration follows the next structure (see Figure 21):

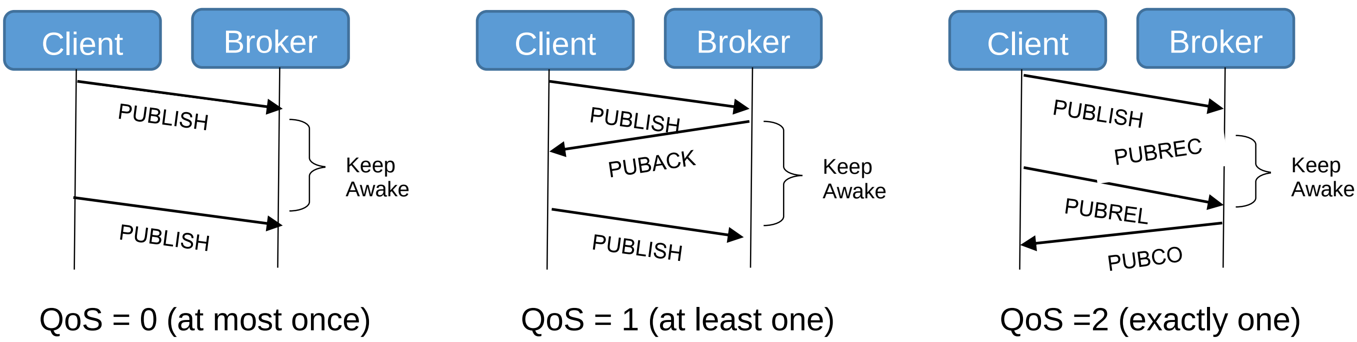


Figure 20. Quality of Service MQTT interactions

1.5. Related projects

The agricultural industry has developed projects related to IoT. The implementation on wireless sensor networks has been working among years.
Therefore, the monitoring of the crops is the first step to implement a complete system, where sensors and actuators control the environmental greenhouse or land.
For example, in [30] the moister and temperature sensor levels activate the irrigation system. The supervision rules were created by the farmer aligned to a threshold in the monitoring in order to automatize the aspersion trigger. This network was developed using Zigbee and GPRS (second cellular generation) embedded in the gateway with Xbee.
The [11] contains main features of a wireless sensor network implementation on the field. The variables measured and the location of the nodes using GPS, give more accurate information determined by the designer. The Human Machine Interface provides useful information and the project was developed with more than one sensor node. The network implemented was a Bluetooth in a star topology.

Meanwhile, the implementation of a simple wireless sensor network with similar implications, as created in this thesis, is considered in the application in [31]. The system in [30]shows the water flow measurements using a website in real-time . The network was constructed in ZigBee, in a region where the obstacles weren't controlled and the water source located in strategic point. In addition, this system was not optimized for battery sources regarding that the packets that were transmitted one per second.

CHAPTER 2. TECHNOLOGY SELECTION

The proper technology selection is done when the application is known and parameters of the project are well selected. Variables and possible issues can impact the final performance if they are not taken into account. The technology evolution (Chapter 1, section 1.1) gives a starting point from which elements are available and sustainable depending on three main features: Price, longevity and documentation available.

- Price: the bill of materials (BOM), considers the components that goes in the final design print circuit board (PCB) and price by component. In many cases it is considered cheaper than buying an end solution. The possible risk with the first option is the complexity in the design and building of those devices. Depending on the final cost of the solution, the proposal is accepted or rejected.
- Longevity: the devices have an end of life when the manufacturers are not anymore able to provide an old component due to the constant upgrade of the materials. The technology has an "expiration" time when the devices are not popular enough into the market, forcing the manufacturer to stop the mass production. It is important to be in contact with a technological distributor or the fabricant website news to update and upgrade the network systems.
- Documentation: the information available online gives some solution to emergent issues in the developments of the project. In case of special devices, the provider can supply more technical information or application notes related to the customer necessities.

2.1 Preliminary agricultural information

The agricultural industry is a sector highly exploited that require attention and management. Communication and information acquisition are fundamental for a continuous analysis, planning contention and improvement of technical mechanism depending on the crop.

Parameters as water, soil and solar light are the tree pillars of the seed germination and development of the plant. The monitoring of the crop requires sensors for environmental condition as humidity and temperature. Those variables are necessary to understand the evapotranspiration[32]–[34] and vapor saturation of the plants. Analyzing individually the sensors data gives information details as quantity of water on the environment monitored.

The level of water consumed by the crops must be higher than the transpiration of the plant. The quality of the product and the harvesting process impact the level of humidity on the greenhouse. The evaporation losses due to solar irradiation and temperature must be compensated with a good irrigation system and the precipitation per sector in the region.

The water retention is affected by the quality of the ground and the texture of the soil. The irrigation system could affect the soil erosion on specific areas when the quantity of water is overpassing the retention limits and excessive furrow or border inflows. The prevention plan can control the soil erosion when a careful study in situ is taken. The study can be done with the information acquired by the sensors.

It exists different equipments and formulas to measure the evapotranspiration of a crop. The atmometer/evaporimeter use the porous ceramic disk maintaining the water levels, but requires a removal freeze protections. Evaporation pans, use open top metal water container, measuring the difference of levels in the water evaporated for the solar energy, and obtaining a coefficient regarding the evaporation rate and crop growth state[33].
The crop potential production is eventually impacted for the historical natural pattern of the location, type, reliability, and climate data.
The aperture as pores called stomata located in the upper and lower surface of the leaves, allows the dioxide of carbon (CO_2) absorption required for the photosynthesis and plant growth. Regarding the size of the crop, the irrigation system must satisfy the difference between evapotranspiration and soil-moisture preventing the water-stress.

2.2 Selection features and devices

The communication protocol explained in the *section 1.4* will again be aboded in this section, focusing on the agricultural side and giving more details for its application. The selection of the possible alternatives between network protocols, microcontroller (in fact, compressed computer board released on the market and facilitate on the laboratory). The four main points in the design that will be explained are: Microcomputer, Network protocol, Transceiver module and sensor.
The greenhouse and long crop extensions have critical aspects indicating for higher to lower impact on the price.

- Low power consumption.
- Low price in the market.
- Robustness.
- Reliability.
- Packet latency.

2.2.1 Single Board Computer

The microprocessor is the computational core of the system and the human machine interface, used for the human interaction between the local network and Internet. This board must be configured as the border router regarding the internet protocols used in each net. The Local or Private Network use the sensor nodes (Transceivers modules) and works over IPv6 while the Internet connected worldwide on IPv4 have different framework as explained in the *section 1.4*.
The microprocessor selected for this project is a cortex-A53 embedded into the Raspberry pi 3 B (see figure 22). This single board computer is highly documented and has a moderate power consumption. This element will be considered as fully supplied regarding their continuous communication between the networks.

Figure 21. Single board computer Raspberry pi 3 B, image

source :https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

The microcomputer selection concerns the equipment availability and the low price in the market. In addition, the peripherals elements as the Ethernet, USB and GPIO are necessary for the project. Other terminals as HDMI port and display can be useful for the HMI (Human Machine Interface) in case to be required, and the low importance terminals for this thesis are the camera and audio ports.

2.2.2 Network protocol

The application conditions and customer needs require the use of low power consumption systems. The technologies proposed in the *section 1.2.1* are usually selected for their ease of deployment and robust network. Most of the protocols exempt Wi-Fi and Bluetooth, work over IEEE802.15.4 which means that they fulfil the criteria for physical and link layer required as part of the objective in this thesis.

The Bluetooth is not part of the objectives of this thesis due to the fact that the network protocol must run over IPv6, and regarding the RFC7668 that consider the 6LoWPAN over BLE.
The communication protocol selected to work on this net, is the 6LoWPAN for the low power consumption, and availability of IP addresses, packet transport and documentation.

Following the stack modules in the figure 20 for multi-hop devices, one of the features of the network deployed is the tree architecture (figure 11.a). Regarding the exterior nodes, it will not interact between them, or only if they require to use another hop to arrive to the coordinator. However, this option will not be put it on practice due to time limitations.

The UDP stack module will be used as transport layer due to the facility of sending information and idle node after certain time. The time slots for the power consumption are established by the 6LoWPAN MAC but, regarding the use of ContikiOS, the MAC protocol used in this operating system is a X-MAC[35] modified, also known as ContikiMAC[36]. The package of the data and transmission of the information using the ContikiMAC protocol can be observed in the Figure 23.a. The reception windows are enable in a timestamp where the sender must transmit the information, otherwise a simpler MAC protocol as the X-MAC in the Figure 23.b shows the timeline of their performance.

The main difference is the ACK location and size of the windows where it is enabled. The radio in the Contiki receptor enables the channel of the node for a slack of period, until detecting the package and begins to listen. Meanwhile the X-MAC is the sender which needs to send short preambles, and consequently the receiver wakes up.

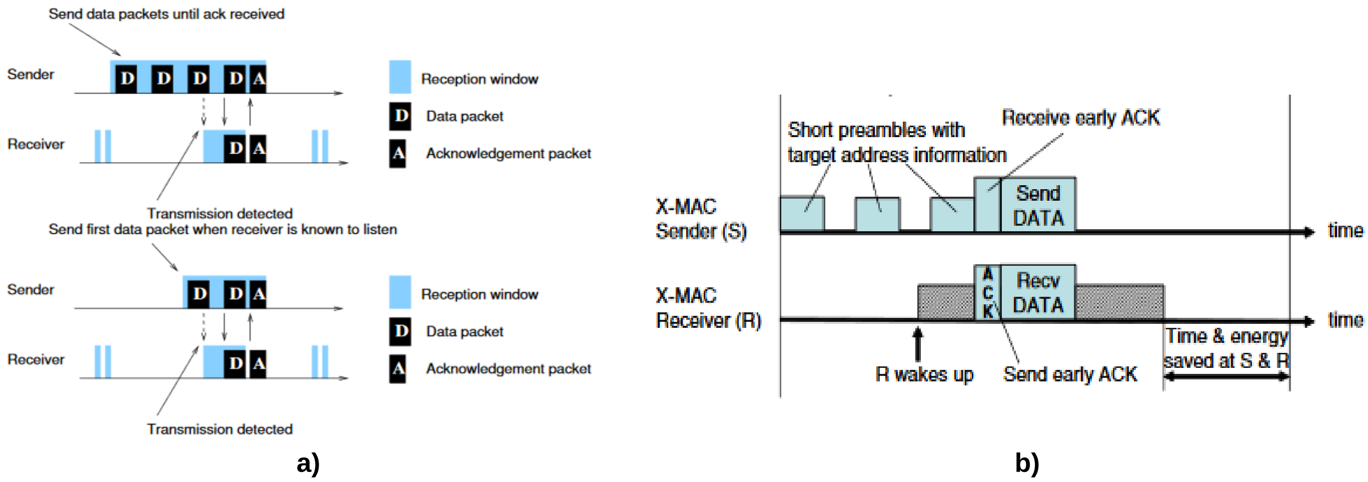


Figure 23.a. ContikiMAC transmission phase-lock[36]
Figure 23.b. X-MAC timelines[35]

Once the data frame reaches the boarder router, the Raspberry must run the server listener from the UDP packages received from the port 5678. Using Python as UDP server working with MQTT, the frames now have JSON structure and point to a static IPv4 where the physical server is located. This section will be explained in more details in the results *section 3.3.*

2.2.3 Transceiver modules

The modules available on the market are designed for general proposed applications and used in specific range of actions. Most of the greenhouse have the advantage of providing a direct line of communication between nodes. Meanwhile, the plants in an adult state are grown enough creating an obstacle for a direct communication. The performance is reduced concerning the shadowing which impact negatively the coverage

. More bridge nodes must be awake in longer periods in order to allow for the external motes to connect with the coordinator.

The transceiver must work over Low Power Wireless Area Network (LPWAN) and for this application in concrete the network must conserve energy, avoiding a possible mesh architecture.

Figure 24. RE-Mote from Zolertia.[37]

The RE-Mote nodes from Zolertia (Figure 24) have been selected for their employability within low power networks, and previous research[3][19][38] with those devices. This node has a cc2538 microcontroller cortex M3 dedicated to the communication coordination for Zigbee and 6LoWPAN protocols. This device works over IEEE802.15.4 and operates at 2.4GHz.

2.2.4 Sensor

The selection of the sensors depends on the requirements of the customer, in this case the Agropolis from UPC. The limitations in this first selection is the facility to acquire the devices, regarding the resources used in this project, for this reason the DHT22 also known as AM2302 sensor[39]. This solution contains both transducer for temperature and humidity variables. Usually projects for weather measurements contain sensors as temperature, irradiation and humidity.
One of the requirements is to create a dataset with the information collected in order to show tables of temperature and humidity.
Possible applications of the information from the database could give hints to improve the crop using alarms to set an autonomous irrigation system enhancing the future crops production in real-time. This end application will be not implemented in this project.
External sensors can be plugged to the GPIO port using the I^2C interface and programming the output in decimal notation, regarding the type of data obtained in the sensor and using the datasheet in order to get the formula translation.
In this case the temperature sensor DHT22 will be plugged to the node in relation to the low sensitivity and high accuracy in comparison to the DHT11 for changes in the environment.

