Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN

Elettronica, telecomunicazioni e ingegneria dell'informazione

Ciclo 31

Settore Concorsuale: 09/E3 - ELETTRONICA

Settore Scientifico Disciplinare: ING-INF/01 - ELETTRONICA

TITOLO TESI

Bluetooth Low Energy based proximity detection and localization in smart communities

Presentata da: **Davide Giovanelli**

Coordinatore Dottorato

Prof. Alessandra Costanzo

Supervisore

Dott. Elisabetta Farella

Esame finale anno 2019

ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA

Abstract

ICT Center
Fondazione Bruno Kessler

Doctor of Philosophy

Bluetooth Low Energy based proximity detection and localization in smart communities

by Davide GIOVANELLI

Internet of things will bring connected devices to a new level of pervasiveness, where any tangible thing of our daily life may embed some electronics. From a sophisticated smartwatch that embeds complex sensing and communication technologies, to the use of a basic electronic component to implement a digital signature, such as RFIDs. All these smart things worn or distributed around us enables multiple functionalities, when they can interact with each other. In this thesis, I describe the design, characterization and validation of a monitoring system based on Internet of Things technologies, for managing groups moving together in a city. Communication and energy efficiency aspects are firstly explored, to identify Bluetooth Low Energy as a promising protocol enabling scalable and energy efficient networks of things. In the thesis, the protocol has been stressed to demonstrate trade-offs between throughput, energy efficiency, scalability and the possibility to perform multi-hop communication. The potential of the protocol has been exploited within the framework of the CLIMB project. Here, the application requirements and constraints fostered the use of Bluetooth for localization and proximity detection, leading to the investigation of novel strategies to improve accuracy without affecting power consumption and ease of use.

Contents

List of Figures

List of Tables

Chapter 1

Technologies for group management in smart communities

1.1 Introduction

In the context of smart cities, an interesting subject of personalized services are groups, for example families, tourists on organized trips, school groups. For this reason, there is a broad branch of research related to the use of identification and tracking technologies to manage groups. Many wireless sensor network applications require the management of group memberships, not only for the pure purpose of organizing the network (node addressing, routing algorithms), but also because the membership itself is the data the application cares about.
I consider the group membership monitoring in general as the problem of managing instances of any kind (also referred as group members) that are grouped following a given rule and that should be managed together by a group manager or by the group members their selves. An example of this are walking buses, where one or more adults are in charge of bringing a bunch of children to school. They follow a predefined path, which is considered safe and convenient, and they walk together joining new children when they meet at the predefined stops. In this example the instances are the children while the group managers are the adults.
This is just a use case that will be further analyzed later, but it is representative of many other applications where we want to know the exact list of (collaborative) attendees. We refer to collaborative attendees because for the rest of the thesis we discuss about technologies that require some identifying tag on the user. When the user is collaborative, the tag will be correctly managed and then it will work properly, instead if the user is not collaborative (i.e. he does not want to be identified or tracked) he can easily sabotage the system by just acting on its tag.

Groups are entity that are very common in the communities and in the IoT world: examples are people with a common interest or a common aim (students, athletes, colleagues), animals to be monitored (herd, bird flock), means of transportation (shared bikes, buses), but also goods in a supply chain or in a storehouse.
In the today's digital domain, the membership of a group can be obtained by employing any technology that provides identification using wireless communication (i.e. radio identification). These technologies require to equip the instance to be identified with a kind of tag. It can be battery-less such as passive RFIDs: given their short range (at most in the order of meters) they typically require explicit action by the user (i.e. passing the RFID close to the reader to be detected). Less interaction is required if active technologies are used since the range is increased by the means of active circuitry on the tag that is typically powered by a small battery.

Common examples of nowadays low power radio standards are: active RFID, Bluetooth, Wi-Fi low power, 802.15.4 (or Zigbee), also long-range protocols such as LoRa or Sigfox[1], each of them can be considered viable solution, the choice depends on the application requirements.

When managing groups, another key information that is often useful to couple with the membership list is the position of the members. For many applications, such as the walking bus mentioned earlier, it is important not only to know that a member is in the proximity of the manager, but also to have a measure of how far it is, so that an alarm can be triggered once the member is going too far. In particular, most of the time when dealing with (moving) groups, what really matters is the relative distance or the relative position of the members with respect to the manager rather than the absolute position (i.e. latitude-longitude coordinates). Even in applications employing static nodes, like environmental monitoring, having the position of sensors is important, both as data source and for maintenance purposes. In fact, if a sensor network collects data about temperature, humidity and atmospheric pressure for whether forecast, it is mandatory also to know the position (absolute position in this case) where the samples have been acquired, otherwise they are useless for the purpose of the forecasts. Regarding static nodes, positions may be assigned manually during the network setup. This is reasonable in low-to-mid sized networks, but with larger ones it may become a cumbersome operation. Moreover, by monitoring the position of nodes in static networks, the manager can understand if there are failures or sabotages and geo-reference them.
Nowadays, global navigation satellite systems (such as GPS or Galileo) can provide absolute localization on most of the earth surface; however, it is not always a viable solution because of three main reasons:

- **Accuracy:** typical accuracy of satellite-based positioning system ranges from less than one to 30 meters [19], which is fine for many applications, but for those relying on the relative distance, errors accumulate, and they might be no more acceptable. There are ways for reducing the error to the centimeter level [14], however cost and energy consumption grows; and in large network deployments the single node cost plays a fundamental role for obvious reasons.
- **Coverage:** the satellites provide a good coverage of the earth surface, however when the open sky is not visible (i.e. inside natural or urban canyons, or indoor) satellites' signals may be too weak or inaccurate to provide good position estimation.
- **Consumption:** typical consumption of commercial GPS receiver is in the order of tens of mW, which can be considered low power, but not ultra-low power that is the target for IoT sensor nodes.

For these reasons it worths studying alternative ways to obtain node location, and one of this is by exploiting radio signals that are already available for enabling wireless communication.

1.2 Group Monitoring

In general, what it is expected from a group monitoring system is to handle everything that concerns the list of members and the logs of events regarding the group.

[1] LoRa and Sigfox are two protocols optimized for transmitting small chucks of informations (in the order of few tens of Bytes) over long distances (> 1 km) while remaining relatively low power.

More specifically, such a system should automatically detect and identify the members of the group that are in the proximity of the manager. By using the distance estimation (or localization) it can understand the actual proximity between the manager and the members, and if some of them is going out of communication range, the system should trigger an alert informing both the loner member and the manager that they are probably too far away for being part of the same group. The system should also provide a periodic journal (daily, monthly, one-off, depending on the application) reporting timestamped events such as:

- **Member in range:** a member is detected in the proximity of the group, but it is still not part of it.
- **Member join:** a new member formally joined the group.
- **Member leave:** a member left the group in a controlled way.
- **Alert:** a member is going too far away, or a member is no more detected (and it never formally left the group).
- **Member location:** relative or absolute coordinates of the member.

Having a log of what happened permits an off-line analysis that can be many fold. It may be used for maintenance (battery change/recharge, software updates), for safety, insurance or statistics (the periodic fee for the insurance or for the system concession may be dependent on the use).

Depending on the application, there may be the need of monitoring the group from within it (*local monitoring*) or from the outside (*remote monitoring*).
Local monitoring can be effectively used for instance when managing group of children on the street: given the reactivity required to intervene in case of danger, the application should have low latency in detecting dangerous situations. However, it is clear that a group manager should pay its attention to the children (that must be in line of sight) rather than focusing on the smartphone. Then the proposed system is intended to be as support of and not in place of the physical person, it can tell if somebody is missing with respect to the expected list of participants, but it cannot stop a (nowadays) car if a child is walking in a dangerous zone[2].
Remote monitoring instead can be employed when the application is more latency tolerant: remaining in the context of children, remote monitoring may be applied at school. When all are in, the information from all the classrooms can be collected in a central point, where a person or a machine can check the missing ones and eventually informs the parents.
This difference between local and remote monitoring is important to the purpose of stack optimization. However, the same system can integrate both and use the proper one depending on the specific application needs.
Given the nowadays' level of penetration of technologies like smartphone and internet, it is strongly advisable to reuse technologies where possible, otherwise people will hardly accept solutions that require application specific devices in place of their own smartphones: this paradigm is called bring-your-own-device [42]. For this reason one of the key features for such a system is to be well integrated into nowadays technological ecosystem.

[2] If we consider autonomous cars (that are only at prototype stage at the time of writing) their sensing system may integrate the technology to detect people using radio frequency identification if available on the person.

1.2.1 Stack description

Regardless the technology chosen for the implementation, the requirements of group management with position awareness cover the full stack: from the physical layer, which is important to analyze for the purpose of localization, to the cloud services that can be used to synchronize expected attendees list (i.e. the list of service subscribers). It is worth highlighting that the data security in IoT applications employing people identification is a crucial point, then it should be carefully addressed at all levels.

The **lower layers** are in charge of handling radio signals to implement wireless communication and perform measurements to obtain location information. The communication channel will be used, at least, to send identity information, which is typically encoded in a numeric code called ID. Another kind of data that may be transmitted is sensor data or other high-level information such as position data. Since the target is the IoT, the energy efficiency of the devices is one of the top priority. To accomplish this, low power radio protocols employ aggressive duty cycling technique to turn off the radio peripheral when not in use. In fact, it is known that the radio is one of the most energy hungry component of a wireless sensor node [78], then optimizing the radio operations is the first step toward the energy efficiency of the device. This optimization is typically done at low level to increase the accuracy of the behaviour and the performance of the code.

Mid layers manage the overall system logic: for instance, the device may use sensors to understand the context it is and perform actions in response of a context. In other words, if the device is steady it may lower the position report rate, saving power. Another task of the mid-levels is to manage and store device configurations, providing tools for an easy, convenient and possibly unified interface to access the settings. This seems to be trivial in nowadays systems, but in the view of IoT we will have really huge amount of sensors, if each family of them has a proprietary tool, it may become hard to set up and maintain deployments.

High layers are mostly focused to the application specific behaviours. They handle cloud database storage to allow a permanent and easily accessible information and event logging. If needed by the application, upon authorization they provide the mapping between the ID number and the actual identity (name, family name). High levels also include the visualization tools like web-based consoles or smartphone applications and some part of reasoning that is use case dependent. For instance, in the walking bus example, a message may be sent to child's parents informing them that their son arrived at school. This kind of operations are managed by the higher layers because they are strictly application dependent.

1.2.2 Applications of group monitoring

Some of the possible applications of group monitoring are analyzed here, with particular attention to those that can be in the ecosystem of the smart city and communities. Moreover, the discussion is restricted to group monitoring applied to living beings (humans or animals) since monitoring of things have different needs. Application specific requirements and challenges are discussed and solutions to main problems are briefly described.

Children monitoring

We already introduced the walking bus application, but it can be generalized as the monitoring of children that move outdoor in one or more organized groups, with

various level of autonomy. Monitoring school trip, recess break in the school's courtyard or outdoor sport (ski school, running, outdoor workouts) belong to the same class of application since all of them have a group of children being monitored by one or more adults that are in charge for their safety.
In this context, it worths highlighting that, in the last decades, childhood in urban areas has changed and the technology can be used to re-establish "the spatial practices of children" [112, 97]. At the same time, there is a scarce adoption of technologies that permit precise position tracking, which are available on the market since last decade (Weenect Kids GPS Child Tracker, Ubisafe, AngelSense Kids GPS Tracker are only few examples). As pointed out by [111], these devices are mainly adopted to relieve the short-term personal anxieties without a real look at the long-term consequences on children's sense of autonomy and independence. On the other hand, non-adopters prefer less intrusive practices, such as curfew or activity boundaries. For this reason, the user interaction with such systems must be carefully addressed to avoid the technology to absorb all the user attention. This consideration regards both the children and the group manager (i.e. the teacher, the parent or the adult in charge of managing the group of children).
From the technical point of view, this class of applications requires a continuous monitoring of the presence of all the children. However, as found in [111], a real time tracking of child position might be counterproductive. Therefore, only important notification should be given to the group manager: for instance, as already proposed, an alert can be risen when someone is moving too far for being effectively supervised by the adults. Since in this kind of monitoring the readiness of the group manager (typically a teacher or a relative) to react is of primary importance, the monitoring should be done from within the group. Therefore, the most convenient way to interact with the system is through a device that everybody has in the pocket and can easily being used, like a standard smartphone. Providing smartphones to all the children such as proposed in [84] might not be acceptable for ethical reasons, then we want a technology that is transparent to the child, requiring no direct interaction with the tag. At the same time, adults' smartphones can have a low latency connection with the children's tags that provide the identification and tracking, and in the meanwhile the smartphone can be connected to the internet to download other kinds of information used by the application.
In the case of school-like applications it is important to compare the actual list of group members with the expected one to obtain the list of absents by difference.
A further yet controversial feature is the communication with the parents. If a child is supposed to be part of a walking bus, but it is not detected, a message could be sent to the parents. Such a message may be alarming for them, therefore a double check is recommended to avoid false alerts due to empty batteries or a tag that has been forgotten at home.

Museum or guided tours

Another quite popular application of group monitoring is related to the guided tours both in museum or outdoor areas. In [29] the group monitoring concept is used in the context of guided tours in museums where RFIDs and Wireless Sensor Network are employed as technology enabler. Authors highlight that the group is composed of peers, and, contrary to *Children monitoring* applications, the guide is the group manager, which is not intended to be a supervisor. She/he will be just a guide, and therefore, the members may be free to step away from the main group for some time if they are interested in something the other members do not care about. Localization

is used to inform the member about the position of the main group in case she/he get lost. Another important requirement is the coexistence of different group in the same area.
The guide becomes an autonomous robot in [23]; in this case a key role is played by the localization techniques (based on a mixture of tactile, infrared, sonar, laser sensors), which is mandatory for ensuring the autonomy of the robot. A simpler ultrasound based system that runs on users' smartphones is presented in [17] where the focus is the use of personal devices (i.e. smartphones) as electronic guide in order to reduce the cost for ad-hoc solutions typically employed in museums. The outdoor scenario is considered in [97], where gamification techniques are applied to digitally augment field trip. The game's hardware architecture is composed by several different kind of devices (pingers) that are carried by children. This application might be also included in the *Children monitoring* class of applications. From the point of view of this thesis, the more relevant pingers are those used for location and/or context detection: in fact, for absolute localization the information is obtained through the *GPS Pinger* complemented with *Dead Reckoning Pinger*, while for relative localization (for detecting the proximity to a point of interest) short range radios (Location Pinger) are preferred. This confirms that the GPS is not the final solution for localization, in particular when the goal is to detect relative proximity rather than the absolute position.

Therefore, in this class of applications, the system with the localization feature can be used for two main purposes: provide a digital guide with description and suggestion based on the position (absolute localization is then a requirement), keep the group membership coupled with members' location to allow sharing this information with the other members and/or the human guide. To these purposes having a basic tag for identification and localization is not enough; instead, the system can be implemented with the users' smartphones in a way that every member can install an application, which will act as interface with the system and, at the same time, the smartphone's resources are used for the identification and localization.
Moreover, also the museum managers can benefit from such a system, in fact they can have a remote console from where they monitor the number of presences both for statistical and safety reasons. For this purpose, all the museum guests have to be forced to carry a compatible tag or the smartphone with the application running.

Sports competition

For this use case, we distinguish between two kinds of sport competitions: sports played in specific fields (like basketball or rugby) and sports played outdoor (like running or cycling). An example of the former is provided in [53] and in [52] where the WASP system is described: a large radio bandwidth (125 MHz) in the 5 GHz ISM band is used to locate players with respect to a number of reference nodes (anchors) and with a high sample rate (up to 200 Hz). The architecture requires ad-hoc hardware that is build using commercial low-cost components. Using this system, athletes can be tracked in the play field with respect to the anchors; however, there is no link nor range estimation between athletes (only between athletes and anchors), then the localization can be done only in confined environment (indoor/outdoor fields) and relative group localization is not considered here.
Instead, for those sports where the key information is the position of the athlete with respect to the others, rather than the position with respect to a play field or to a fixed infrastructure, cooperative localization [92] is a more adequate technique. In cooperative localization, nodes perform range estimation not only to the anchors, but also

to the peer nodes. In the absence of fixed anchors, mobile nodes can still localize themselves with respect to the others. This can be very effective in competitions like cycling or marathons, since many athletes (hundreds) might be moving in group. In this situation, the absolute location of the single participant is not relevant, however, the relative position inside the group can be an important information for the strategy.
In this use case, the group members (i.e. athletes) do not like to carry extra weight; then the devices need to be small and they should work without the need of user interaction since the monitoring is mainly remote: this means that the system should be as transparent as possible for the athletes. Their assistant or trainers will communicate only relevant information using other channels (just voice or using radios if they already in place). If this is applied to non-professionals athletes, without a personal training team, the same information can be accessed directly by athletes if they accept to carry a smartphone like device.

Crowd monitoring

In large public events such as concerts or religious events being able to monitor the crowd can be highly valuable for safety reasons. It is not so unusual that large crowds go out of control creating panic, then wounded and even deaths. Of course, a monitoring system will not directly protect event participants, but it may help the organizers to manage the crowd avoiding creating dangerous situations.
In [13] it is highlighted how passive sensing techniques such as those based on cameras can detect crowd flow, can count people and estimate the crowd density; however, they fail when accurate mobility patterns need to be analyzed on the long scale. The author indeed identified the Bluetooth Low Energy (BLE) to be a viable technology to the purpose; in fact, it is cheap, low power and pervasive. In the proposed experiment, where the aim was to detect people mobility, we could say that "the crowd is sensed by the crowd". In fact, both the BLE beacons and the BLE scanners (smartphones equipped with an application that scans for BLE devices) are carried by participants. Contacts between the beacons and the scanners are pushed to an online server together with the GPS location of the contact. Even in such collaborative scenario the privacy can be ensured if no association between beacon ID and the person who is carrying it is made during the deployment, the same consideration applies to the smartphone app.
It is also important to note that, in this class of applications, it is impractical to tag hundreds of thousands of users: tagging only a relevant subset of them is a choice, using people's own devices to this purpose is another choice (in [13] a mixture of the two is employed), complementing it with other complementary technique (camera based) is another choice.

Smart City

Being cities and communities made of groups of people, the concepts of group management and localization find several possible applications in the context of Smart Cities and Communities. Smart Cities experts envision applications that are described as: smart transportation, smart grid, smart health, smart waste collection, authors in [90] define it *Smart World*. To make a city (or the world) *smart*, a set of components are required. Probably the three most important are: a network to allow information communication between entities, a processing power to process the available information, and the data sources that generate the information. As

pointed out by the authors, location and time are the most relevant underlying features of any observation (after the observation itself), and since Smart City applications will be fed with data coming from observations, localization in Smart City must be more pervasive and accurate. In fact, for what concerns the time, technology already permits to timestamp the data with a very high accuracy and resolution, while for what concerns location this is not always realized.

Wildlife monitoring

Not only groups of humans can be monitored, also animals can, in fact one key tool for zoologist and biologists is to monitor animals' movements and interaction in their daily routines. For instance, authors in [93] applied a technique similar to the one described in Section 1.2.2 but using much more constrained devices. In fact, they use a low power contact detection technique (also referred as neighbour discovery) applied to 802.15.4 radios to detect proximity between mid-sized animals (deer, fox). Any contact between animals activates a GPS receiver, coordinates are stored to a memory and the georeferenced contacts are then off-loaded using either fixed nodes in the wood or alternatively, in the absence of the previous, using an on-board modem. Energy consumption is also studied, and a model based on a weighted average is used to help biologist choosing the proper set of parameters to reach the most appropriate trade-off between accuracy/resolution and consumption. Much smaller animals are instead considered in [79] where a miniaturized BLE based device targets the collection of small birds' vocalizations. In this case the main challenge is to compress weight and size to avoid stresses to the animal and hence a change in the behaviour, which is obviously unwanted. To this purpose minimal components are allowed on the device; however, it can still acquire audio from a contact microphone and also, temperature. Even if not explicitly described in the paper, the presence of the short range BLE radio allow to geo-reference acquired data. If these techniques are applied for studying the animals, remote monitoring with offline data analysis may be enough; instead for other animal related application the requirements may be the opposite. When dangerous animals (bears, wolfs) reach villages they can be harmful, by having a system that detects them an automatic bollard may discourage their stay or alternatively a communication may be sent to forest rangers. This is what has been done in the BearFence project [89], where contact detection of bears (based on collar mounted RFID) has been used to localize the animal in a protected area.
It must be noted that an animal cannot be considered a collaborative user and, to detect it, the tag must be deployed on the animal itself. This procedure may be challenging and if only a fraction of the herd is tagged, the effectiveness of the system drops as in the crowd monitoring (Section 1.2.2) class of applications.