The AM2302 sensor has a digital output, and the voltage supply recommended for the module is from 3.3 to 5.5 V. In order to maintain the sensor in the operation mode and the power consumption reduced, the sensor will be installed in the pins 13, 12 & 11.
The ADC port mapping must supply the sensor's requirements mentioned previously. The configuration mapping of the pin out the ADC channels from the sensor node is shown in the Figure 25:

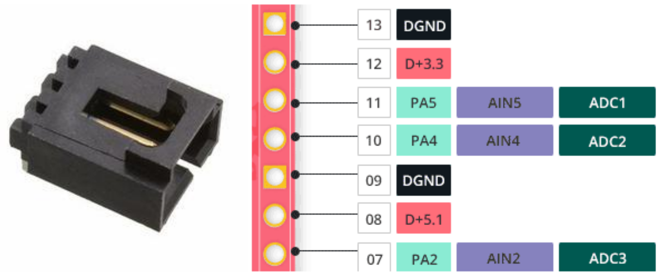


Figure 25 RE-Mote terminal pin out mapping.

source: https://github.com/Zolertia/Resources/wiki/RE-Mote

The analog-digital converters ADC0, ADC3 have a voltage output of 5V1 and 5V2 respectively, while ADC1 andADC7 work with an output voltage of 3V1 and 3V2 in the MSP430 microcontroller[20].
The ADC available in the RE-MOTE are ADC1,2 and 3, but the ADC2 is not enable physically on the socket[40]. Regarding this event, ADC1 fits with the minimum requirement of the voltage supply in the DHT22.
The signal conditioning requires a pull up resistor as shown in the Figure 26. This pull up is recommended by the supplier to ensure the state of the signal. The value of the resistance will be selected regarding the cable’s length from the MCU to the module.

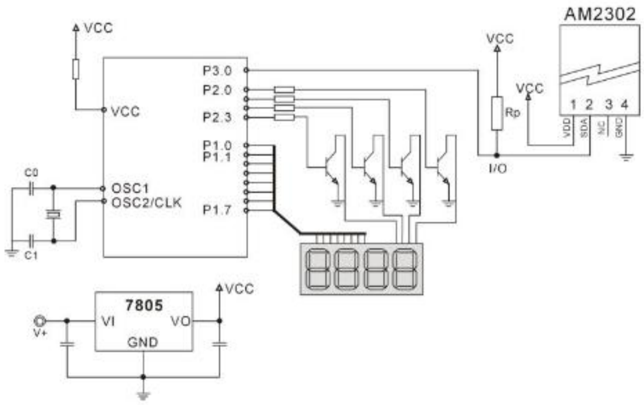


Figure 26. Typical schematic for single bus [39]

In the section 7 from the user manual of this module, it is recommended to work with 3.3V. It is necessary that the length of the cable is not greater than 100cm, in order to avoid a possible voltage drop and measurement error. Thus, the recommendation for the module installation is to connect it with a cable measuring less than 20 cm and 10Kohm as pull-up resistor.
The output of this modules depends on the synchronization between the MPS430 and the DHT22, otherwise the output will generate an error. The data structure consists in 40 bits split in 5 bytes. The first 16 bits are for humidity, the other 16 bits are for the temperature and the last 8 bits are for the parity.
In [39], the table 5 shows the format definition and the data, allowing to write the C script for the controller showed in the Annex 3. The main script contains machine time for a successful synchronization.

CHAPTER 3. SYSTEM IMPLEMENTATION

The selection of the elements in the *section 2.2* will be used to develop the network over IPv6 using the network protocol 6LoWPAN.
The border router will be implemented with the raspberry pi 3 B, the main objective of this element on the network is to create a gateway to exchange information from the IPv6 to IPv4. The configuration of the board will be explained in the *section 3.2* where it will be operative, point possible errors and solutions. In addition, the configuration of the server allocated in another raspberry pi B+ will be explained.

3.1 Operative System

The usage of the raspberry requires to flash a SD card, and an external computer to record the operative system using flashing memory programs. In this case the program used is *Etcher*[41], and the procedure can be followed online in[42].
The operative system employed in this master thesis is the Raspbian based on Debian using the version Jessie over the raspberry pi 3 B. The Linux kernel will allocate the CETIC-6lbr[43] used as border router based on ContikiOS. The structure can be observed in the Figure 27. Contiki has modules already prepared for the development of 6LoWPAN and RPL network protocols.

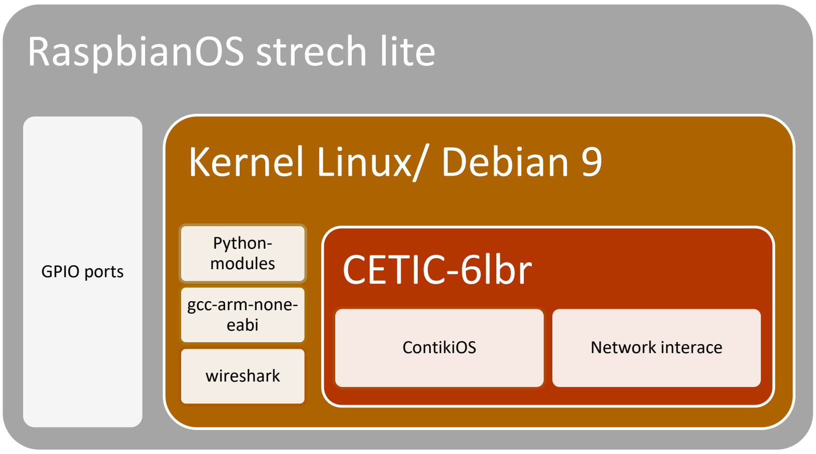


Figure 27. Operative system composition

The RaspbianOS stretch lite is the version without user interface and workbench frameworks. This simplify framework allows to allocate more resources to the border router, and the limited memory from the SD Card is necessary to readjust the space and execute the program from command prompt.
For the first time execution of the Raspbian Jessie, it is necessary to use an USB-Keyboard and a display with port HDMI. The security of the OS in the new versions of raspbian SSH (Secure Shell) is disabled. It is required to use the command *sudo raspi-conf* and enable the ssh interface connections regarding the security protocols established by debian/raspbian. In the Annex 4, it is possible to observe the procedure for the first-run.
Other possible versions of raspbian as whezzy have the ssh protocol enabled on the operate system and can be found in the repertories of raspbian. The main problem of the whezzy version is that some libraries mismatch with the last version of CETIC-6lbr, and require complementary repositories to download the packages.

The software tool employed in the case of IPv4, changes in the raspberry pi after enabling the SSH interface, and is known as the Advance IP scanner[44]. The computer must share the same network, regarding all the mobile stations connected in a common access point.
The SSH is used to create virtual connection between the border router and an external computer with the Putty[45], creating the virtual client for the communication from the port 22.

3.2 Installing and configuration

The CETIC-6lbr is designed over Contiki as it was explained previously. This framework can be catalogued as a middleware, used to manage the data and node configuration of the IPv6 subnet.
At this point, the raspberry must be connected via Ethernet with internet connection to a router.
The packages and repertories may be updated and upgraded before any changes on the machine.

```
sudo apt-get update && sudo apt-get upgrade # Takes long but necessary
# Be careful with the size, using the command df -h
```

The libraries are required for a proper installation of the tap-bridging mode, git, fetching and building.

```
sudo apt-get install bridge-utils git build-essential libncurses5-dev
```

The drivers for the microcontroller msp430 usually are installed on the Linux kernel. In case the distribution do not have this libraries installed, it is possible to download them using the following command-line:

```
sudo apt-get install binutils-msp430 gcc-msp430 msp430mcu mspdebug
```

The GNU used in the operative system requires the arm embedded tool chain installation to compile the executable scripts on C.

```
sudo apt-get install gcc-arm-none-eabi gdb-arm-none-eabi
```

The serial interfaces, objects and hexadecimal format file are necessary to compile the motes into the script as in the examples.

```
sudo apt-get install python-serial
```

The hexadecimal reader and writer are used with the library intelhex, but this module of python must be installed following the next functions.

```
sudo apt-get install python-pip  #in case that is not installed
sudo pip install intelhex python-magic
```

The last version of 6lbr downloaded require the following commands.

```
sudo git clone https://github.com/cetic/6lbr
cd 6lbr
sudo git submodule update --init --recursive
cd examples/6lbr
```

In the folder 6lbr of examples it is necessary to build the applications, tools and plugins.

```
sudo make all
sudo make plugins
sudo make tools
```

Now built the middleware, it is necessary to install the plugins and let the CETIC-6lbr by default.

```
sudo make install
sudo make plugins-install
update-rc.d 6lbr defaults
```

The configuration by default use the local and Ethernet interfaces with the dhcp enabled for the IP address selection. The network interface in raspbian Jessie diverges with the 6lbr thus, the dhcp in /etc/dchpcd.conf must be disabled adding the line:

```
denyinterfaces eth0 tap0
```

In addition, the network must be assigned to a static local IP and it is necessary to create a bridge between the physical Ethernet interface and the private network created by the nodes. Consequently, it is necessary to build a bridge interface on the file /etc/network/interfaces:

```
auto lo
iface lo inet loopback

iface eth0 inet static
    address 0.0.0.0      # write here the static IP for the implementation

auto br0
iface br0 inet dhcp
    bridge_ports eth0
    bridge_stp off
    up echo 0 > /sys/devices/virtual/net/br0/bridge/multicast_snooping
    post-up ip link set br0 address `ip link show eth0 | grep ether | awk
'{print $2}'`
```

The interface configuration uses the IP address assigned by the DHCP of the router. In the case of an existing static IP, it is necessary to change the *iface eth0*. In the Annex 5 it is possible to find the proper configuration of each raspberry and local computer to enable the connection in the switch number 3 on the Agropolis. Once the network interface has been set, it is necessary to restart the network to establish the new configuration, using the command:

```
sudo /etc/init.d/networking      restart
```

After the interfaces have been successfully configured, using the command *ifconfig,* the networks bridge *br0* must be set with the new configuration as shows the Figure 28.

```
br0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.4.110.205  netmask 255.255.255.0  broadcast 10.4.110.255
        inet6 fe80::ba27:ebff:fe65:bd5f  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:65:bd:5f  txqueuelen 1000  (Ethernet)
        RX packets 44669  bytes 3104879 (2.9 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1763  bytes 233120 (227.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 28. bridge interface configured

The border router will connect the sensor network through the Contiki application. CETIC-6lbr connects the IP network to the subnet previously configured on the local router. The raspberry configuration is not ready until the middleware on the machine has been set as border router.

The objective is to create the network interface on the Raspbian physical net (interface Ethernet –eth0-) and subnets (virtual bridge –br0-).
After the packages and modules installation on raspbian, it is necessary to configure the network as shown in the figure 32 as router mode. cetic – 6lbr, provides several motes configurations regarding the proposed network. The three configuration or modes in this platform are:

Device as ***Bridge***, ***Router and transparent bridge.***

- Bridge: this mode works over IPv6 with RPL for Wireless Sensor Network mesh topology, and acts as a NDP Proxy as shown in the Figure 29[43].

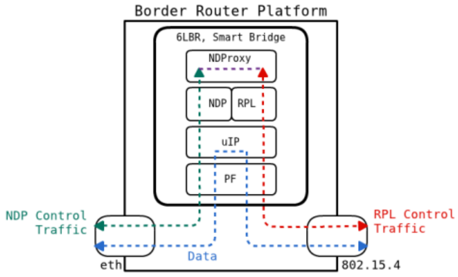


Figure 29. Bridge network configuration

- Router: This mode provides a second virtual interface working as a Gateway between the Ethernet and 6LoWPAN RPL. Redirecting the source address from RPL to eth ip local address as shown in the figure 30 [43].

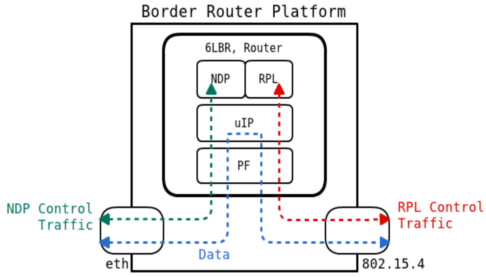


Figure 30. Router network configuration

- Transparent bridge: this mode behaves as a standalone bridge with basic switching capabilities, and all the packets are forwarded from the WSN to Ethernet and vice versa. The bridge can use the one-hop mesh with an IPv6 network using NDP or be statically routing the mesh network with an IPv6 network in the Figure 31[43].

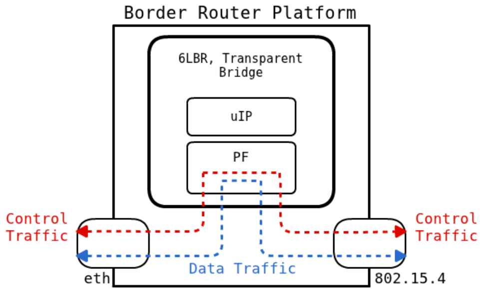


Figure 31. Transparent bridge configuration

The router mode in the Figure 30 must follow the next configuration parameters in order to invoke in a right way the router structure of the 6lbr.
Modify and write a file in the network interface of the 6lbr must be set with the command, *sudo nano /etc/6lbr/6lbr.conf.* Thereafter it is necessary to replace the script located in this file with the next instructions.

```
MODE=ROUTER
RAW_ETH=0
BRIDGE=1
DEV_BRIDGE=br0
```

```
DEV_TAP=tap0
DEV_ETH=eth0
RAW_ETH_FCS=0
DEV_RADIO=/dev/ttyUSB0
BAUDRATE=115200
IFUP=/usr/lib/6lbr/6lbr-ifup
IFDOWN=/usr/lib/6lbr/6lbr-ifdown
LOG_LEVEL=6
LIB_6LBR=/usr/lib/6lbr
BIN_6LBR=$LIB_6LBR/bin
WWW_6LBR=$LIB_6LBR/www
IP_CONFIG_FILE=/etc/network/interfaces
```

After the network configuration, it is necessary to disable the port ttyAMA0 due to the possible collision between port interface AMA0 and ttyUSB0 that will be used in this project.
The characteristic error appearing regarding the availability of AMA0 is the conflict of multiple serial interfaces discussed in the *section 4.3*
Other 6lbr configuration modes can be set on the devices following[43], and for the case used in the master thesis it is necessary to choose the router mode. As many projects that works with 6LoWPAN over IPv6, the router mode works as a gateway [46].
Regarding the dataflow and packet control the system operates using the Contiki-MAC with CSMA/CA. While other existing protocols reduce the power consumption optimizing the transmission power of each node [22], using models to the define the wireless sensor network is highly recommendable for ultra-low

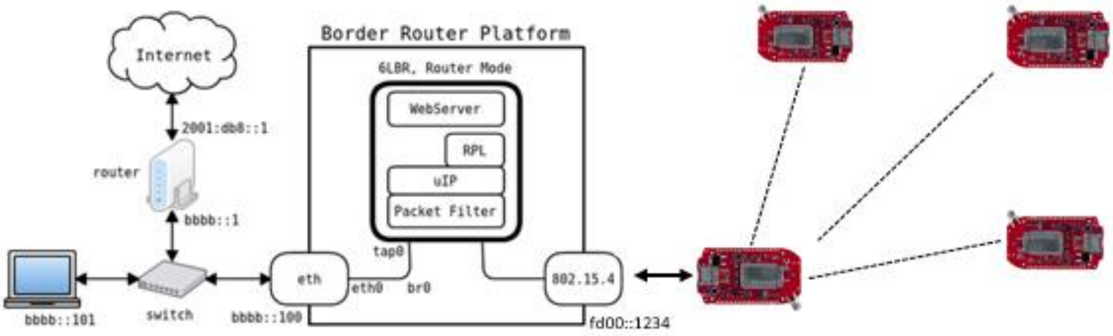


Figure 32. **Border router configuration**

power consumption in addition of LP devices.
The router mode in 6lbr route inside RA (Router advertisement) transmit messages that Linux ignores by default, therefore it is recommendable to set the next commands as super user.