Some of these applications are already addressed by specific technologies, for instance in the context of sport, position trackers based on GNSS and inertial sensors are already in use. For the purpose of crowd monitoring phone cells data could be analyzed, and some specific wildlife trackers are used by biologist. However, it is very difficult to find a technology with enough flexibility to fulfill the requirements of more than one application.

1.3 Requirements and challenges

Using radio signals to identify and localize things or people is not a new concept. The radar is one of the first devices exploiting the radio signal to localize object by listening to the signal echo reflected by the object itself. However, a technology that can fulfill all the requirements of the previously mentioned use cases have not still emerged. There are many technologies nowadays fulfilling very well a specific set of application requirements; however, a general yet flexible and convenient solution for all cases still does not exists.
Therefore, to choose the technology that better fits the application, it is mandatory to define a set of design parameters, on which the application requirements will be based. The most relevant are:

- Energy efficiency: a typical requirement of an identification device (i.e. the tag) is to guarantee a long battery life in the order of years with reasonably small battery. A bracelet-like device, for example, to be wearable and not cumbersome needs a small-size battery such as those available today, e.g. a 15x10x2.7mm^3 battery with a capacity of 25 mAh. However, to guarantee that such kind of battery lasts 1 or 2 years, the average consumption should be in the μW order [3]. Properly configured Bluetooth Low Energy devices are not that far from these values [44]. BLE Manufacturers often rate their beacons to last more than one year with a small coin cell battery.

- Range: communication range depends on many factors: transmission power, modulation scheme, antennas gain, receiver sensitivity, carrier frequency and bandwidth, environment and other. For most of the mentioned applications, the required range does not overcome 100 m. Since the data is used locally, a longer communication range is not mandatory. A longer communication range might look always welcome; however, this should be traded-off with power consumption and with scalability. In fact, the easiest way to increase the range is to increase the transmission power, however this leads to higher energy consumption; at the same time a longer communication range will reduce the spatial reuse of spectrum, since communication collisions can happen on a wider area effectively reducing the maximum supported node density. Furthermore, the network can be equipped with a gateway connected to internet; this, with the help of cloud services and protocols can make the data available anywhere there is an internet connection.

- Localization accuracy: radio based localization is a well studied topic in particular in the wireless sensor network community. Many signal processing techniques have been proposed [113, 69, 12, 74, 91]. However, what really sets the accuracy is the metric that is used for estimating the distance, which can be for instance, RSSI or time-of-flight.
  The required localization accuracy for almost all applications mentioned in this chapter is in the order of 1 m (sport applications typically require higher accuracy).

- Reliability: if the radio device is supposed to be used in safety critical applications, the reliability of both the acquired data (sensors, location) and the device itself (it must be rugged) plays a fundamental role. For instance, many of the

[3] The maximum accepted consumption that satisfies the wanted battery life (given the battery size) is estimated with: $\frac{\text{capacity}}{\text{target battery life}} = \frac{25[mAh]}{730[days]24[h/day]}3.7[V] \approx 5\mu W$

Bluetooth modules on the market are made with commercial grade components that may fail in critically harsh environments.

- Scalability: if radio identification and tracking is performed on large scale, the technology should be scalable enough to avoid network congestion. It is a common experience to have problems with the cell phone system when there are too many people that use the phone in a small area. In that case a single cell cannot handle all the traffic and some of the user will be put offline. In this regard requirements are application specific; then, an adaptive solution is advisable: when many nodes are in the neighbourhood it should use conservative settings such that the network does not get congested, while when the group is small the settings may be tuned to optimize other parameters (such as latency)

- Latency: because of scalability and energy efficiency aspects, the radio cannot transmit all the time; some form of duty cycle has to be applied. This makes the system active only a fraction of the time and then the detection can suffer of latency.

- Security: security of data is another key aspect of a radio identification beacon. If the personal device broadcasts the user identity all the time in clear, an offender may sniff the data and understand where the user is or where she/he is not violating basic privacy rights.

- Throughput: links speed is one of the few relaxed requirements, in fact to recognize a device only the ID must be transmitted. The ID length can range up to 128 bit (16 bytes) that is the length used in the definition of UUID given in ISO/IEC 9834-8:2005 that is often used when a unique identifier is required for any reasons. If the full ID is transmitted every 100 ms the required throughput is 1.28 $kbps$, which can be considered low throughput (it is less than 1 % of what available on modern wireless communication standards [21]). There are however cases where higher throughput is necessary, in fact if the application requires multihop contacts detection, any node, together with its local information, will also forward the list of its neighbours, and maybe also the list of the neighbours of its neighbours and so on. This increases the required throughput exponentially with the size of the network, imposing severe limits on the system scalability.

1.3.1 Application specific vs generic solution

Given the wide spectrum of possible applications, a question arises: do we need an application specific device for each of them, or a common solution may be employed? Requirements are similar, then the same technology may apply; unfortunately, such technology does not exist yet. An ideal candidate for group management in IoT application should be flexible from the physical layer up to the cloud to fit the widest range of applications.
A viable solution may be to create a hybrid design, with heterogeneous radio interfaces for the different purposes. Anyhow, such hybrid solution might become complex, because different standards (Bluetooth, LoRa, 802.15.4, UWB) will expose proprietary and not compatible interfaces. Moreover, the hybrid solution built with the fusion of multiple standards should manage these standards such that they do not conflict and their use is well defined. In other words, a standard use of standards

should be defined. This seems not to be a viable way because concerting different standards to work together while guaranteeing interoperability and compatibility is not convenient. Then for now, system designers are forced to make application specific solutions.

1.4 Case Study and motivation: The CLIMB project

During my three years of PhD school I have tested some of the technologies that are suitable for group management. I tested them within the context of walking buses organized to bring children at school.
The CLIMB project of Fondazione Bruno Kessler aims at promoting the independent mobility of children inside the neighbourhood . To this purpose IoT technologies are used to monitor their way to school. Parents volunteer to supervise the walking bus, which have a time schedule and predefined stops, and the children, instead of reaching the school on board of the parents' car, they walk to it together. For insurance and safety reasons, adult volunteers of the walking bus are requested to write a daily journal to keep track of all the children presence. Usually, this journal was compiled manually, which is not the most appropriate way if at the same time the volunteer has to supervise some tents of children on the road.
For this reason, at Fondazione Bruno Kessler we designed an IT solution employing Bluetooth tags for recognizing the children and help the volunteers recognizing them through a smartphone application. From the technological prospective, I identified the following high-level functional requirements for the basic group management:

- Continuous discovery of new group members
- Detection of member leave (relative members localization)

while non-functional requirements are:

- Detection latency should be less than 5 s
- Scalability should be guaranteed up to hundreds of nodes (the target will be 150)
- Communication range in the order of hundred meters: it often happens that children form long caravans when walking on the sidewalk, the radio contact to the manager should not be lost in that case.
- Compatibility with personal devices such as smartphones will help the adoption
- Low maintenance efforts: being this system management (i.e. the need of manually turning on/off devices) or pure maintenance (i.e. the battery life should not be shorter than three months).

Others and more specific technical requirements will be given in the next chapters.

The group management system will be also complemented by a game (Kids Go Green, [81]) to improve user engagement, then encourage the adoption and keep the motivation high on the long period. The design, validation and description of this game is out of this thesis's scope.
I took the CLIMB project as an opportunity for experimenting group management

in the IoT and smart city ecosystem, providing basic functionalities to walking bus users and at the same time exploring more advanced functionalities offered by the technology.

1.4.1 Evolution of the walking bus management system

The present section is an overview of the thesis. It describes the evolution that the walking bus management system has undergone during the doctorate.
Bluetooth Low Energy (or BLE) has been chosen as the most appropriate technology because of three main reasons:

- Energy Efficiency: manufacturer declare that in its basic use (beacons) devices can last a couple of years when powered by a small battery with capacity in the order of 200mAh
- Pervasiveness: BLE is now supported by almost all personal devices (smartphones, tablets, pc)
- Communication range: in open field the range of a BLE beacon (with 0 dBm transmission power) overcome 60m, when nodes are carried by children in the backpacks the signal propagation is perturbed, but it is still adequate for the application.

BLE have many interesting features that perfectly fits this sort of applications, however lower layers are not optimized for localization or range estimation. Moreover, latency-scalability trade-offs are not well defined and sometimes device dependent. To the purpose of better characterize and understand the possibilities offered by the BLE devices, several experiments and applied studies that asses the unknown aspect of BLE have been carried on in this thesis.
First assessments have been focused on a basic characterization of throughput and consumption trade-offs during data streaming operations (Chapter 2). Even if the use case is a bit different from the group management task, we used the same analysis technique in several other situations.
After this preliminary study, the first version of the group management system (MIGnOLO - ManagIng Groups with bluetOoth LOw energy) has been developed and it is described in Chapter 3. Localization on version 1 of MIGnOLO is based only on RSSI and literature on this technique suggest that the accuracy is low [54]. The first attempt to cope with the low accuracy is to approach it with *brute force*, which means adding more input data to the problem and let a machine learning algorithm to find out patterns of behaviour (Chapter 4). From this point of view, version 1 of MIGnOLO was able to collect really a large amount of data, anyway it suffered of two interconnected problems:

MIGnOLO (ManagIng Groups with bluetOoth LOw energy)

- Excessive current consumption: to collect the additional data needed for enforcing the localization algorithm, the radio duty cycle had been substantially increased, reducing battery life to 3-4 weeks. This is not acceptable for a system that should be transparent to the children and their parents.
- Localization accuracy brought by the brute force approach did not increase sufficiently to justify the battery life reduction.