```
sysctl -w net.ipv6.conf.br0.accept_ra=1
sysctl -w net.ipv6.conf.br0.accept_ra_rt_info_max_plen=64
```

At this point, the coordinator mote must be plugged to the border router and it should appear as an external component detected on the machine. The simpler way to observe the devices connected in Linux is using the command *sudo lsusb,* another method to search is using:

```
sudo dmesg | grep tty
```

The device detect the name of the interface connected via USB (cp210xx). The default interface must name as ttyUSB0, otherwise the dev_radio in 6lbr.conf should be assigned with the port interface detected.

Once the USB interfaces had been detected configure in the radio, the Raspberry must compile the slip radio script on the Zolertia RE-MOTE, initiating the coordinator node and listen the channel. The slip radio is required to open the 6lbr webserver and obtain the IP address of the network.
The slip radio script is located in ***/6lbr/examples/ipv6/slip-radio*,** in the folder the platform must be compile and debug.

```
sudo make TARGET=zoul savetarget
sudo BOARD=remote-revb make slip-radio.upload PORT=/dev/ttyUSB0
```

After the cc2538 RE-MOTE compilation and the mapping address on memory. The border router can be started runing the 6lbr service in the machine.

```
sudo service 6lbr start
```

For successful initialization is possible visualized the file *6lbr.ip.* This file must be located on the folder */var/log/.* The IP address texted on 6lbr.ip is the cetic website route, allowing an external computer connection on the same net than the gateway. Using the search bar from the browser is possible enter to the framework of 6lbr see annex 6, the next URL is assigned by default by the console and it works over IPv6, http://[bbbb::100]/. The sensor nodes located in the network may be configured with the boot loader, debugger or the border router using the USB interface.
A virtual gateway can be generated between the address fd00::/64 and the bridge over ethernet bbbb::100.

```
route -A inet6 add fd00::/64 gw bbbb::100
```

The cetic-6lbr framework examples are allowed to simplify the process for flashing and compiling the C scripts. Reproducing the same procedure done with the coordinator node, it is possible to compile each sensor node from cetic-6lbr using the libraries python-magic, intelhex and gcc-arm-none-eabi.
The sensor nodes are configured with a simple UDP package transmission and is possible to identify using the Cetic website as it is indicated in the annex 6, the probe of data transmission can be observed in the annex 7.
The server configuration shows the data on a user interface. The website must be friendly with the customer with a safety access to any person connected into the university network. Many alternatives could be implemented for SaaS (Software as a Service) and the cloud service available in several websites on internet.
The [47] created a network system using a template that support the microcontroller MPS430 among others. The comparison table at the end of the article in [48] explain in details the main different between the open source platforms available online. The main characteristic to select thingsboard.org is the availability to create a service insitu. In other words, the server can be located into the raspberry pi b+. Considering the administrator who has the full control of the server and services provided for the free platform.
Consequently the technical support must be specialize on this device and the maintenance goes for the manager of the system.
The installation of the platform is available in [49]. Raspberry is supported in the installation guideline which required last version of JAVA and PostgreSQL (recommended) for a properly use.
Be patient, the service required time to initialize on the localhost, it is possible to make the first login with the next credentials:

Username: tenant@thingsboard.or
Password: tenant

The credential can be changed on the PostgresSQL. The Entity creation is explained on the starter guide as same as the client construction for the MQTT URL and the ACCESS TOKEN. Those values are required to connect the UDP Server script to the IoT platform. Consequently, the MQTT BROKER implementation where the JSON data frames from are saved on the database, see Figure 33.
The application is constructed using AngularJS for the web browser. The queries required the IP address from the localhost, the broker, the topic name (v1/devices/me/telemetry) and the access password. Following installation steps from the website: https://thingsboard.io/docs/user-guide/install/rpi/
The dashboard includes several types of widget as examples designed by the platform. Those widgets can be used by free, other plugins are develop for the professional edition (PE). To export data in csv/excel format is require pre-install

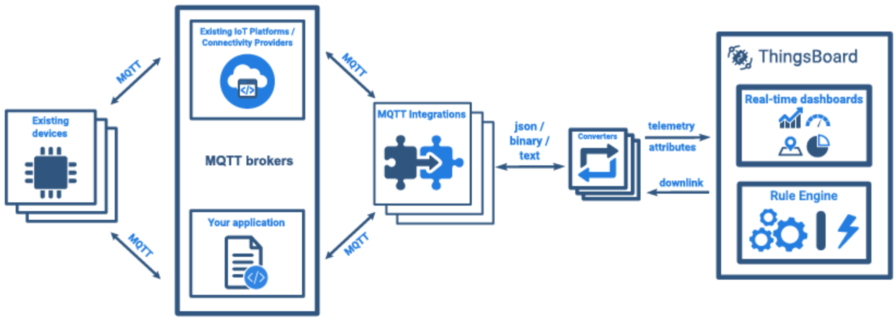


Figure 33. MQTT Integration between the Border Router (Existing Device) and the server with Thingsboard

the PE version otherwise in the free platforms gives the option to develop from sketch the widget that you required.
This project required the CSV data export and the construction of the widget using JavaScript on AngularJS. The main instruction of the name values are given in thingboard.org. The script of the widget designed can be observed in the annex 8. The final template of the dashboard can be observe in the figure 34.

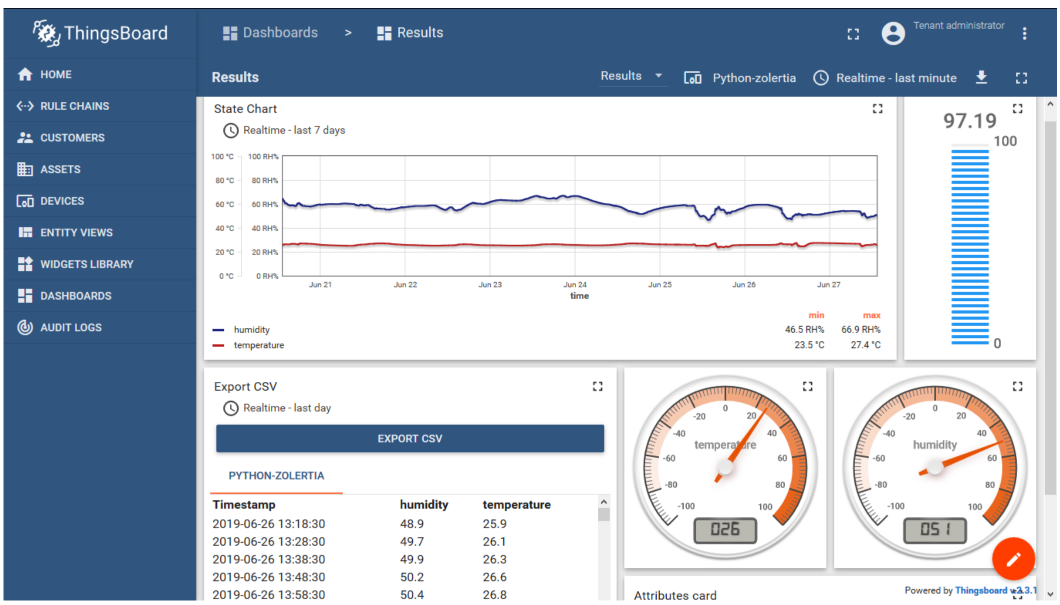


Figure 34. Dashboard of the results from the sensor node to the server

CHAPTER 4. EVALUATION AND RESULTS

This chapter is evaluating the system on the field and explaining the end functionality. The results gathered in this thesis are from the laboratory 125 in the building 4 where the network is installed. The values used in this section are compared with the environmental behavior of the Agropolis. The final application is aimed to be used in the greenhouse. The first testing scenario was simulated with three sensors nodes to confirmed the properly the script configuration. The final node implemented in the real scenario the number of sensor nodes have been limited to one considering the sensors availability.

4.1 Simulated scenario

The network configuration could create implications in the implementation. Therefore, it is necessary to simulate a scenario where the node could interact using the examples that will be implemented from Contiki. Many examples are on the old Zolertia Z1 version, considering the migration to RE-MOTES following[37] in the wiki of 6lbr. Furthermore, the internal sensors(due to the lack of virtual sensors) are used in the simulation following the section 3.5.7 from the book[20]. The examples are developed to support platforms already designed for the contikiOS. For example: Skymotes, Zolertia devices (z1 and zoul/re-remote), ST, wismote, nordic family nrf52, among other possible observed in[43].
The Figure 35 shows the network configuration implemented in this scenario, including some interferences and reduction of the transmission range and reception. The node 1 is configured using the *rpl-udp/udp-server.c* script, working as coordination which concentrates the packets from the sensor motes, and transmitted to the slip-radio that is connected in the border router.

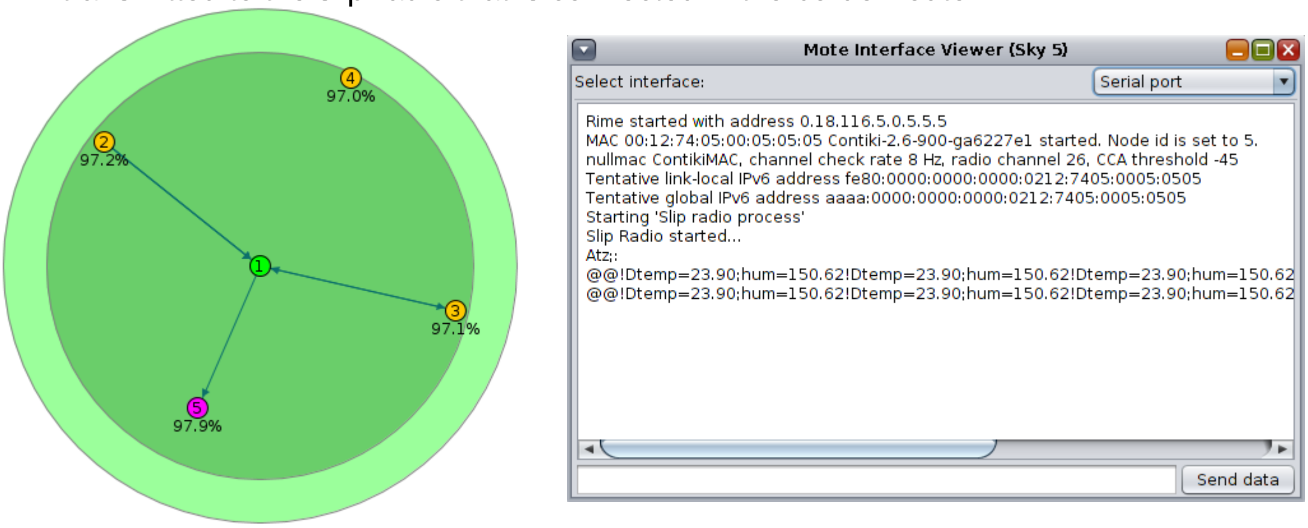


Figure 35. Simulation in Cooja using the rpl-udp example as guide

After the example configuration fits with the thesis idea, understanding and modifying the scripts is required considering the customer and devices limitations. In the *section 2.1* explained the variables where the system must focus.
The udp-client/udp-server script configuration could be modified following[20]. The sensors configuration could be imported to the main transmission/reception algorithm. In addition, the DHT22 library has to be set and the path files of the header and main scripts must be properly establish. In the simulation, the internal sensors on board used in the SkyMote had been used to test the script on the

Annex 3. Once the script compile successfully on Cooja the values on the right side of the Figure 35 are plot in the interface viewer.
The contiki directories must reach the paths created into the Makefile source.

4.2 Laboratory scenario

After the simulation, the next step is recreate a controlled environment where the node configuration and server could be managed. Stable condition could be perform on the laboratory 125 from the building 4. The EETAC faculty was selected for testing regarding the Ethernet port availability, the node where located in several point and measure the variables. The best location for the node is the first floor laboratory as the Annex 13 exposed.

First of all, the data frame must be transported from the node to the border router though one terminal port to the server. The packet transmission script follow the next structure:

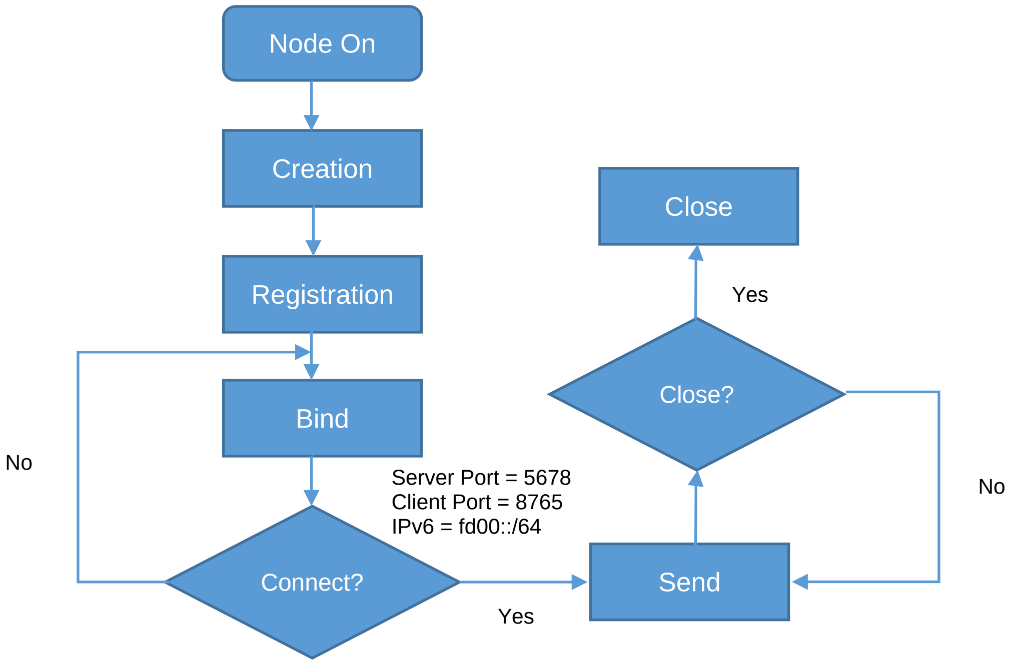


Figure 36. Flow Chart for UDP API

The UDP-client followed the figure 36 and maintained the send loop on the client port 8765. The border router is always listening the channel meanwhile the server script running over Python in the annex 9 create a web socket sniffer the server port 5678. The throughput goes into buffer (1 Kb) in the client port 8765 where UDP-server port is configured. The buffer contained an object structure that must be split and reassembly emulating a JSON structure, for the forward transmission on IPv4 using MQTT.

The main inconvenient is to emulate a full JSON structure over the object class received. Paho-mqtt do not support JSON objects and the string class has to be dump the JSON by segments. On the annex 9 is explained the solution to avoiding possible transmission problems. The values from the DHT22 are observed in the Figure 37, using the command *sudo make login PORT=/dev/ttyUSB1.* The loop created on the script shows the output from the port in use, otherwise exist the possibility to avoid devices specification in case that no other node is connected.