The first version of MIGnOLO have then been updated to the version 2 with reduced functionalities (with respect to the version 1), however it required much lower maintenance efforts due to the reduced energy consumption. This version employed the simplest form of BLE device: beacons. They periodically broadcast a small packet of information containing (in our case) a numerical ID. This packet can be received by all neighbouring devices just by activating the BLE scanning and without the need of performing any kind of pairing or synchronization. On the other hand, data transfer is not acknowledged and the available throughput is rather low.
Still, some other open challenges remain in this version of MIGnOLO; in fact, scalability and latency where assessed only experimentally for a reasonably low amount of beacons (less than 30). This is enough for a single line of walking bus; however, when multiple lines merge during their way or when all the lines meet at school the number of beacons easily overcome 100, then a deeper analysis is needed to ensure system stability.
The focus of the work moved then to the study of the performance of more complex networks involving communication over the BLE advertising mode (Chapter 5). For this a BLE Link Layer simulator that aims at detecting packet collisions has been developed, given that colliding packets will be always corrupted at the receiver side. In real world this is not true, in fact, if two packets collide but one has much higher power with respect to the other (this happens when the first node is close to the receiver, while the second is far away), the first one may be properly received while the second remains masked[4].
The same simulator, properly adapted, have been used for assessing some energy efficiency techniques that are suitable for the consumption optimization of the version 1 of MIGnOLO. I will show that the technique is effective in Chapter 5; however, it requires a new hardware component, a BLE compatible wake-up radio [106, 100, 94] that at the time of writing is not available on the market.
The last and largest part of the study was addressed at the improvement of localization techniques to be used on top of BLE. Great effort has been spent to improve RSSI (Received Signal Strength Indicator) ranging and localization. However the drop of performance when passing from ideal testbed to real world deployment was not acceptable. For this reason, the approach has been changed by embedding the time-of-flight as input information into the localization algorithm. Radio signals' time of flight is known to be a good distance estimator. In fact, it is employed in GNSS (Global Navigation Satellites System) and on many of its terrestrial predecessors. Since the time-of-flight is not available on commercial BLE modules, an ad-hoc software library has been developed for that purpose. It adds this functionality to the BLE just by using the resources available in the system-on-chip (SoC). This means that no specialized hardware is required and although the library demonstrate the principle only on one family of BLE chips, similar functionalities can be ported on all BLE SoC. The description and characterization of this alternative ranging technique can be found in Chapter 6.
Time-of-flight alone is not a breakthrough in BLE based localization since the noise level of raw data is quite high (up to 10 meters) to be directly used; however, the overlying noise is more gaussian with zero mean with respect the error found with

[4] Note that if a beacon is far away and its packets are always masked by closer beacons, the receiver will never *discover* the remote one. We are interested in having the whole list of beacons (i.e. children) then if a node is never discovered it means that the system is not reliable, therefore the simulation results remain valid even if they discard the effects of the physical layer.

RSSI. This means that by applying the proper filtering techniques distance estimations can be cleaned out and effectively used for localization and tracking. In Chapter 7, the description of the filtering and tracking technique for indoor anchor-based applications is provided. With minor changes this technique can be extended to perform anchor free relative localization.
The final version, i.e. version 3, of the MIGnOLO system will then use time-of-flight measurements to estimate relative members distance. The algorithm that performs the filtering will be embedded in the tag nodes while the group shape will be computed in the adult's smartphone.
At the time of writing, version 3 of MIGnOLO is still in progress, therefore I will present only ideas and proposals for the future work.

1.4.2 Premise and contextualization

The present thesis has been carried out in a constrained context. The CLIMB project started as a relatively confined experimentation (one school, twenty parents, forty children), and in a couple of years more schools joined bringing the number of testers (children and parents) to more than 600 people. It must be clear that the development of smartphone applications, cloud services, the game, and all the components not mentioned in this thesis has been a work of the Smart Community team of FBK, who worked with me and other colleagues in order to embed the proposed group management solution into the CLIMB smartphone application.
Practical limits in such context are countless:

- At the beginning of the project (2015), only a fraction of our users' smartphones where equipped with BLE connectivity since Bluetooth was supported only in its previous versions. Even where present, the software support was poor and device dependent, making it very hard for the developers to create a stable and usable application.
- A variety of smartphones platform (Android, iOs, Windows Phone) further complicates the development.
- The environment of the experimentation are mainly the roads of small villages in the neighbours of Trento. In such situation very small attention is given to the experimentation itself since the volunteers, obviously, pay more attention to children rather than the application running in the smartphone, making it difficult to get feedback about encountered problems.
- For technical and privacy issues, it is almost impossible to retrieve a ground truth to be compared with system's observations.
- Deploying a new feature on the BLE devices requires, in most of the cases, to flash a new firmware on all the devices. This means retrieving all of them, flash one by one and return them to the testers. We found this to be really impractical, then we gave more importance to the stability and the absence of maintenance of the solution, rather than improving the system with more functionalitie. Therefore, some of the methods described in the thesis are tested on smaller testbeds that are easier to control).
- Given the high number of BLE devices that can be found in a typical scenario (hundreds), ensuring the stability it is not easy. For instance, performing a scalability test in a real scenario would require all the hundreds of devices

to be together, this is not easy because of cost and feasibility (any parameter change requires all the devices to be reprogrammed).

For these reasons not all the techniques proposed in this thesis are actually in use in the walking bus. Nevertheless, reported results are based on real experiments executed on real hardware and where possible the test setup tries to mimic what can be found in the walking buses. Simulations are only used to asses Link Layer behaviour in crowded environments (hardly feasible otherwise). The custom simulator developed is firstly validated in Chapter 3.5.2, and then it is refined and expanded in Chapter 5.

Note about the *IoT* acronym: the application scenario of this thesis is the nowadays ecosystem of heterogeneous devices with communication capability and some form of processing power (this often make them be defined *smart things*, even though the *smartness* of something should be weighted considering how the processing power is used). In literature this is often referred with Internet of Things or IoT.
During the PhD I head this acronym more times than I can contemplate, and I found that it is used in at least two different scopes.
For people working at network or cloud level, the most important part of *IoT* is the *Internet*, and for them IoT refers to the set of tools, rules, protocols and technologies that allow the communication of any kind of data between heterogeneous devices and services, how the data is produced is not of primary importance for the developer.
Instead, for those that work on hardware, or low-level communication (typically wireless) the most important part of *IoT* are the *Things*, and how the data is transferred to the end user is not a key point.
It is clear that, in the IoT context, the internet without the things (and vice versa) will be of scarce use, then there is no reason to consider one more important than the other.
In the present thesis the focus is the second scope, the *Things*, and when mentioned, *IoT* will be considered from the *Things* point of view.

Chapter 2

Bluetooth Low Energy Analysis

2.1 Introduction

Bluetooth and in particular the Low Energy version emerged in the last years as enabler for many applications. Given its range (from few meters to 100 in optimal conditions) and its energy efficient controller, it has found many applications in particular in the context of wearables. Such devices at least employ some sensors, a radio interface and an MCU that is used both for the management of the overall system (reading sensors, managing radio interface) and for preprocess data to be sent through the radio.
The most frequent scenario is the use of wearable Body Sensor Networks (BSNs) for the monitoring and analysis of selected features of the human body. Dedicated signal processing techniques extract high level information from sensor data, both in real time and offline, providing useful insights of the user's state and performance. A common BSN architecture is formed by multiple sensing nodes and one central unit, acting as data collector, processing hub and gateway towards further systems. The sensing nodes usually have a very limited dimension to be comfortably worn on the point of interest, hence they are carefully designed for wearability embedding the very needed sensing and communication technologies [24].
While in the past years there was research and development of custom wearable central units, now smartphones are the standard option in most of cases. This is supported by several reasons: their cheap availability and the possibility for the development of custom applications, the ease of use and the robustness of such solutions and their ever growing computational and communication capabilities [7].
Radio interfaces and communication protocols have always been a crucial part of BSNs. Their choice determines the nature of the data that the network will be able to exchange, setting the BSN specifications in term of supported number of nodes and data throughput or latency [26]. Every application has its requirements in terms of throughput, latency, number of nodes, then the flexibility of the protocol is crucial to tune it to the specific application. Moreover, the communication subsystem is usually the most power-consuming one, hence the need for its optimization and energy efficient power management [15, 77].
Since the beginning of the development of BSNs, several standard radio protocols have been employed, including Bluetooth, ZigBee, ANT [34], along with ad-hoc proposed solutions [22]. With the diffusion and adoption of smartphones, it has become more convenient to rely on standard protocol supported by almost all mobile phones and tablets, eliminating the need for additional hardware. This resulted in a wide adoption of Classic Bluetooth[1] for the wearable nodes, even with its limitations in terms of number of devices and high-power consumption.

[1] We call *Classic Bluetooth* all the Bluetooth versions before 4.0

The development of the Bluetooth 4.0 standard (also called Bluetooth Low Energy or BLE) was directly targeted to increase the energy efficiency and device connectivity of such protocol, with positive impact for wearable devices and BSNs for both consumer and medical applications [5]. However, it is not compatible with previous versions of the standard, having different stack layers and data exchange protocols and its low energy optimized profiles result in a reduced data rate.
In this chapter we start analyzing the BLE protocol from an application point of view, trying to extract an analysis methodology that can be applied also to other scenarios. For doing so we target data streaming applications with high throughput and reliability requirements, such as wearable healthcare applications, and we explore the capabilities of this protocol in relation to its tunable parameters. To avoid being too biased by the BLE implementation we will repeat the same analysis on three widely used BLE chips.
We consider a BSN scenario with up to 6 wearable devices connected with a smartphone and we focus on application-level development. While existing literature relies mainly on simulation and hardware/software customization of the used devices, we target real-life scenarios for application development, avoiding the need for hardware or stack development. The chapter discusses the observed results and provides indications if BLE is suitable for an application and guidelines for efficient deployment of BLE based BSNs.
In the following Section 2.2 we introduce the related works and, for the reader's convenience, we present an overview of the BLE protocol focusing on the top layers of the stack in Section 2.3. Power consumption is analyzed and modelled in Section 2.4, while Section 2.5 describes the implemented BSN. The experimental setup and results are reported in Section 2.6 and discussed in Section 2.7, together with guidelines for an optimal BLE performance. Finally, the last Section VIII concludes the work.