```
ID: 175, core temp: 21.429, Temperature 24.5 °C, Humidity 69.7 RH, batt: 3236, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 21.667, Temperature 24.5 °C, Humidity 69.8 RH, batt: 3236, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 21.429, Temperature 24.5 °C, Humidity 69.9 RH, batt: 3235, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 21.667, Temperature 24.5 °C, Humidity 69.9 RH, batt: 3236, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 22.143, Temperature 24.6 °C, Humidity 70.1 RH, batt: 3236, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 21.905, Temperature 24.7 °C, Humidity 70.2 RH, batt: 3235, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 22.143, Temperature 24.7 °C, Humidity 69.9 RH, batt: 3237, counter: 1
Send readings to 1, count 12'
ID: 175, core temp: 22.381, Temperature 24.8 °C, Humidity 69.8 RH, batt: 3237, counter: 1
Send readings to 1, count 12'
```

Figure 37. **DHT22 reading from de node**

The udp-client reports to the border router using tunneling from the localhost (127.0.0.1) to the bridge. In order to check the bit traffic the Figure 38 shows the configuration of the tunnel and the packets transported from the 6LoWPAN network. For more details when the border router is created see Annex 10.

```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 127.0.1.1  netmask 255.255.255.255  destination 127.0.1.1
        inet6 fe80::1  prefixlen 64  scopeid 0x20<link>
        inet6 fd00::1  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::72a0:1505:620:5f27  prefixlen 64  scopeid 0x20<link>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 80  bytes 4800 (4.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 37  bytes 3336 (3.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Figure 38. Tunnel created using border-router.c

The packets captured are taken from the port 5678 where the python script is listening and printed the value on the command prompt. In the Figure 39 shows the packets received from the sensor node with the JSON structure used in the annex 9. The MQTT generated and send through the port 1883 using the URL assigned in the script.

Before running the script in the annex 9 the border-router.c (see Annex 10) had been executed on the border router.

```
UDP6-MQTT server side application V2.1
Started 2019-06-17 08:23:03.312381
UDP6-MQTT server ready: 5678
msg structure size:  12

2019-06-17 08:23:27 -> fd00::212:4b00:60d:b2da:8765 12
***
id:175  counter:256  core_temp:14426  temperature:251  humidity:689  battery:42508
***
MQTT: Publishing to {0}... 0 (175)
MQTT: Published 1
```

Figure 39. Sensor data transmission using the UDP client service.

The values sent via IPv4 to the MQTT broker are shown in Figure 34 using an static IP address, in order to publish the data is required been subscribed in the topic created on the dashboard that it is known as an entity.

4.3 Errors and solutions

The errors obtained compiling and debugging this project are frequently and this section shows the more commons, including the solutions. The errors depends of the scenario. For example: the node Makefile route, when the webserver

service is started the border router waits the slip radio connection and the node configuration do not allow a successful memory flashing.
The folder located in */var/log/6lbr.log* allowed a first overview over the possible warnings or errors. The big error has texted on the file *6lbr.err*.
The *waiting for ttyUSB0* connection is a normal *log* appearing considering the absence of a node in the port (node unplugged), using *dmesg* | *grep tty* is possible verified the USB serial interface connectivity on the border router.
The log *fetching to a MAC address* happens when the RE-Mote is not debugged with the script *slip-radio*. The *bin* file has not set the ttyUSB0 as the serial interface configured in */6lbr/6lbr.conf*.
Open the port in the nodes required to use the command *sudo make login /dev/ttyUSBx* depending of the USB node interface.
Any conflict with the serialdump-linux is recommendable follow the next commands:

```
cd 6lbr/tool/sky
sudo rm serialdump-linux
sudo make serialdump
sudo mv serialdump serialdump-linux
sudo chmod -x serialdump-linux
```

The motes required the python magic and intelhex libraries updated and working properly otherwise the programming of the nodes will not compiled.
A common error regarding the previous modules missing could be fixed ensuring the existence of java-sdk and python-pip properly installed.

```
sudo update-alternatives --config java
```

The synchronization problems are log as: *ERROR: Timeout waiting for ACK/NACK after 'Synch (0x55 0x55)'* , this issue can be solved setting the flashing mode manually on the RE-MOTE. The programmable mode can be enabled pressing the debug button and hold it while push the user button for a while and release the user button first followed by the debug button. Other log error as *'Get ChipID(0x28)'* solution is explained and solved on the issue #1533 [50]. The error *'device reporst rediness to read but returned no data'* happen when the serial port AMA0 is enable and USB0 is trying to access at the same time creating a port conflict.
The sensor compilation uses modules in the script for motion transductors that are not available in the packet downloaded of 6lbr. For this reason, in case of missing this script *motion-sensor.c* is necessary to create a C file with the algorithm in the next url.
https://github.com/alignan/contiki/blob/iot-workshop/platform/zoul/dev/motion-sensor.c
The implementation of the DHT22 has been done following the example in [51] and the package structure in:
https://github.com/contiki-os/contiki/blob/master/examples/zolertia/zoul/test-dht22.c
Possible problem can appear regarding the sample time and the files existence or pointed in the MakeFile.
Before start the border router is necessary to stop the 6lbr service, therefore both scripts runs over /dev/ttyUSB0 and create a conflict on the connection.

4.4 Results

The results are divided in two sections, farm and laboratory results The data for greenhouse scenario has been provided by the manager of the Agropolis.

4.4.1 Agropolis results

First of all, the variable relevance of the temperature and humidity must be independent variables with lower correlation, to verify this affirmation is necessary to prove the relationship between environmental measured and literature review. Furthermore, the Agropolis provided the information obtained in the last years using their greenhouse system HortiMax [52].
The slot of time selected by the farm is one hour per sample every day in one year, for this reason the considerations taken in this section are enclose to the values from previous years. The full data gathered from the HortiMax has many missed values. One discarded consideration was merged all the data tables but the different type of structure and blank spaces make chaotic the possible analysis.

The Figure 40 shows the correlation between variables after being processed on Python using annex 11, it is possible to confirm the low correlation between temperature and humidity. It means that those variables are independent even though exist second of them that it's possible to discard regarding their high correlation between themselves (I'm talking about Temperature and Humidity, 1 and 2). The CO2 measurement are not selected considering this variable useless in this use case. After cleaning the data from the values that this project will not take into account (see figure 41) is appreciable that the intensity light expressed as radiation is high correlated with the temperature.

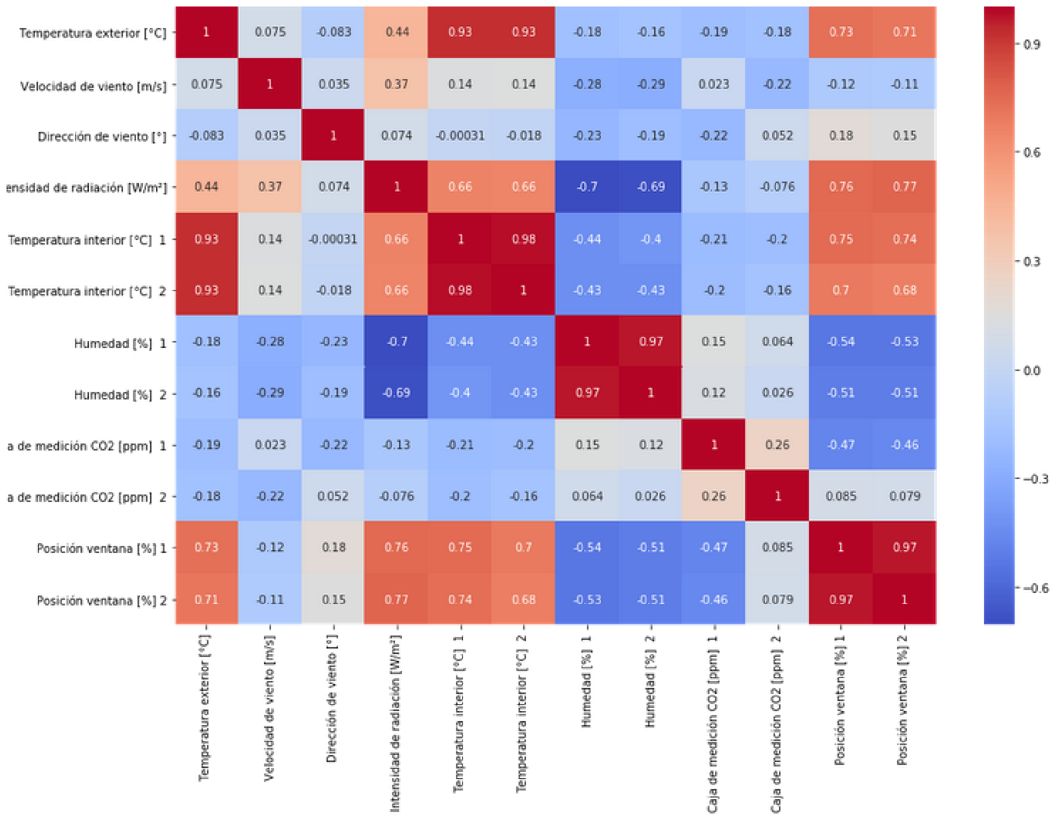


Figure 40. Full matrix correlation

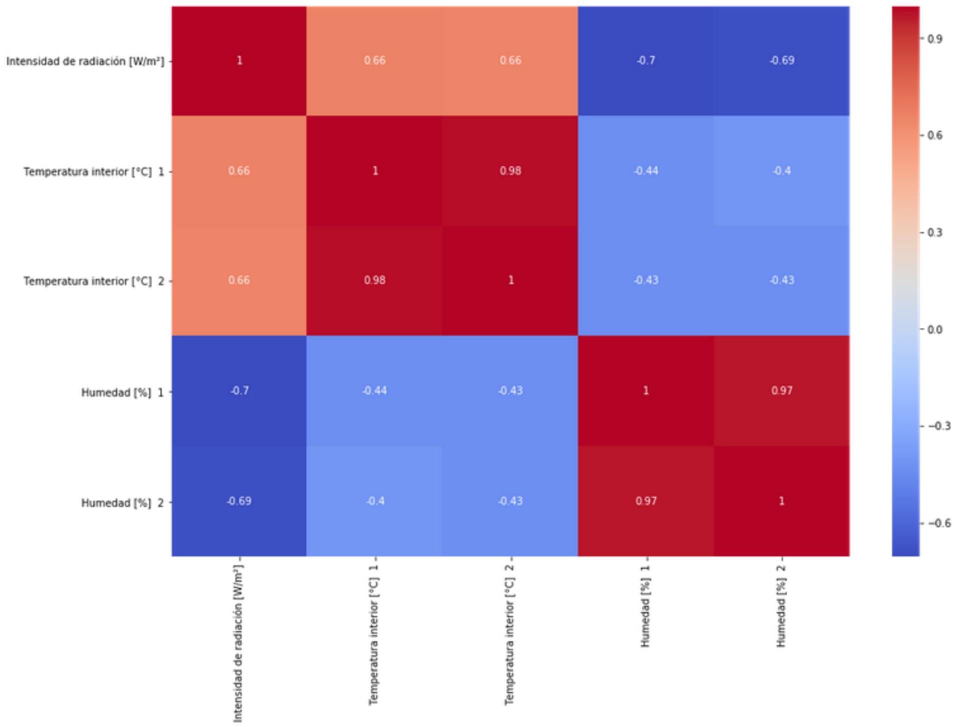


Figure 41. Cleanned matrix with independent variable

Humidity 1 & 2 are highly correlated as same as temperature 1 & 2, for this reason the figure 42 will be expressed only in terms of temperature and humidity 1. In addition, the Figure 42 explained how many degrees the environment could be considered more humid and vice versa. It is a first overview of the environmental behavior. Those details have been collected with the information of greenhouse in 2018 creating a map of the environmental conditions controlling the internal space to keep the crops in good standard condition.