2.2 Related work

Since the introduction of BLE in 2010, its benefits in terms of connectivity and low power consumption have been exploited in BSNs, with several applications: motion sensing [76, 117, 66], ECG and biopotential sensing [118, 58, 107, 32], blood pressure measurement [72]. BLE is optimized for low power applications and its power consumption has been compared to competing communication protocols (Bluetooth, ZigBee and ANT) [34]. In particular, the comparison with older Bluetooth versions highlights the energy benefits of this widely adopted solution [118, 32].
The drawback of the BLE solution is the relatively low data throughput that a device can achieve. In [46], a model for throughput estimation is presented with a calculated maximum throughput of 236.712 $kbit/s$, but it does not consider the limited hardware resources available on real BLE chips. Furthermore, the calculation is done in master-to-slave direction and a master device usually has significantly less hardware/software constraints than a BSN node. In [83], a BLE performance analysis is accomplished and compared with results obtained from other wireless transmission protocols. For BLE they report a maximum experimental link layer throughput of 122.6 $kbit/s$, but this work did not use the upper BLE stack layers and the achieved throughput considers also the packet overhead, which does not contribute to the application data throughput. In [47] similar theoretical results have been reported with the additions of experimental tests, which highlights that with a real device the

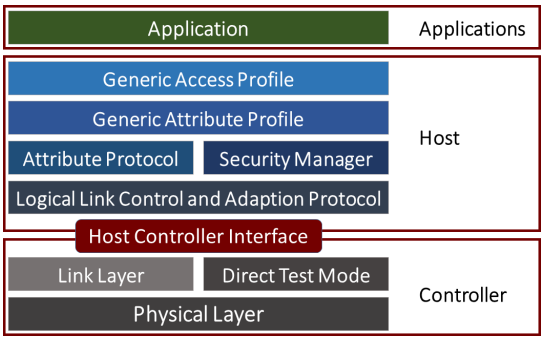


FIGURE 2.1: The graphical representation of the BLE stack.

application level throughput is 58.48 $kbit/s$. In this chapter two identical modules are used for master and slave devices, with custom software on both ends.
To allow the connected devices to effectively employ duty cycling strategies, data exchange in BLE can only be performed sending data packets on pre-determined connection events. Given the application requirements, different choices of the period of the connection events and the number of packets to send at each event can satisfy the needed throughput, affecting in different ways the overall performance of the system. While theoretical and preliminary studies on the connection parameters optimization have been carried out [103], the real application-level performance, which is dependent on the hardware and stack employed, has not been carried out. Therefore, I investigated the performance in a realistic scenario analyzing different commonly used BLE chips for BSN nodes and using an Android smartphone with its default stack. Moreover, I also evaluated a multi-slave scenario, where up to 6 data streaming nodes were connected to a smartphone, analyzing the network performance in relation to the choice of the protocol parameters.

2.3 Overview of the BLE communication protocol

BLE is the last version of the protocol specification designed by the Bluetooth SIG and first released in 2010 [18]. It is targeted to low power, low throughput and low cost devices and it is not backwards compatible with previous versions of the standard. It has been designed for short range wireless communications for BSNs and wearable technologies. As for previous versions, a BLE network has a star topology, with multiple slave devices connected to one master.
The BLE stack (Fig. 2.1) can be divided in three main layers: Controller, Host and Application. Controller and Host layers can reside in two separate chips or on the same chip and they communicate via the standard Host Controller Interface (HCI). Application and Host layers can also reside in separate chips, but no standard exists for their interface.
In this work, I target application level optimization of BLE, hence I focus on its upper layers. In particular, the Logical Link Control and Adaptation Protocol (L2CAP) is responsible of protocol and channel multiplexing, segmentation and reassembly of packets for the lower levels, along with error and flow control. The Security Manager (SM) is responsible for device pairing and key distribution, while the Attribute Protocol (ATT) allows a device to expose a set of attributes (also called Services and Characteristics) and their associated values. The Generic Attribute Profile (GATT)

defines a service framework for the interaction with attributes, and the Generic Access Profile (GAP) defines the procedures related to the discovery and link management of the device's connection. It also defines the role of the device, which can be Broadcaster, Observer, Peripheral, and Central. While the L2CAP is transparent to the user application, it interacts with the GAP and GATT for the data exchange and ATT and SM during connection initialization.

2.3.1 Data exchange in BLE

In a BLE connection there is a Central (or master) device and one or more Peripheral (or slave) devices. The master initiates the connection with the Peripherals and once the desired network is established, the connected devices can expose their Services and Characteristics to the Central device (it can be also that the Central device exposes its attributes to Peripheral device). The device that exposes its attributes is called the GATT Server and the other one assumes the role of the GATT Client. The GATT role is independent form the master/slave GAP role.
Within an established network, data exchange can only be performed through the exposed Characteristics, which are data containers for 8 bit data values, arranged in arrays of up to 512 octets. Similar Characteristics can be grouped into Services and they all are stored in the attribute table in the form of a data structure. Both the GATT server and client have their own local copy of the table.
Each Characteristics has its own Properties that define how the GATT client can interact with it. The available Properties are:

- *Read*: the client can read the Characteristic value;
- *Write*: the client can write the Characteristic value;
- *Notify*: the client can be notified when a Characteristic value has been updated by the server, without the need of a read operation;
- *Indicate*: as Notify, but with an application level acknowledgment of the notification to the server.

There are two ways for transferring data from the GATT server to the client: the first is through notifications or indications and the second is through explicit read requests. With notifications/indications, when the GATT server changes a Characteristic value in its local attribute table, the new value is automatically updated on the client's attribute table. With read requests, the update is done only when the GATT client requests the reading of that value. Transferring data from the client to the server is allowed only through write requests.
Data exchange between connected devices is always performed within Connection Events (CEs) and between two consecutive CEs the radio is in sleep mode to save energy. The time interval between two consecutive CEs is called Connection Interval (CI): the BLE standard permits CIs from 7.5 ms to 4 s (with steps of 1.25 ms). An example of typical current absorption during BLE operations is shown in Fig. 2.2. Another important parameter in a BLE connection is the Slave Latency (SL). This parameter represents the number of consecutive CEs that can be skipped by the Slave device if it has no data to send. In the ideal case, the best overall performance is obtained with SL set to zero [65]. The connection parameters and timings are negotiated at the connection set up and they can be changed any time during the connection. However, each device can accept or reject a timing change request based on its hardware and stack implementations.

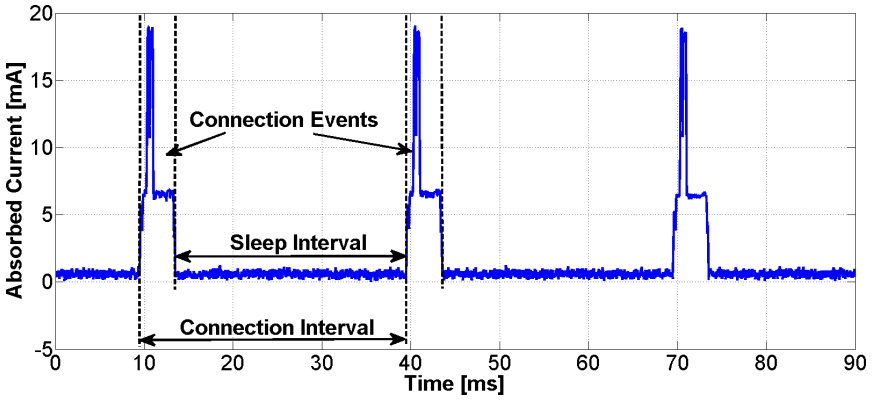


FIGURE 2.2: Example of BLE duty cycling strategies: in this case the CI is set to 30 *ms* and only one PPCE is sent.

Each CE contains at least one TX and one RX event, allowing to send/receive data Packets (i.e. one or more Characteristics values). Each Packet contains up to 20 bytes of application data and the standard defines procedures for segmenting and reassembling longer Characteristics, which will result in more than one TX/RX events to be sent. There is no imposed limit to the maximum number of Packets per CE (PPCE) that can be sent, but in practical applications limitations occur because of the timing requirements, the used stack implementation and the limited hardware resources of the devices.

2.3.2 BLE for streaming applications

BLE is a general purpose protocol suitable for a wide range of applications, but it is mainly optimized towards low power consumption rather than high data throughput. While the low power consumption is an universal specification for battery powered BSN devices, different applications and scenarios may lead to very different data throughput and latency specifications.
Typical use cases may range from the requirement to exchange a few bytes of data per second (i.e. event notification) to consistent data rates to be delivered in real time (i.e. high-resolution ECG or inertial sensing). A 9-axis inertial measurement unit (IMU) sampled at 100 Hz requires 14.4 $kbit/s$, while biopotential applications such as ECG or EMG range from 8 $kbit/s$ for a 3-lead ECG signal to 64 $kbit/s$ for 12-lead ECG signals or 8 channels of EMG sampled at 500 Hz. Moreover, advanced sensor nodes can dynamically change their requirements, adapting data sampling, compression and transmission to the context.
Given this variable scenario, a desired data rate can be achieved with different combinations of CI and PPCE, which can lead to different results in terms of power consumption and data latency. Therefore, in this work I explore the most efficient use of the BLE protocol evaluating the highest application data rate in relation to the connection parameters and the number of connected devices.

2.4 Power consumption model

In order to evaluate the efficiency of a BLE module, I analyzed the power consumption profiles during data exchange and I extracted a model to calculate the current consumption given the CI and the number of PPCE. This model is in a way like to

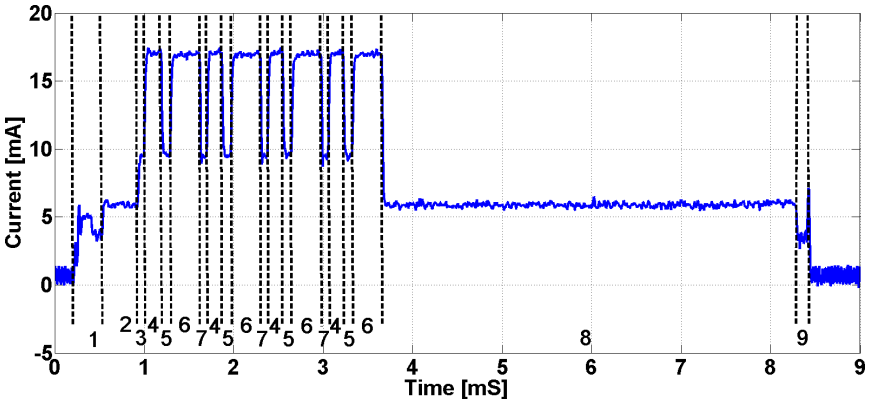


FIGURE 2.3: Current absorption of CC2541 sending 4 packets in one CE: 1) startup, 2) pre-processing, 3) pre-RX, 4) RX, 5) RX-TX inter frame, 6) TX, 7) TX-RX inter frame, 8) post-processing, 9) pre-sleep. The phases 4 to 6 are repeated PPCE times, phase 7 is repeated PPCE-1 times, the rest is executed only once.

the one presented in [9], but targeting the BLE standard and in particular the three chips under investigation in this chapter.
To set up the model, I identified the different phases of the BLE operations as reported by the manufacturers [62, 88] and illustrated in Figures 2.2 and 2.3. Some of the sections are fixed regardless the number of PPCE or CI duration (phases 1, 2, 3, 9), others are repeated PPCE or PPCE-1 times (phases 4 to 7), while others increase their length proportionally to PPCE (phase 8). The model has two parameters, the CI and the PPCE. I considered a constant current absorption during each phase and we expressed all the temporal repetitions and durations in function of the two parameters. The resulting current consumption is then computed as follows:

$$I_{CEmean} = \frac{I_F t_F + PPCE\ I_P t_P + (PPCE-1) I_{P_1} t_{P_1}}{t_{CE}} \tag{2.1}$$

$$I_{mean} = \frac{I_{CEmean} t_{CE} + I_{sleep} t_{sleep}}{t_{ConnInt}} \tag{2.2}$$

In this model, I_F is the mean current and t_F is the duration of the fixed phases that do not depend on $PPCE$ (phases $1, 2, 3, 9$); I_P and t_P have the same meanings but for phases with length proportional to $PPCE$ ($4, 5, 6$ and 8); in the same way I_{P_1} and t_{P_1} are for phase 7 iterated $(PPCE - 1)$ times; I_{sleep} is the sleep current, t_{sleep} is the Sleep Interval and t_{CE} is the total duration of the Connection Event (from phase 1 to the end of phase 9 in Fig. 2.3). In other words, the model is a weighted average average: the current consumption of each phase is averaged weighting it with the relative duration of that phase.
For each analyzed chip, each phase was characterized measuring its mean duration and absorbed current over several intervals, allowing to analytically evaluate the current consumption for different combinations of connection parameters. To validate this approach, I performed extensive current measurements in different configurations and observed an error between the computed and measured currents below 5%.

2.5 Implementation

To evaluate the performance of selected BLE chips, I implemented a firmware to use them as nodes of a BSN. I employed modules from three different vendors and, as much as possible, I implemented the same functionalities on all of them. In particular, to evaluate data throughput and consumption, I programmed the modules to be generators of dummy packets to be delivered to the central unit.
Before a module can transmit data over the BLE connection, it needs to configure hardware (timer, interrupts) and software (Bluetooth stack) components. Once the basic configuration is performed, the needed GATT services are added to the GATT server (which resides on the BLE module) and the BLE is set to be visible and connectible. Now the BSN node is ready, and it waits to get connected by the smartphone.
To implement the data streaming between the nodes and the smartphone, I created a service composed by two characteristics: *DATA_CHARACTERSITIC* (20 Bytes with notify and read properties) and *ENABLE_CHARACTERISTIC* (1 Byte with write and read properties). The first one is for the data exchange: every time the sensor node has new data, it is packaged in 20 Bytes and it is written in the *DATA_CHARACTERSITIC*. The transfer is done with the notification mechanism, so the master device will automatically receive the new data on the next CE. The *ENABLE_CHARACTERISTIC* is used to control the data stream from the master side, enabling it when set to 1 and disabling it when 0.
The high efficiency of the BLE standard is largely due to the duty cycling between active and sleep states (see Fig. 2.2). The BLE radio of a sensor node is usually in sleep and wakes up at pre-defined intervals (CIs) to exchange data with the connected device. To maximize the benefits of the duty cycling policy, our BLE sensor nodes synchronize all their application tasks with the CEs and thus perform all the needed operations right before or after a CE and go to sleep in between.
The CI time is used to define the data throughput/power consumption trade-off. Given a needed application throughput, there is a degree of freedom in setting different combinations of CI and PPCE to achieve it. Since each configuration leads to different performance, these parameters are investigated to find the most convenient settings for the application requirements.

2.6 Experimental Results

2.6.1 Experimental Setup

The test system used in this chapter is composed by three different BLE modules, which have been selected for their availability and large diffusion. The modules are: CC2541 from Texas Instruments (TI), nRF51822 from Nordic Semiconductor and BlueNRG from ST Microelectronics.
Each of them is used on a development board and with the provided proprietary BLE stack (BLE-STACK 1.4.0 for CC2541, SoftDevice S110 v8.0 for nRF51822 and BlueNRG FW v6.4 for BlueNRG). The TI and Nordic modules do not need an external microcontroller to be used and they allow to add some user application code to the microcontroller embedded in the chip (an 8051 for TI and an ARM Cortex M0 for Nordic). The BlueNRG module is different since its embedded Cortex M0 microcontroller is reserved for the BLE stack and it is not accessible to the user. In this case it is mandatory to use an external microcontroller to run the desired application and for the tests I used a development board equipped with a STM32L chip.

The BLE current consumptions and timings used as power model parameters have been measured from the boards using a low-side shunt resistor where possible and high-side shunt resistor with an amplification circuit otherwise.
The master device is a smartphone (Motorola XT1039) running Andorid 4.4.4 with a custom application that connects to the desired BLE devices, receives the data stream and logs it on a file. I used the device as is, without hardware modification and I developed the application using only standard Android APIs to evaluate the use of the smartphone as a master node for reliable BSN applications with streaming nodes. In the BlueNRG module it is not possible to synchronize the application task with the CEs, since they are not notified to higher stack levels. Given this configuration, for a close comparison of the radio chips, for the BlueNRG I report only the measured current consumption relative to the BLE module.

2.6.2 Results

Given the BLE optimization towards low data rates, I first evaluated the maximum application-level data throughput that can be achieved. The standard imposes the minimum CI to be 7.5 *ms*, hence the maximum application throughput can be achieved sending as much as possible PPCEs with the lowest CI. In a scenario where one node streams data to the smartphone, I found that, regardless of the BLE module used, the PPCE is limited by the smartphone (to 3 in this case) and imposes a maximum throughput of 64 $kbit/s$.
With longer CIs, a greater number of PPCEs can be sent and if the application can tolerate some degree of data buffering, the same throughput of 64 $kbit/s$ can be again achieved. To evaluate the performance of the different devices, I thus analyzed all the configurations allowing for the maximum data throughput and compared their current consumption. The result of this analysis is shown in Fig. 2.4, where I plotted the mean current and the energy per bit for each module with different CI and PPCE settings used to achieve 64 $kbit/s$. We can observe how the TI module is able to send up to 15 PPCEs, while the BlueNRG and the Nordic stop at 7 and 6 respectively. The BLE standard does not provide specifications on this, hence these differences are due to the different stack implementations.
In the second experiment I analyzed application scenarios less demanding in terms of data throughput, hence I used the proposed power model to compute the node's current consumption varying the throughput from the maximum value down to zero. Since the maximum efficiency is achieved sending the maximum value of PPCEs allowed, I set the PPCE to that value for each device and decreased the CI from 7.5 *ms* to 4 *s*. The mean current consumption for each device is shown in Fig. 2.5. As expected, the current consumption decreases dramatically with the decrease of the data throughput.
The last experiment was targeted to evaluate applications needing more than one sensor device connected to the smartphone. This multislave configuration has been tested with up to 6 slave devices and each node has been set up with the same configuration (same CI and PPCE). For each number of nodes, I wanted to find the maximum achievable throughput. For this purpose, for each configuration, I recorded 5 minutes of streaming data for 5 times and the result of this test is summarized in Table 2.1. The throughput is calculated knowing the number of PPCE, the payload (20 bytes) and the CI; F_s is the corresponding sensor sampling frequency if one sample is sent each CE. The data logged during each test has been analyzed to verify its integrity and that the streaming was consistent with the settings. When the throughput was more than 3 % lower than the expected one or the lost packets were more

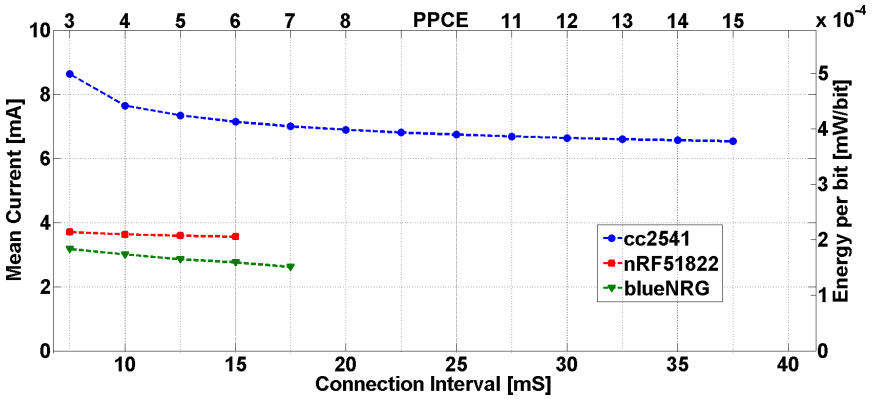


FIGURE 2.4: Mean current and Energy per bit with fixed data throughput (64kbit/s). For each CI the relative PPCE values are reported on the top horizontal axis.

than 3 %, the connection has been considered not reliable. For each number of connected nodes, I lowered the CI until I found a reliable connection. Moreover, in this multislave case I could establish a reliable data stream only with 1 PPCE, even with longer CIs up to 20 *ms*

2.7 Discussion

The reported results highlight that in a single streaming device scenario an optimized choice of the connection parameters improves the system efficiency while maintaining the desired throughput. Buffering data to transmit it less frequently can give up to 47 % of energy saving. This saving is affected by the maximum value of PPCE the module supports, the throughput needed and by the overall current profile of the selected device. In some situations, the proposed optimizations have a limited effect due to the hardware and stack implementations that limit the usable configurations (see the red trace in Fig. 2.4).
As a guideline for optimal BLE connection parameters, one may consider the equation to compute the application throughput in the ideal case (no packet loss):

$$TH = \frac{PL \times PPCE}{CI} \tag{2.3}$$

TABLE 2.1: Result of multislave connection test

n	**CI** $[ms]$	**PPCE**	**TH** $[kbit/s]$	**Equivalent freq.**$[Hz]$
2	7.5	1	21.3	133
3	7.5	1	21.3	133
4	10	1	16.0	100
5	13.75	1	11.6	73
6	16.25	1	9.85	62