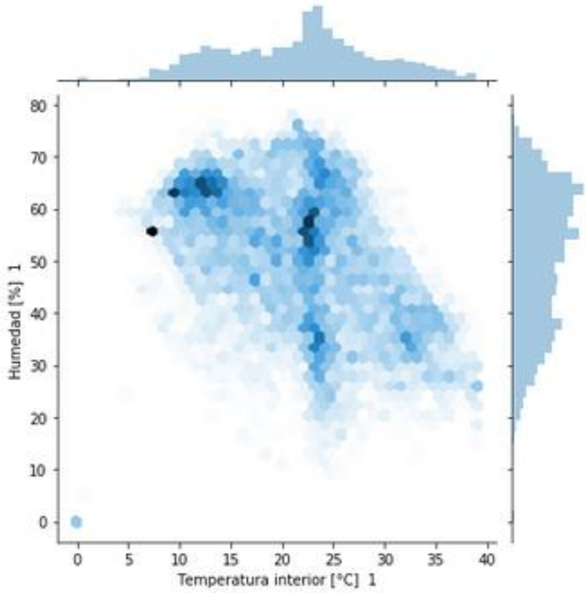


Figure 42. Relationship between temperature and humidity

4.4.2 Laboratory results

Location and disposition of the material had been selected the accessibility and availability of the infrastructure. Regarding the distance of the packet beamed via radiofrequency showed in [3] has a maximum range about 200 meters on direct view and the power consumption in comparison of other modules[53][31] improve in low power consumption[19].
Even though the measures had been obtained inside a building where the obstacles are higher, and the electrical lines contribute to reduce the range and performance of the measurements. In the annex 12 the location of the nodes shows the real indoor performance. The color scale are given by the next performance. Blue, border router location. Green, Successful data transmission. Red, failure transmission and close the MQTT listener. Trying to send total 5 packages per point.

Total Packages N: 65	Floor 1	Floor 2	Floor 3
Succesull	83%	75 %	0 %
Fail	17 %	25 %	100 %

Table 2 Successful vs failure rate

The information collected for the sensor will be saved in Postgress database where the time, temperature and humidity will be recorded and located into a table. The time format used in AngularJS system is in millisecond, which means that the initial value goes from Jan 1st of 1970 until now.
The table contents the information necessary to calculate the possible values of the evapotranspiration and vapor saturation.
The temperature and humidity values created a local picture of the greenhouse. The implementation of irrigation plant regarding the absolute minimums in dry season.
The MQTT explained in section 1.4.5 required a callback for messaging acquisition on top of the transport layer and the UDP protocol. The sensor information will be located in localhost using the UDP-MQTT-server script in the Annex 9*.* The UDP-client packet transmission in Annex 3 open the port 5678 for the server and the 8765 for the client.
The server script is a compilation of the UDP-server in the address fd00::1 due to the border router in the Annex 10.
The information structure comes into a JSON format, listening the port 5678. Therefore, the client sensor node acquired the data and encapsulate it for the transport. Meanwhile, the server platform obtained the data structure and published on the local address. The communication profile can be interpreted as a linear method UDP-C/MQTT/UDP-S, been *C for client and S for server*.

The values gathered in this master thesis allow the farmer a continuous variable observation where the variables can be used in the Penman-Monteith method. One of the standard methods to obtain the evapotranspiration that depends on factors as the *Solar radiation, Air temperature, air humidity and wind speed*[54]. The lack of resources and sensors supplies, required the use of the suppositions and possible estimations considering the values obtained in this memory. The sensor information allowed the mean calculus of a periodically air temperature samples, depending the case of study, period from minutes to hours in a lapse of 10 minutes.
The mean temperature is used to calculate the slope of saturation vapor curves[54], the relationship between the temperature and the vapor pressure

saturation in expressed in the equation 1, where the variable has a none linear behavior.

$$e°(T) = 0.610e^{\frac{17.27*T}{T+237.3}} \quad (1)$$

$e°: saturation\ vapour\ pressure\ at\ the\ air\ temperature.$

The values related to different possible saturation vapor pressure can be observe in the Annex 14[55]. In the figure 43 is recreate the vapor saturation using the dashboard.

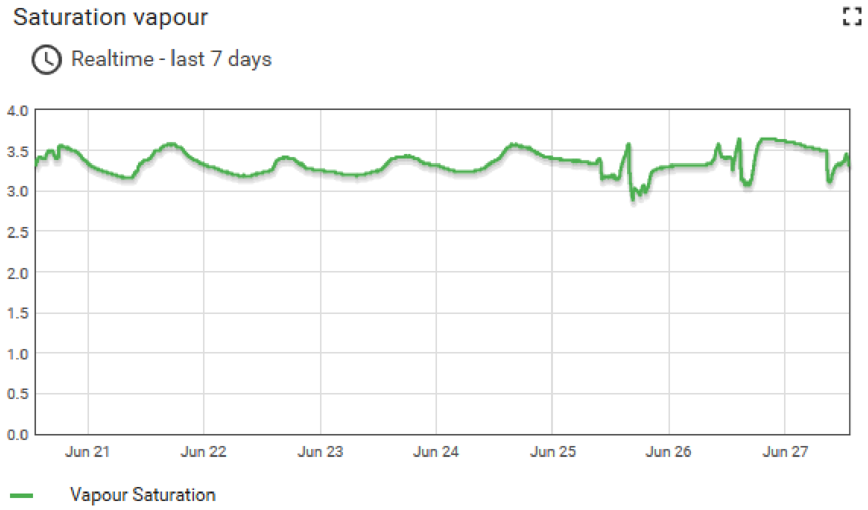


Figure 43 Vapour saturation pressure in Kpa

Annex 12 debugged the next values where contrast the laboratory results see Figure 44 and the values observe in the agropolis on the same period of time see Figure 45. The main difference is the number of samples per day. In addition, the possibility that the students can turn on the Air Condition.

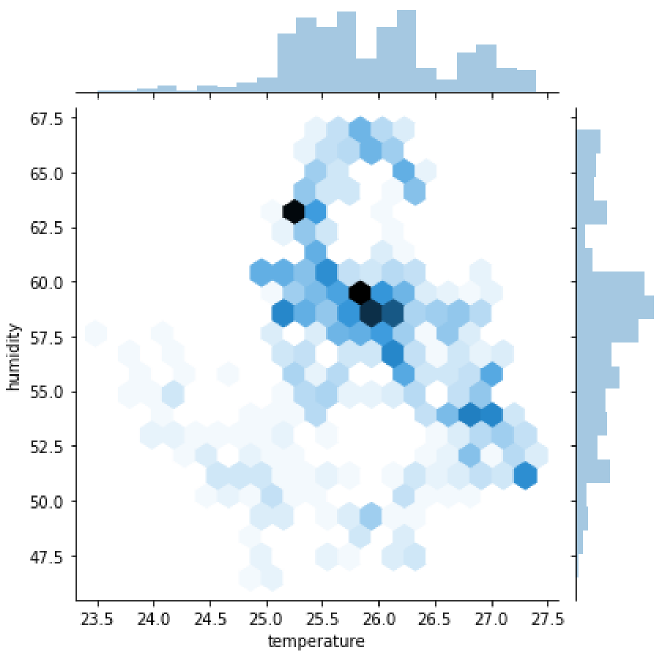


Figure 44. Laborary results from the 20 to 27 of july 2019

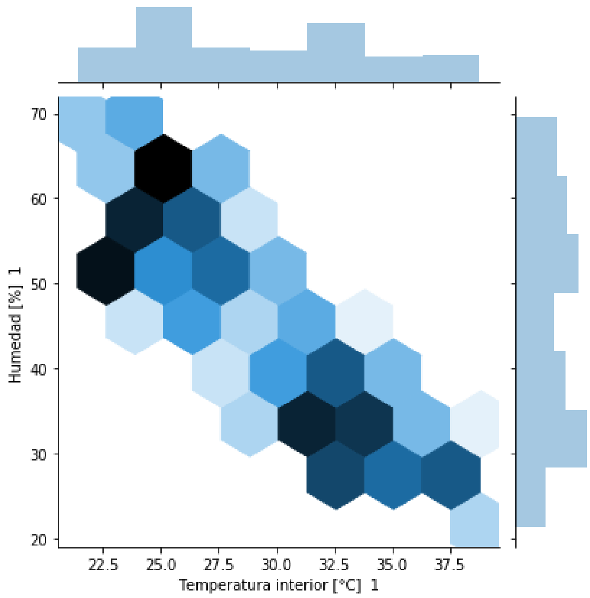


Figure 45. Agropolis results from the 20 to 27 of july 2018

The descendant diagonal explained the behavior of the temperature in the same range instead of the vertical trend of the complete year of 2018. The results of the laboratory has been affected by external impacts than humidity and temperature a component of human interaction make the diagonal slightly moved into the left region of the temperature axis. This movement can be considered as the absolute error of 5°C, furthermore considering the acclimatization and disposition of the data.

CHAPTER 5. CONCLUSION AND FUTURE WORK

5.1. Conclusions

The general objective of this thesis was completed for the designed and implementation a Wireless Sensor Network, applying IoT for a continuous monitoring. Regarding, the full implementation on Agropolis was not boarded considering the ports availability and configuration issues on the RAC number 3. The location accessibility and the bounded schedule limited the specific objective, but it was replaced for a controlled enviroment. The technology evolution generate big changes in the industry and lifetime of the object. New devices implementations are necessary to optimize and innovate the process in all the departments. The agricultural sector is aware of those changes which require techniques updates regarding the market high demand for products of better quality. Greenhouse crops monitoring moderates the possible impacts that the plant handled during certain periods of the year. Plagues and parasites presence are related to changes of weather and water retention of the crop.
The data acquisition of variables related with the possible environmental changes could give the farmer the possibility to create a contention plan in order to improve the crop’s production.

The network system had been tested under the laboratory conditions instead of the greenhouse one, and considering the final implication on the environment.
In terms of security, the centralization of the data could be an issue regarding the data migration to internet. For applications of low risk as isolated greenhouses or university laboratories this situation does not imply a big impact. Indeed, the security level can be establish as used in this project by 2 filters: the necessity to be accredited in the server platform and use the university VPN in order to establish connection into the network.
Several agricultural industries use meteorology station to gather information, whereas the solution showed in this thesis rely on the use of sensor nodes with a maximum coverage range of hundreds of meters on directed view. The objective was to create a real time network with data gathered in strategic points, analyzing the behavior of the crop and weather impact. On the other hand, the measurement have been realized on the laboratory infrastructure.
The system can be considered for greenhouse and farms of small scale, no more than 2 hectares. Therefore, the reliability of the system can be affected for the fading and shadowing of the plants and workers located in the region. In practice, I observed that the orientation of the node and cumulative obstacle can impact the performance as observed in the annex 13.
The fading interference considering other radiofrequency sources as sensors nodes operating at the same wake up time weren’t used in this project. The shadowing introduced by the plant or wall can be assume as a disruptive blackout for the package transmission. On the other hand, the transmission period can be manipulated transmitting one package by minute or hour, giving more spaces between preambles and radio wake up.

The project developed in this thesis has as a main objective the local monitoring and the implementation of the Wireless Sensor Networks showing the advantage of real time acquisition. Regarding the multiple types of sensors that could be implemented on the network, the data necessary for updating environmental tables is in constant construction. The techniques used for mining data can improve and optimize the decisions taken in base of the datasets. The connection to the cloud gives a longer support for online platforms joined in the same data structure. The software defined in this thesis is a preamble for further data extraction recreating models using the FAO Penman-Monteith energy method[55].

In my opinion, the network presented in this memory using the 6LoWPAN protocol has to be spread among the users, due the benefits of the free band (ISM) space and low power consumption. The implementation of the IPv6 protocol provide numerous advantages, in this case many applications with several nodes used on this network protocol.

For the case of medium and big extensions of territory, it is recommendable to implement a system more reliable, with a direct connection to the IoT cloud or the server. New technology trends as NB-IoT and LoRa have been implemented in many companies for open spaces, considering the protocol robustness against shadowing and fading impacted on the network. The agricultural sector tends to a LoRa integration, due to the high market demand and as shown in the literature review of this protocol. Meanwhile, the NB-IoT still in development process, but the increase of the solutions using this tool indicates a possible future use.

5.2. Future work

This thesis opens new avenues for future research. An example could be the implementation of this network in a real greenhouse and the measure of the packet losses regarding the shadowing of the plant. In addition, the exterior condition impact of the machine, how the salinity and water can affect the RE-Mote and the autonomy after apply the low power consumption mode can be explored.

Considering another requirements of the final client, the implementation of other sensor, programming and executing different script regarding their alternatives on the market. Furthermore the modification and evaluation of other MAC scripts, using topologies for parent-son regarding the location can be investigated in order to reduce the power consumption.

Finally, the study of the possibilities to use a LoRa module and the implementation of the network with this technology can provide meaningful results in the agricultural sector.

REFERENCES

[1] P. Beinschob and C. Reinke, "Graph SLAM based mapping for AGV localization in large-scale warehouses," in *Proceedings - 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing, ICCP 2015*, 2015, pp. 245–248.

[2] M. F. Sanchez, "Implementation of a personal area network for biomedical measurements for Internet of Things (IoT)," 2018.

[3] S. Molinero Baixas, "Internet of Things (IoT) implementación con nodos Zolertia RE-Mote," Feb. 2018.

[4] I. Texas, "TI CC2538: Users Guide," no. May, 2013.

[5] L. Chen, "Scholarship at UWindsor Security Management for The Internet of Things," pp. 4–14, 2017.

[6] M. Gurtu, "Anaphoric relations in Hindi and English," 1992.

[7] B. Pang, "Energy Consumption Analysis of ARM- based System," pp. 1–68, 2011.