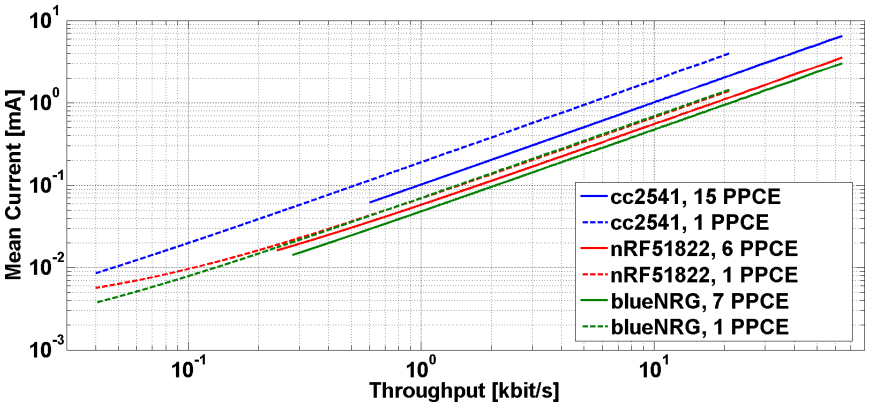


FIGURE 2.5: Current consumption as function of Throughput. This is computed using maximum number of PPCE allowed on each module and for minimum number of PPCE (i.e. one PPCE).

where, PL is the application level payload, whose maximum allowed value is 20 bytes, CI is the Connection Interval and TH is the calculated throughput. This relation can be rewritten as:

$$CI = \frac{PL \times PPCE}{TH}. \tag{2.4}$$

Knowing the throughput needed for the application and the maximum value of PPCE allowed (this is affected by the chosen module, its BLE stack and also by the smartphone and its operating system), it is possible to calculate the CI that gives the best energy performance.
As example, if the application uses data from a 9 axis, 16 bit motion sensor sampled at 100 Hz, each BLE packet can contain an entire sample of the sensor. In fact: 16 bits each of the 9 axis gives 144 bits or 18 bytes each sample, two remaining bytes can be used as counter. The needed throughput is 160 $bits \times 100\ Hz = 16\ kbit/s$. Suppose to use cc2541 module, taking some margins a possible choice for parameters is 10 PPCE and, appling eq.2.4, 100 ms for CI. If a latency of 100 ms is too much PPCE and consequently, CI can be reduced.
It has to be remarked that the proposed optimization is an energy/latency trade-off, since the data needs to be buffered between CEs and therefore a delay is introduced. The introduced latency must be evaluated for each application considering how much data has to be sent, the tolerated error rate, CI and SL parameters. In an ideal case SL should be zero to avoid energy waste on the master side, but if there are packet transmission errors, using a non-zero SL and a shorter CI will improve the overall performance.
In the multislave configuration, I tried to send as many PPCEs as possible, with relatively low values of CI and as many nodes connected as possible. Since the master device can serve only one connection at the time (time multiplexing is used for connecting multiple slaves), the number of connected nodes is dependent on the CI duration and the consequent ability to serve all the desired devices within one interval. Concurrently analyzing the operations and current consumption of a connected network of devices, I found that the distribution of CE slots and their order in one CI are randomly set by the master, they are managed by the Android stack and are not controllable by the user. As general result, when more slaves are connected to the smartphone, we obtain that a slave can reliably send only one PPCE. With longer

CIs (i.e. 100 *ms*) these restrictions expire and more PPCE can be reliably sent.
I have also documented some situations where two nodes are overlapped each other sharing the same CE window. In this case, the connection remains active but data is transferred only from one of the two nodes at the time and they switch their role about every second. Overall optimizations discussed above are valid in one slave configuration or multislave configuration with long CIs (i.e. longer than 100 *ms*). With multiple slaves and CI in the order of 20 *ms* (and shorter) only one PPCE can be sent, therefore the only optimization is to properly set the CI using Eq.2.4. For a comparison, the current Apple stack implementation for iOS limits the minimum CI to 20 *ms* with one PPCE.
In all cases the sensors reading and/or data elaboration in the node should be performed in synchrony with CEs to allow an effective duty cycling and prolonged sleep states. Unfortunately, this is not always possible, since the BLE standard does not define a unified notification of the CE between stack layers. Some of the stack implementations add this feature, which results very useful but at this time it can not be considered part of the standard.

2.8 Conclusion

In this work, I analyzed the BLE standard and its use for BSN applications, with particular interest for healthcare for devices with high data throughput and reliability requirements. I introduced the protocol and its features from an application-level perspective, evaluating its characteristics and how they are implemented in available devices and stacks. I employed a smartphone as the network master device, taking the advantage of the availability of such devices and their ability to handle the needed communication and processing requirements without the need for additional hardware.
An evaluation of three widely adopted BLE modules was performed and maximum application level throughput of 64 $kbit/s$ has been obtained. This limit has been demonstrated to be imposed by the smartphone and its software instead of BSN node. The effect of connection parameters optimization has been evaluated for the BLE chips and their stack implementations. The multislave connection has also been tested, but since in this configuration there are many variables that play a role in network performances and lot of them are out of user control, evaluation has been done in the most critical conditions and with a maximum of 6 active connections.

Chapter 3

The MIGNOLO System: Managing Groups with Bluetooth Low Energy

3.1 Introduction

Recent technological innovations such as miniaturized networked sensing devices, widespread connectivity and advanced but affordable remote storage and processing services have paved the way for the Internet of Things (IoT) [49]. Taken together, these offer a rich ecosystem that enables the development of smart cities and the diffusion of innovative applications. Integrated mobility and the development of IoT systems and services for sustainable, efficient and independent mobility within the future smart cities is an active and inspiring application area [8, 63, 98].

In this smart urban scenario, pedestrian mobility and the management of dynamic groups of people represent an important corner stone. I consider groups such as organized tours, families with children, as well as informal groups of friends. I take inspiration from a scenario with both educational and societal impact, namely the safe and secure mobility of children throughout their neighbourhoods. Within the CLIMB project [1], we are exploring ways that technology can be used to support teachers, parents and students in the goal of increasing both children independence and safety as well as encouraging and educating next generations towards sustainable mobility. The proposed multi-pronged solution incorporates wearable IoT devices as well as gamification to increase motivation and participation.

This chapter focuses on a first step toward the goal of independent mobility, namely supervised mobility of a group of children walking from their homes to the school, known as a walking bus. This scenario, with a few adults responsible for multiple children, maps directly to school outings where the responsibility lies with the teachers. My focus lies with on IoT angle, developing a group management system for individuals carrying wireless, mobile devices. It supports *(i)* group formation, e.g., tracking which children are aboard the walking bus, and *(ii)* membership monitoring, e.g., raising an alarm if a child wanders too far from the *driver* of the walking bus. The solution uses small, autonomous nodes carried by all participants, as well as a smartphone and the accompanying app carried by a single parent monitor.

In this chapter, I first outline related efforts from the literature then focus on the application requirements for the walking bus and how they generalize to other

[1] http://www.smartcommunitylab.it/climb/

group scenarios. I move on to describe the design and implementation of the proposed solution that uses the connectionionless capabilities of the Bluetooth Low Energy (BLE) standard to build a flexible, efficient group monitoring service. This version 1 of the group management system is built using only broadcast messages between group members but it uses this to build bi-directional information exchange between the parent and child nodes to increase the reliability of the system. A more sophisticated communcation is employed at the manager side where broadcast (or advertising) based and connection-based communication are used at the same time (concurrently).

The experimental evaluation analyzes performance in terms of network latency and delay, system scalability and battery life. Moreover, I present results from a controlled test of the walking bus scenario in an outdoor, real-life environment.

3.2 Related work

Group management through wireless wearable nodes has been tackled by the research community in the past using a variety of technologies. The work presented in [25] focuses on the problem of decentralized group management in general, from a more theoretical perspective. Three different protocols are introduced for the dissemination of group membership information. The protocols are compared in a simulated environment in terms of energy consumption, accuracy, and latency, but no real world implementation is provided.

In emergency and/or catastrophic situations, group communication and group membership can be of vital importance, in fact coordinating rescue teams and properly address them in the first hours of emergency is the first priority. A peer-to-peer wireless network, which is conceptually like the proposed solution, that targets this is presented in [71] where laptop wi-fi is used to build the TCP/IP network. The peer-to-peer network just provides the communication tool, on top of this support services such as Walkie-Talkie, Push-to-Talk, and VoIP, they also run the Rescue Information System for Earthquake Disasters (RISED) [57], which helps rescuers by providing accurate and updated rescue-related information such as disaster locations, possible damages to both lives and constructions, available rescue and relief resources.

Other works take a more pragmatic approach concentrating on the problem of child kidnapping, and designing systems for the tracking of children through wearable wireless devices and their physical vicinity to trusted GPS enabled devices. Lee et al. in [70] propose a system based on the de-facto standard TelosB WSN node. Kids wear TelosB devices operating as beacons, while monitoring nodes are composed of a TelosB node with a Bluetooth module that attaches to a smartphone. The system is validated in a very limited setup of 5 nodes.

Recently, commercial start-ups, e.g., Lineable [73], have also begun to address related problems, such as child safety. The Lineable solution is based on a wristband used as a BLE beacon. Beacons are detected by smartphones, and presence information is shared among the participating smartphones through the cloud. None of the above, however, supports bi-directional communication in large groups, a technological advantage that we exploit in our solution.

3.3 Requirements

Recent IoT technologies offer several possibilities for the core of a system to support the management of a group of people in motion. Miniature devices that are capable of dynamically forming wireless networks and sub-networks and estimating their distances represent ideal candidates. Here, I describe the requirements of a group management system composed of low power wearable wireless modules, able to transmit and receive small amounts of data independently from the chosen protocol.