[8] "LPWAN networks | Internet of Things Wiki | FANDOM powered by Wikia." [Online]. Available: https://iot-fpms.fandom.com/wiki/LPWAN_networks. [Accessed: 26-Jun-2019].

[9] "Internet of Things forecast – Ericsson Mobility Report." [Online]. Available: https://www.ericsson.com/en/mobility-report/internet-of-things-forecast. [Accessed: 05-Feb-2019].

[10] "Bluetooth Smart Technologyfrom Trend To Standard | Bluetooth Technology Website." [Online]. Available: https://blog.bluetooth.com/bluetooth-smart-technologyfrom-trend-to-standard?_ga=2.216819093.284992924.1549380961-803718038.1549380961. [Accessed: 05-Feb-2019].

[11] Y. Kim, R. G. Evans, and W. M. Iversen, "Remote Sensing and Control of an Irrigation System Using a Distributed Wireless Sensor Network," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 7, pp. 1379–1387, Jul. 2008.

[12] "Bluetooth Technology: What Has Changed Over The Years." [Online]. Available: https://medium.com/jaycon-systems/bluetooth-technology-what-has-changed-over-the-years-385da7ec7154. [Accessed: 05-Feb-2019].

[13] "Topology Options | Bluetooth Technology Website." [Online]. Available: https://www.bluetooth.com/bluetooth-technology/topology-options/. [Accessed: 27-Jun-2019].

[14] Nordic Semiconductors, "Bluetooth 5 - nordicsemi.com." [Online]. Available: https://www.nordicsemi.com/?sc_itemid=%7BF6228DC4-3FFC-47D2-9139-7D42FDADB5CA%7D. [Accessed: 25-Jun-2019].

[15] C. N. Templeton, N. V. Carlson, A. Leon-Garcia, and I. Widjaja, "Communication Networks," *Ref. Modul. Life Sci.*, pp. 1–24, 2018.

[16] "Three Ways to Set the Stage for Incredible Connected Experiences in the Smart Home - IoT@Intel." [Online]. Available: https://blogs.intel.com/iot/2018/05/30/three-ways-to-set-the-stage-for-incredible-connected-experiences-in-the-smart-home/?_ga=2.7395570.1799002128.1561655113-624775127.1561655113. [Accessed: 27-Jun-2019].

[17] "Home." [Online]. Available: https://www.threadgroup.org/. [Accessed: 05-Feb-2019].

[18] "OpenThread." [Online]. Available: https://openthread.io/. [Accessed: 05-

Feb-2019].

[19] C. E. Hernández Chulde, “Internet of Things (IoT) implementation with nodes based on low power MSP430 microcontroller,” 1000.

[20] A. Liñán, C. Alvaro, V. A. Bagula, M. Zennaro, and E. Pietrosemoli, “Internet of things in 5 Days.”

[21] T. Members and M. Open, “ZigBee Technical Overview,” *Control*, 2008.

[22] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet - Shelby - Wiley Online Library*. 2011.

[23] Keysight, “IEEE 802.15.4g Standards,” 2012.

[24] B. (Avnet S. Matilla, “IoT Connectivity,” Spain, 2018.

[25] K. Radnosrati, G. Hendeby, C. Fritsche, F. Gunnarsson, and F. Gustafsson, *Performance of OTDOA Positioning in Narrowband IoT Systems*. .

[26] M. Richardson and I. Robles, “RPL-Routing over Low Power and Lossy Networks.”

[27] M. Yuan, “Conociendo MQTT ¿Por qué MQTT es uno de los mejores protocolos de red para el Internet de las Cosas?,” 2018.

[28] “Mqtt-s | MQTT.” [Online]. Available: http://mqtt.org/tag/mqtt-s. [Accessed: 20-Jun-2019].

[29] A. Stanford-Clark and H. L. Truong, “MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1.2,” 2013.

[30] B. Karthikeyan, M. Karthikeyan, R. Narayanan, and C. Suresh, “A Novel for Unmaned Irrigation Based on WSN Architecture,” *Int. J. Adv. Res. Innov.*, vol. 3, no. 1, pp. 188–192, 2015.

[31] J. Granda, M. Almanza, J. Fontalvo, and M. Calle, “Irrigation Measurement System for Dry Areas Based on WSN,” *Int. J. Interdiscip. Telecommun. Netw.*, vol. 9, no. 3, pp. 10–20, 2017.

[32] Senninger Irrigation Inc., “Low Pressure Sprinklers Make Farming More Sustainable | Senninger Irrigation,” *SABI Magazine*, 2015. [Online]. Available: http://www.senninger.com/low-pressure-sprinklers-make-farming-more-sustainable/. [Accessed: 18-Dec-2016].

[33] NRCS, “Sprinkler.Irrigation.pdf,” in *NRCS Irrigation Guide*, Tennessee, pp. 1–40.

[34] R. D. Bliesner, D. Spare, and A. M. Beutler, “Can Irrigation Scheduling Improve the Yield/Consumptive Use Ratio?,” in *Impacts of Global Climate Change*, 2005, pp. 1–12.

[35] M. Buettner, G. V Yee, E. Anderson, and R. Han, “X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks * Low Power Listening, Adaptive Duty Cycling, Networking, Wireless Sensor Networks,” pp. 307–320.

[36] A. Dunkels, “The ContikiMAC Radio Duty Cycling Protocol,” 2011.

[37] “Migrate from Z1 to Re-mote.” [Online]. Available: https://github.com/Zolertia/Resources/wiki/Migrate-From-Z1-to-RE-Mote.

[38] N. Ramírez, “Internet of things implementation with Raspberry Pi,” p. 77, 2014.

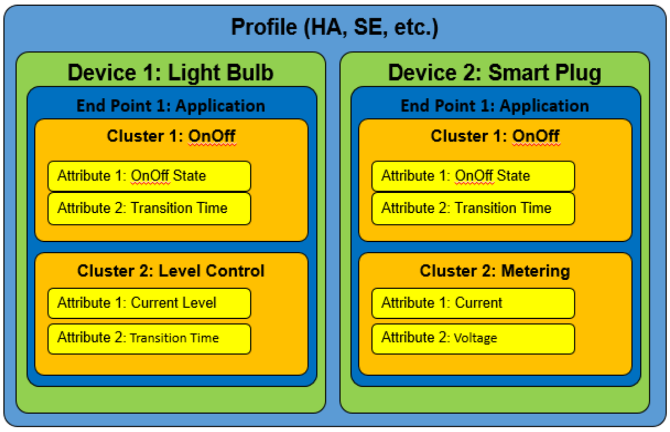
[39] Cytron Technologies, “DHT22 Temperature and Humidity Sensor,” pp. 1–6, 2019.

[40] A. Lignan, “Zolertia RE-Mote Platform.” [Online]. Available: https://github.com/Zolertia/Resources/wiki/RE-Mote.

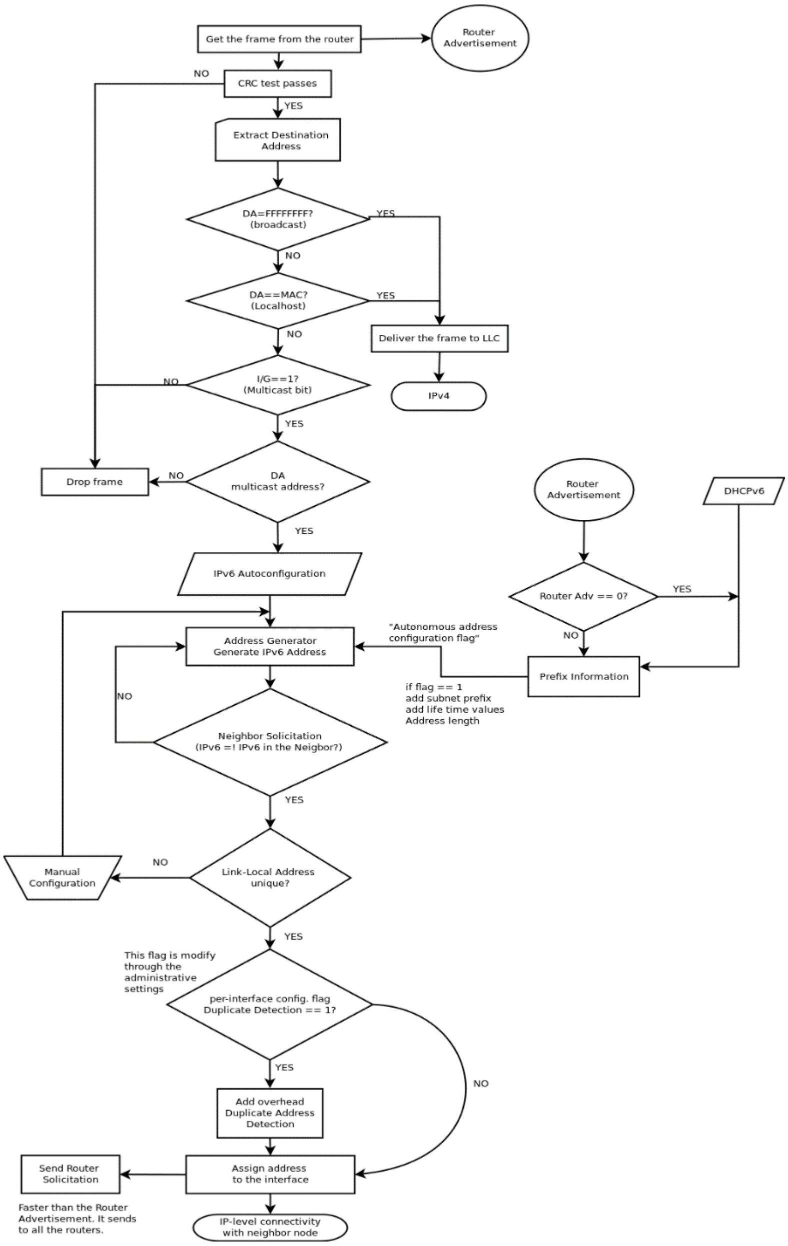
[41] “balenaEtcher - Home.” [Online]. Available: https://www.balena.io/etcher/. [Accessed: 05-Feb-2019].

[42] "Installing operating system images - Raspberry Pi Documentation." [Online]. Available: https://www.raspberrypi.org/documentation/installation/installing-images/. [Accessed: 05-Feb-2019].
[43] "CETIC-6lbr wiki." [Online]. Available: https://github.com/cetic/6lbr/wiki/Other-Linux-Software-Configuration.
[44] "Advanced IP Scanner – Explorador de redes de descarga gratuita." [Online]. Available: http://www.advanced-ip-scanner.com/es/. [Accessed: 05-Feb-2019].
[45] "Download PuTTY - a free SSH and telnet client for Windows." [Online]. Available: https://www.putty.org/. [Accessed: 05-Feb-2019].
[46] Z. Suryady, M. H. M. Shaharil, K. A. Bakar, R. Khoshdelniat, G. R. Sinniah, and U. Sarwar, "Performance evaluation of 6LoWPAN-based precision agriculture," *Int. Conf. Inf. Netw. 2011, ICOIN 2011*, pp. 171–176, 2011.
[47] J. Kukkamäki, R. Costa, V. Heck Júnior, and E. N. Bitencourt, "IoT Centralization and Management Applying ThingsBoard Platform."
[48] H2S Media Team, "9 Best & Top Open source IoT Platforms To Develop the IOT Projects |H2S Media." [Online]. Available: https://www.how2shout.com/tools/best-opensource-iot-platforms-develop-iot-projects.html. [Accessed: 17-Jun-2019].
[49] Thingsboard, "Getting Started | ThingsBoard." [Online]. Available: https://thingsboard.io/docs/getting-started-guides/helloworld/. [Accessed: 17-Jun-2019].
[50] P. Team, "Timeout waiting for ACK/NACK after." [Online]. Available: https://github.com/contiki-os/contiki/issues/1533.
[51] "Contiki-NG: DHT22 temperature and humidity sensor." [Online]. Available: https://contiki-ng.readthedocs.io/en/latest/_api/group__zoul-dht22.html#ga72fdbe6befa6c06eeb0118b6064e177c. [Accessed: 19-Jun-2019].
[52] "Hortimax – Hortimax." [Online]. Available: https://www.hortimax.co.uk/. [Accessed: 21-Jun-2019].
[53] E. Chain, R. Sara, J. Arias, E. Escorcia, and M. Calle, "OTS-WSN: A Wireless Sensor Network implemented with off-the-shelf components," in *IEEE Symposium on Wireless Technology and Applications, ISWTA*, 2013, pp. 327–332.
[54] "Chapter 3 - Meteorological data." [Online]. Available: http://www.fao.org/docrep/X0490E/x0490e07.htm#TopOfPage. [Accessed: 07-Feb-2019].
[55] "Chapter 4 - Determination of ETo." [Online]. Available: http://www.fao.org/docrep/X0490E/x0490e08.htm#calculation procedures with missing data. [Accessed: 07-Feb-2019].
[56] Sébastien Bonnet, "MQTT: un protocolo específico para el internet de las cosas | Digital Dimension." [Online]. Available: http://www.digitaldimension.solutions/es/blog-es/opinion-de-expertos/2015/02/mqtt-un-protocolo-especifico-para-el-internet-de-las-cosas/. [Accessed: 20-Jun-2019].
[57] NXP, "Product_Positioning_with_KWx_JNx_QN." .

ANNEXES



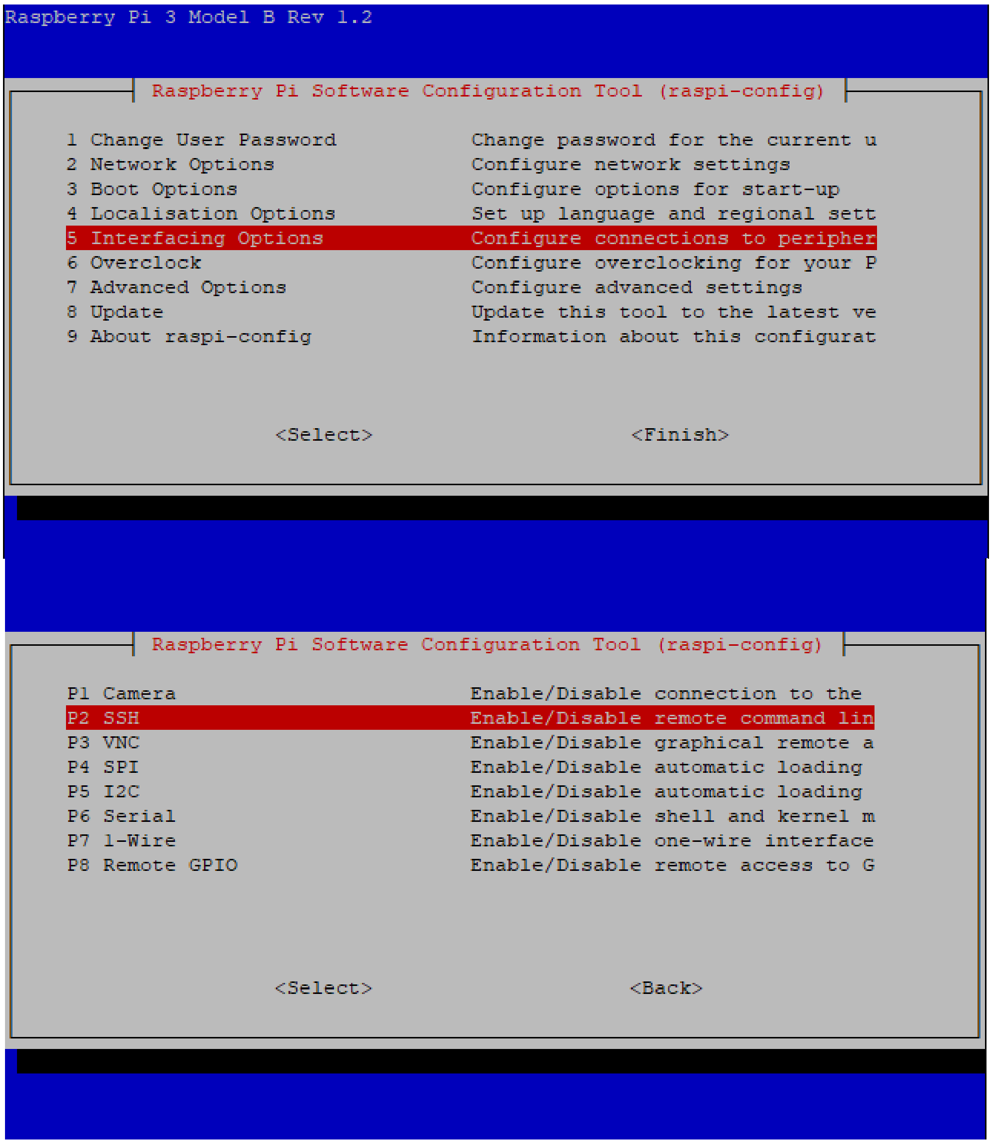
Annex 1. Profile composition of devices using clusters



Annex 2 Flow diagram of the auto configuration function of IPv6 devices

```
/*---------------------------------------------------------------------------*/
#include "contiki.h"
#include "lib/random.h"
#include "sys/ctimer.h"
#include "net/ip/uip.h"
#include "net/ipv6/uip-ds6.h"
#include "net/ip/uip-udp-packet.h"
#include "sys/ctimer.h"
#include <stdio.h>
#include <string.h>

#include "dev/adc-zoul.h"
#include "dev/zoul-sensors.h"
#include "dev/dht11.h"
#include "dev/button-sensor.h"
/*---------------------------------------------------------------------------*/
/* Enables printing debug output from the IP/IPv6 libraries */
#define DEBUG DEBUG_PRINT
#include "net/ip/uip-debug.h"
/*---------------------------------------------------------------------------*/
/* Default is to send a packet every 10 minutes */
#define SEND_INTERVAL   (600 * CLOCK_SECOND)
/*---------------------------------------------------------------------------*/
/*---------------------------------------------------------------------------*/
/* This is the UDP port used to send and receive data */
#define UDP_CLIENT_PORT   8765
#define UDP_SERVER_PORT   5678

/*---------------------------------------------------------------------------*/
```

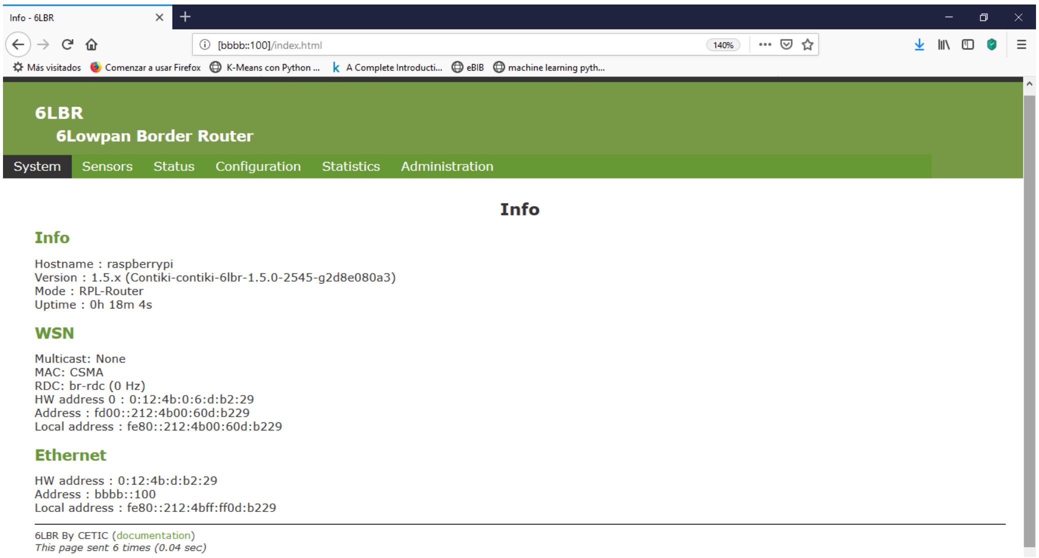
Annex 3 Node configuration code, click to open the object.

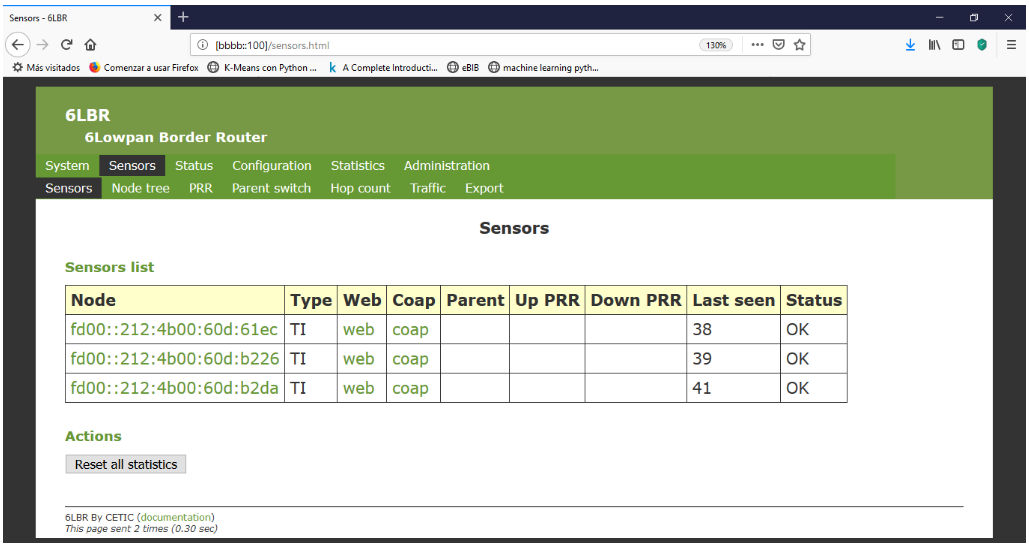


Annex 4.SSH Enable in Jessie Raspbian OS

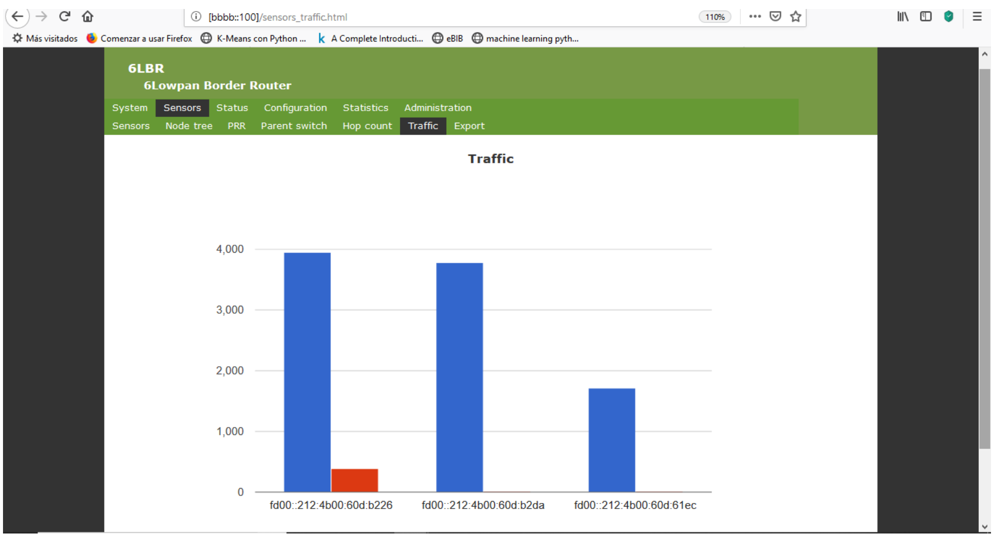
Devices	**Raspberry 1**	**Raspberry 2**	**Laptop**
IP	147.83.218.72	147.83.218.73	147.83.218.74
Mask	255.255.255.128		
Default Gateway	147.83.218.1		
DNS	Default		
IP name server	147.83.2.3		
IP name server	147.83.194.4		

Annex 5 Network parameters





Annex 6 CETIC 6lbr webserver operative & Sensor nodes connected to the coordinator



Annex 7 Data traffic registered by node

HTML Configuration

```
<div flex layout="column" style="height: 100%;">
  <md-button class="md-primary md-raised" ng-click="Download()">Export CSV</md-button>
  <md-tabs md-border-bottom>
    <md-tab ng-repeat="datasource in datasources track by $index"
      label="{{datasource.name}}">
      <table style="width: 100%;">
        <thead>
          <tr>
            <th>Timestamp</th>
            <th ng-repeat="dataKeyData in
datasourceData[$index]">{{dataKeyData.dataKey.label}}</th>
          <tr>
        </thead>
        <tbody>
          <tr ng-repeat="data in datasourceData[$index][0].data track by $index">
            <td>{{data[0] | date : 'yyyy-MM-dd HH:mm:ss'}}</td>
            <td ng-repeat="dataKeyData in
datasourceData[$parent.$index]">{{dataKeyData.data[$parent.$index][1]}}</td>
          </tr>
        </tbody>
      </table>
    </md-tab>
  </md-tabs>
</div>
```

JavaScript configuration

```
self.onInit = function() {

  self.ctx.$scope.datasources = self.ctx.defaultSubscription

    .datasources;

  self.ctx.$scope.data = self.ctx.defaultSubscription

    .data;



  self.ctx.$scope.datasourceData = [];



  var currentDatasource = null;

  var currentDatasourceIndex = -1;



  for (var i = 0; i < self.ctx.$scope.data.length; i++) {

    var dataKeyData = self.ctx.$scope.data[i];
```

Annex 8. Widget export dashboard

```
#! /usr/bin/env python


#----------------------------------------------------------#
# UDP example to forward data from a local IPv6 DODAG in thingsboard
# Juan Granda<jegranda.93@gmail.com>
#----------------------------------------------------------#
import sys
import json
import datetime
from socket import*
from socket import error
from time import sleep
import struct
from ctypes import *
import paho.mqtt.client as mqtt
ID_STRING      = "alpha version"
PORT           = 5678
CMD_PORT       = 8765
BUFSIZE        = 1024
ENABLE_MQTT      = 1
DEBUG_PRINT_JSON  = 1
MQTT_URL        = "10.4.110.207"
ACCESS_TOKEN      = "python_zolertia"
MQTT_PORT        = 1883
MQTT_KEEPALIVE    = 610
MQTT_URL_TOPIC      = "v1/devices/me/telemetry"
var1 = "core_temp"
var2 = "temperature"
var3 = "humidity"
```

Annex 9 Python script UDP forward data to thingsborad

```
using saved target 'zoul'
(cd ../../../tools && make tunslip6)
make[1]: Entering directory '/home/pi/6lbr/tools'
cc     tunslip6.c tools-utils.c   -o tunslip6
tunslip6.c: In function ‘stamptime’:
tunslip6.c:134:3: warning: implicit declaration of function ‘gettimeofday’ [-Wimplicit-function-declaration]
   gettimeofday(&tv, NULL) ;
   ^~~~~~~~~~~~
make[1]: Leaving directory '/home/pi/6lbr/tools'
sudo ../../../tools/tunslip6 fd00::1/64
********SLIP started on ``/dev/ttyUSB0''
opened tun device ``/dev/tun0''
ifconfig tun0 inet `hostname` mtu 1500 up
ifconfig tun0 add fd00::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 127.0.1.1  netmask 255.255.255.255  destination 127.0.1.1
        inet6 fe80::8aeb:c25d:a7f8:474f  prefixlen 64  scopeid 0x20<link>
        inet6 fe80::1  prefixlen 64  scopeid 0x20<link>
        inet6 fd00::1  prefixlen 64  scopeid 0x0<global>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

*** Address:fd00::1 => fd00:0000:0000:0000
Got configuration message of type P
Setting prefix fd00::
Server IPv6 addresses:
 fd00::212:4b00:60d:b229
 fe80::212:4b00:60d:b229
```

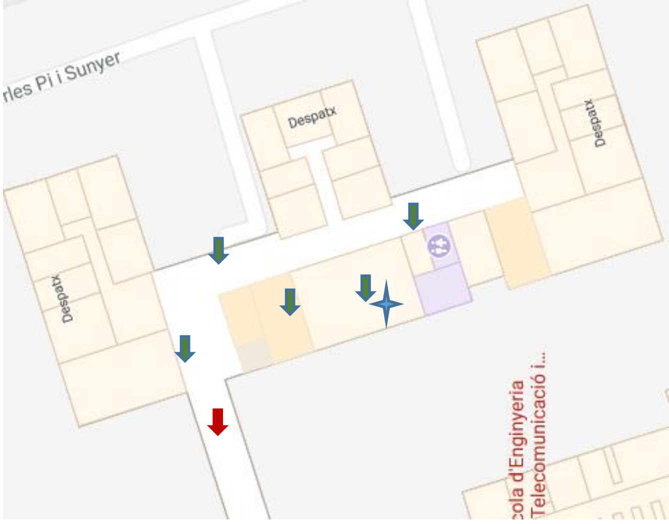
Annex 10 Border router all log output execution

```
* \file
 *      border-router
#include "contiki.h"
#include "contiki-lib.h"
#include "contiki-net.h"
#include "net/ip/uip.h"
#include "net/ipv6/uip-ds6.h"
#include "net/rpl/rpl.h"
#include "net/rpl/rpl-private.h"
#if RPL_WITH_NON_STORING
#include "net/rpl/rpl-ns.h"
#endif /* RPL_WITH_NON_STORING */
#include "net/netstack.h"
#include "dev/button-sensor.h"
#include "dev/slip.h"

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

#define DEBUG DEBUG_NONE
#include "net/ip/uip-debug.h"

static uip_ipaddr_t prefix;
static uint8_t prefix_set;

PROCESS(border_router_process, "Border router process");
```

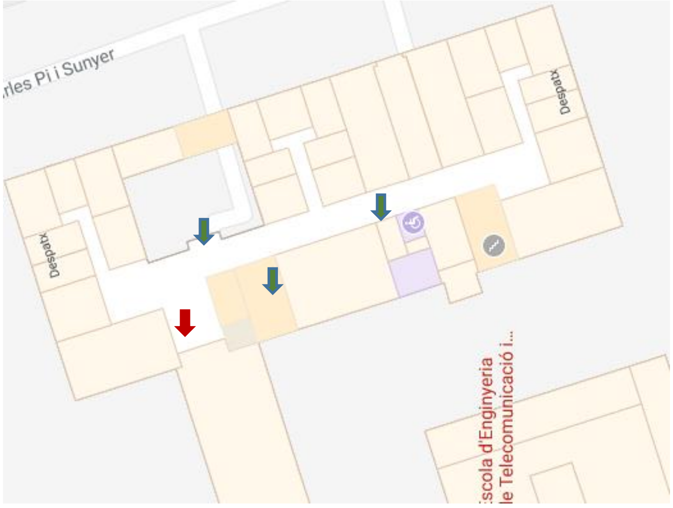
Annex 11 Border router script

```
import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import os

A18 = pd.read_excel("./reproyectoinvernadero/Anual 2018.xlsx")

def miss_value(data):

   # Number of missing in each column

   tr_missing = pd.DataFrame(data.isnull().sum()).rename(columns = {0: 'total'})


   # Create a percentage missing

   tr_missing['percent'] = tr_missing['total'] / len(data)



   return tr_missing.sort_values('percent', ascending = False).head(10)

def eda(data):

   bashline = 10*'-'

   # print(bashline,"Top-5- Record",bashline)

   # print(data.head(5))

   print(bashline,"Information",bashline)

   print(data.info())

   print(bashline,"Data Types",bashline)

   print(data.dtypes)

   print(bashline,"Missing value",bashline)

   print(data.isnull().sum())

   print(bashline,"Null value",bashline)

   print(data.isna().sum())

   print(bashline,"Shape of Data",bashline)

   print(data.shape)
```

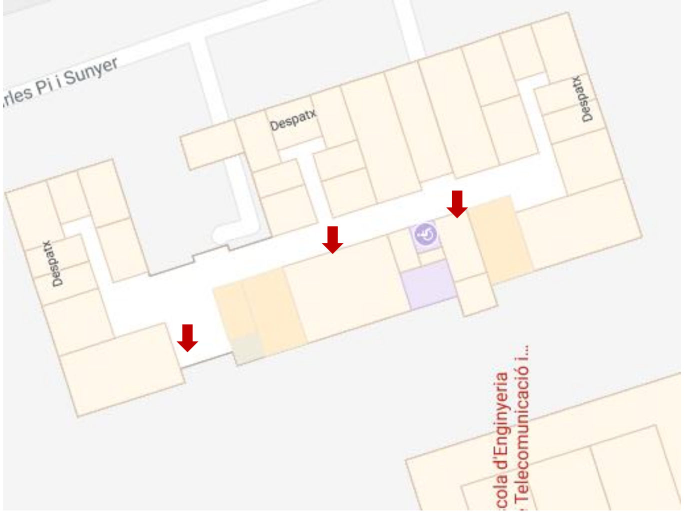
Annex 12. Correlation matrix script using data from Agropolis



Node location, first floor



Node location, second floor



Node location, third floor

Annex 13. Node location in the EETAC Building four

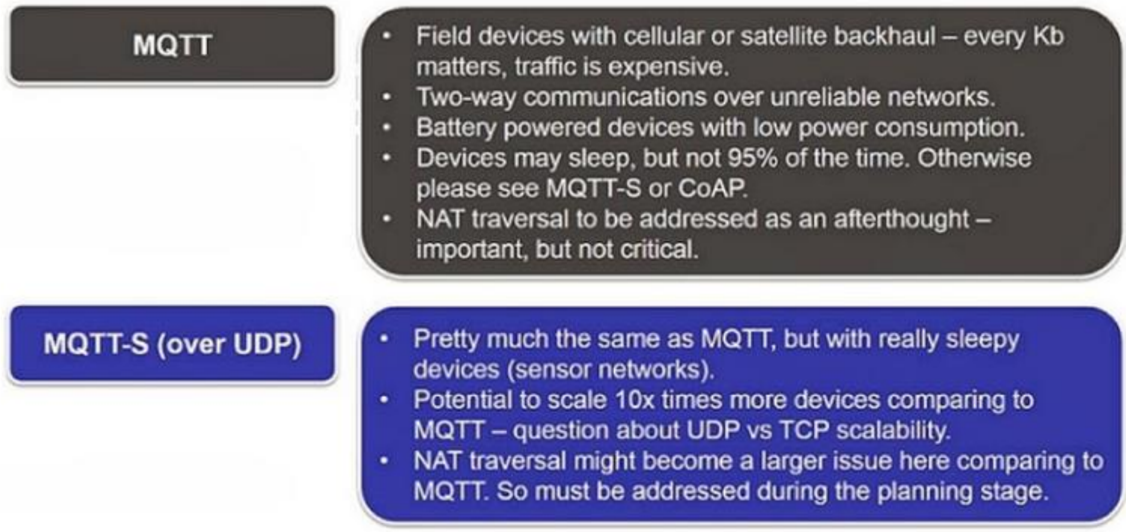
$$e^o(T) = 0.6108\exp\left[\frac{17.27T}{T+237.3}\right] \quad \textbf{(Eq. 11)}$$

T °C	e_s kPa	T °C	$e^o(T)$ kPa	T °C	$e^o(T)$ kPa	T °C	e_s kPa
1.0	0.657	13.0	1.498	25.0	3.168	37.0	6.275
1.5	0.681	13.5	1.547	25.5	3.263	37.5	6.448
2.0	0.706	14.0	1.599	26.0	3.361	38.0	6.625
2.5	0.731	14.5	1.651	26.5	3.462	38.5	6.806
3.0	0.758	15.0	1.705	27.0	3.565	39.0	6.991
3.5	0.785	15.5	1.761	27.5	3.671	39.5	7.181
4.0	0.813	16.0	1.818	28.0	3.780	40.0	7.376
4.5	0.842	16.5	1.877	28.5	3.891	40.5	7.574
5.0	0.872	17.0	1.938	29.0	4.006	41.0	7.778
5.5	0.903	17.5	2.000	29.5	4.123	41.5	7.986
6.0	0.935	18.0	2.064	30.0	4.243	42.0	8.199
6.5	0.968	18.5	2.130	30.5	4.366	42.5	8.417
7.0	1.002	19.0	2.197	31.0	4.493	43.0	8.640
7.5	1.037	19.5	2.267	31.5	4.622	43.5	8.867
8.0	1.073	20.0	2.338	32.0	4.755	44.0	9.101
8.5	1.110	20.5	2.412	32.5	4.891	44.5	9.339
9.0	1.148	21.0	2.487	33.0	5.030	45.0	9.582
9.5	1.187	21.5	2.564	33.5	5.173	45.5	9.832
10.0	1.228	22.0	2.644	34.0	5.319	46.0	10.086
10.5	1.270	22.5	2.726	34.5	5.469	46.5	10.347
11.0	1.313	23.0	2.809	35.0	5.623	47.0	10.613
11.5	1.357	23.5	2.896	35.5	5.780	47.5	10.885
12.0	1.403	24.0	2.984	36.0	5.941	48.0	11.163
12.5	1.449	24.5	3.075	36.5	6.106	48.5	11.447

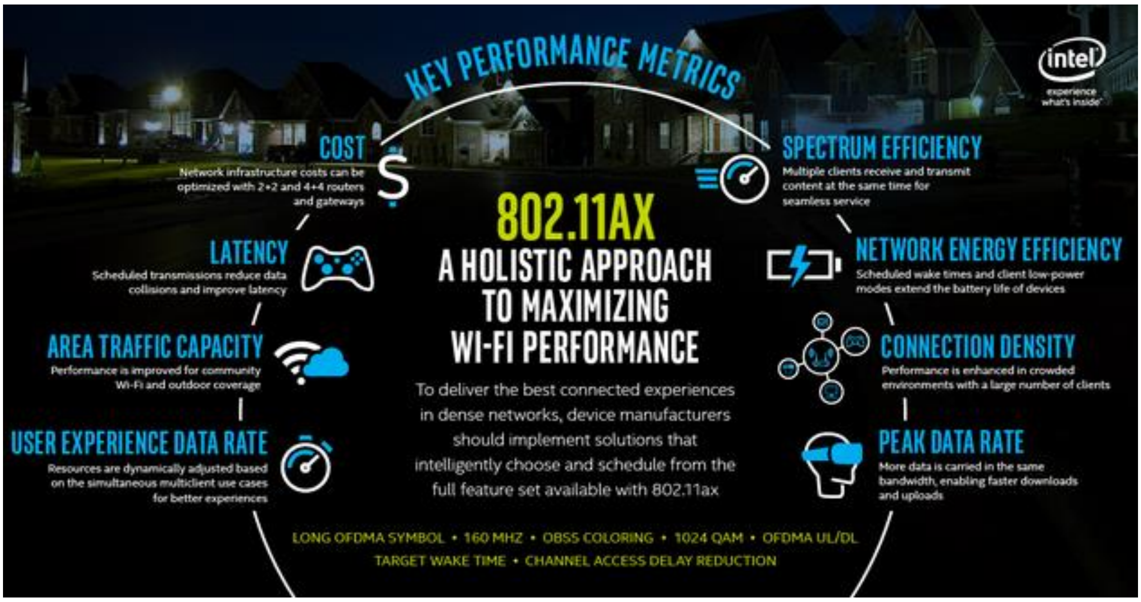
$$\Delta = \frac{4098\left[0.6108\exp\left(\frac{17.27T}{T+237.3}\right)\right]}{(T+237.3)^2} \quad \textbf{(Eq. 13)}$$

T °C	D kPa/°C	T °C	D kPa/°C	T °C	D kPa/°C	T °C	D kPa/°C
1.0	0.047	13.0	0.098	25.0	0.189	37.0	0.342
1.5	0.049	13.5	0.101	25.5	0.194	37.5	0.350
2.0	0.050	14.0	0.104	26.0	0.199	38.0	0.358
2.5	0.052	14.5	0.107	26.5	0.204	38.5	0.367
3.0	0.054	15.0	0.110	27.0	0.209	39.0	0.375
3.5	0.055	15.5	0.113	27.5	0.215	39.5	0.384
4.0	0.057	16.0	0.116	28.0	0.220	40.0	0.393
4.5	0.059	16.5	0.119	28.5	0.226	40.5	0.402
5.0	0.061	17.0	0.123	29.0	0.231	41.0	0.412
5.5	0.063	17.5	0.126	29.5	0.237	41.5	0.421
6.0	0.065	18.0	0.130	30.0	0.243	42.0	0.431
6.5	0.067	18.5	0.133	30.5	0.249	42.5	0.441
7.0	0.069	19.0	0.137	31.0	0.256	43.0	0.451
7.5	0.071	19.5	0.141	31.5	0.262	43.5	0.461
8.0	0.073	20.0	0.145	32.0	0.269	44.0	0.471
8.5	0.075	20.5	0.149	32.5	0.275	44.5	0.482
9.0	0.078	21.0	0.153	33.0	0.282	45.0	0.493
9.5	0.080	21.5	0.157	33.5	0.289	45.5	0.504
10.0	0.082	22.0	0.161	34.0	0.296	46.0	0.515
10.5	0.085	22.5	0.165	34.5	0.303	46.5	0.526
11.0	0.087	23.0	0.170	35.0	0.311	47.0	0.538
11.5	0.090	23.5	0.174	35.5	0.318	47.5	0.550
12.0	0.092	24.0	0.179	36.0	0.326	48.0	0.562
12.5	0.095	24.5	0.184	36.5	0.334	48.5	0.574

Annex 14. Saturation vapor pressure for different temperatures



Annex 15 MQTT-S/SN features[56]



Annex 16 Perfomance metrics for the 802.11AX[16]