The requirements for the proposed application can be summarized as follows:

- **Discover new users when in range** and, if they are part of the desired group, add them to a current *friend list*.
- **Detect if one of the current members separates from the group**, notifying both the disconnected user and the group manager and possibly the rest of the group.
- **Spatial localization** of the group members relative to the rest of the group or to the group manager. There is no need for highly accurate localization; a coarse, relative estimate is enough.
- **Bi-directional and multi-hop communication** to allow the manager and different group members to sense each other's presence even if they are not in direct contact due to occlusions or interference.
- **Latency and scalability** are interconnected as many nodes with frequent transmissions will result in many concurrent transmissions causing packet collisions and data loss. The maximum number of nodes that can be in the communication range is set to 150 and the maximum tolerated packet loss to 10%[2]. The maximum discovery latency is 5 seconds.
- **Long battery life** to avoid frequent recharging, my target is one school year (i.e. approximately 9 months).
- **Compatibility with established technologies** to allow the system to integrate with existing devices and infrastructure.

3.4 System description

3.4.1 General overview

To implement a group management system, a Leader-Member architecture is defined, with one node acting as the group leader and all others taking the member role. In our network, the Leader also acts as a gateway towards the external world for data exchange and user interaction. Member nodes are intended to be autonomous and require minimal user interaction. While this architecture naturally maps to supervised groups, e.g., with the teacher taking the Leader role, it can easily be adapted to peer groups, e.g., by automatically electing the Leader during group formation.

[2] At this point of the research I had no information to properly tune this requirement, than I choose a value that looked reasonable, however, later on in 5.4 I found that an optimal configuration exist that optimizes latency given the number of nodes. Here I just anticipate that the system can tolerate up to 50 % of packet loss if the latency needs to be minimized.

TABLE 3.1: Matrix of possible Member node's states

		Wireless network state	
		Out of range	In range
Group state	Loner	BY_MYSELF	CHECKING
	Joined	ALERT	ON_BOARD

Given this architecture, the Leader node will track and manage the state of the group, while the Member nodes assume different states depending on their localization and network state. Table 3.1 summarizes the proposed states for a Member node, which is a combination of its wireless link state (i.e. if it is in contact with the Leader) and of its group state (i.e. if it is part of a group). The four resulting possibilities are:

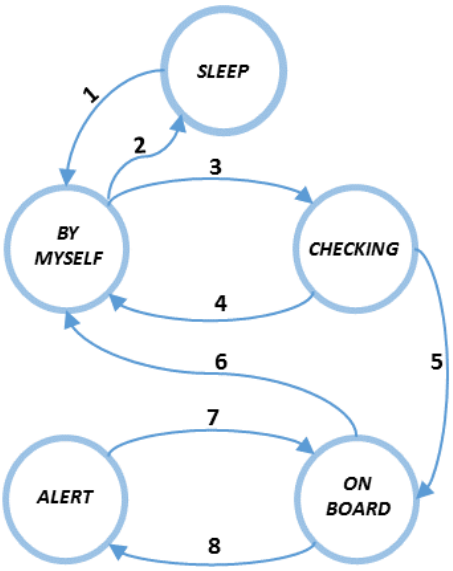
- **BY_MYSELF:** The Member node is far from any Leader and it is not part of any group.
- **CHECKING:** The Member is in direct contact with a Leader node, but it is still not part of its group.
- **ON_BOARD:** The Member is in direct contact with a Leader node and it is part of its group. In this case the Leader is monitoring its presence.
- **ALERT:** The Member is part of a group but, at the moment, it is out of range and not in direct contact with its Leader.

Since the Leader supervises the network, it directs the state transitions of its Members, except for the ALERT state. In this case, when the Member is out of communication range with its Leader, the ALERT must be autonomously detected at both sides. For increased reliability, each state transition is requested by the Leader and subsequently acknowledged by the Member. This approach requires bidirectional communication and ensures correct synchronization of the group nodes states.

The proposed system behaviour is the following:

- At power on/wake up a Member is in BY_MYSELF and remains in this state until it comes in contact with a Leader.
- When the Leader discovers a new Member, it requests that Member to move to the CHECKING state, which will be performed and acknowledged.
- The Leader checks if such node is part of its group, which may happen according to a set of policies (e.g. accepting all nodes, checking a list or requesting user interaction).
- To add the Member to the current group, a state transition to ON_BOARD is requested from the Leader and acknowledged by the Member. From now on, both the Leader and the Member will monitor each other's presence.
- If the Member goes out of the Leader's range and stops receiving its communications for a certain time both nodes will trigger the transition to ALERT and will notify the user.

The complete set of states and the events that trigger their transitions are depicted in Figure 3.1. I also consider a SLEEP state when a node is not in use.



	Event
1	Timer or button event
2	Timer or button event
3	The Member node enters in the Leader communication range
4	The Member goes out of Leader range or the Leader reject to monitor the Member
5	The Leader adds the Member to its friend list
6	The Leader stops monitoring the Member
7	The Member returns in Leader communication range
8	The Member no longer hears the Leader

FIGURE 3.1: Member state machine along with the high level events that trigger the state transitions.

3.4.2 Network implementation

The proposed group monitoring system is based on the exchange of wireless broadcast messages among proximate nodes. In a nutshell, the group Members periodically announce their presence. The Leader listens for these announcements and forms a local group membership list. To control the state changes of the Members, as outlined in the previous section, the Leader periodically announces this list, which is received by the Members, who update their states.

The current implementation is built on top of BLE, from which we utilize the *BLE advertiser* and *BLE observer* modes for respectively sending and receiving broadcast messages without establishing connections. BLE natively supports periodic advertisements, at the so-called *advertise interval*, T_{AI}, in the range of 20 ms to 10 s. This forms the core of the Member behaviour, where the advertisement message of the Members contains the 8-bit node identifier and the application state, which is sent every $T_{AI,m}$. Immediately after each transmission, the Member switches into the observer state to listen for other advertisement packets. The Leader also uses periodic advertisements to announce its state, namely a list of the nodes in its group, and the application state that the Leader intends for them to switch to. This period is set to $T_{AI,l}$. The Leader switches to listening between advertisements.

It is worth noting that the payload of the BLE advertisement is 31 bytes long, limiting the size of the membership list to 9 ID-state pairs. To handle larger groups, the Leader cycles through the Member list in subsequent advertise packets. For example, with a group size of 12 Members, the Leader will announce the first 9 in one advertise packet, then after $T_{AI,l}$ it will announce the remaining 3 and will repeat the first 6. This scheme offers a deterministic communication latency, dependent on the maximum size of the group.

To increase usability, this core solution is extended in three key ways. First, we note that the Members must actively listen for advertisements from the Leader in order to update their states. While listening, Members also overhear the advertisements of other Members. In this first extension, we simply save this information to augment knowledge about node connectivity. Specifically, each Member node maintains a list of its neighbouring nodes, storing their ID and the RSSI values of

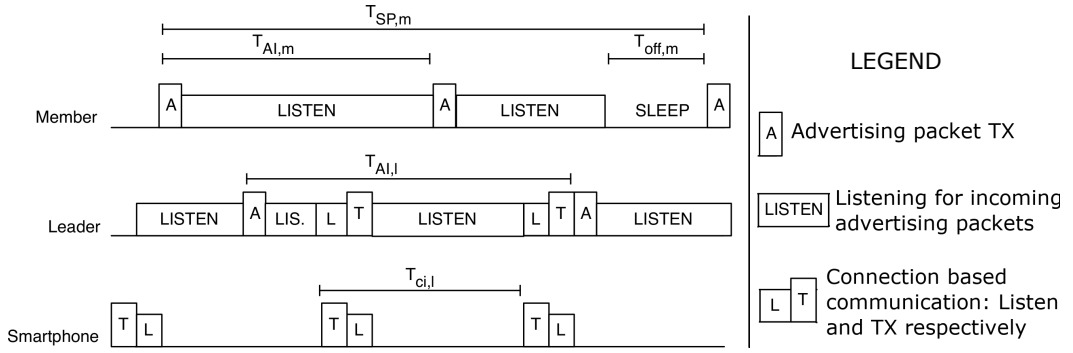


FIGURE 3.2: Schematic of the communication pattern. Note that this is not drawn to scale. Specifically transmission events have a very short duration.

the received packets. This information is sent along with the ID-state pair as part of the Member's advertisement. As we have the same limit to pack at most 9 pairs of information in each advertisement, we employ the same mechanisms to rotate among the neighbour information in subsequent packets. This allows the leader to discover nodes within two hops. Given that the communication range of BLE in outdoor spaces reaches 60m [3], a two hop detection means an (ideal) radius of detection of more than 120m. This is considered enough for the walking bus application.

Second, while Member nodes are required to listen to receive their updated states from the Leader, we must also consider the battery consumption of these devices. Therefore, I choose to put the radio to sleep for slightly less than one advertising period $T_{AI,I}$ every two periods, yielding the periodic behaviour shown in Figure 3.2 in which the node wakes up, transmits its state and the connectivity information of some of its neighbours, then either listens for the whole period or listens for slightly more than half the period then goes to sleep. Clearly when the node is sleeping, it does not hear the advertisements of the other nodes, and it may miss the Leader requesting it to change state. Nevertheless, as the node is likely to hear the subsequent transmission, the overall, correct behaviour is maintained, albeit with a delay acceptable in the target applications.

Finally, I extend the behaviour of the Leader node to allow communication with a proximate, more powerful device, e.g., a smartphone, useful for user interaction. For this, a connection between the Leader device and the smartphone is established, with the Leader acting as a *BLE peripheral*. Information between these devices is exchanged at a period of $T_{CI,I}$. This is shown schematically in Figure 3.2 as short listening (L) and transmission (T) events between the Leader and the smartphone.

3.5 Experimental Evaluation

3.5.1 Experimental Setup

The prototype of the proposed system is based on the CC2650 chip by Texas Instruments [27]. It is a System-on-Chip, which includes all the RF circuitry and a Cortex M0 core dedicated for the lower layers of the BLE stack implementation and one additional Cortex M3 core for user application and higher BLE stack layers. For the test deployment, I used the SensorTag development kits, which include the CC2650, a set of sensors (inertial, temperature, light), two application LEDs and buttons on a $32 \times 42mm$ board powered by a coin battery (CR2032). Both Leader and Member

[3] This data refers to a preliminary experiment. Architecture: CC2650, Tx power: 0dBm

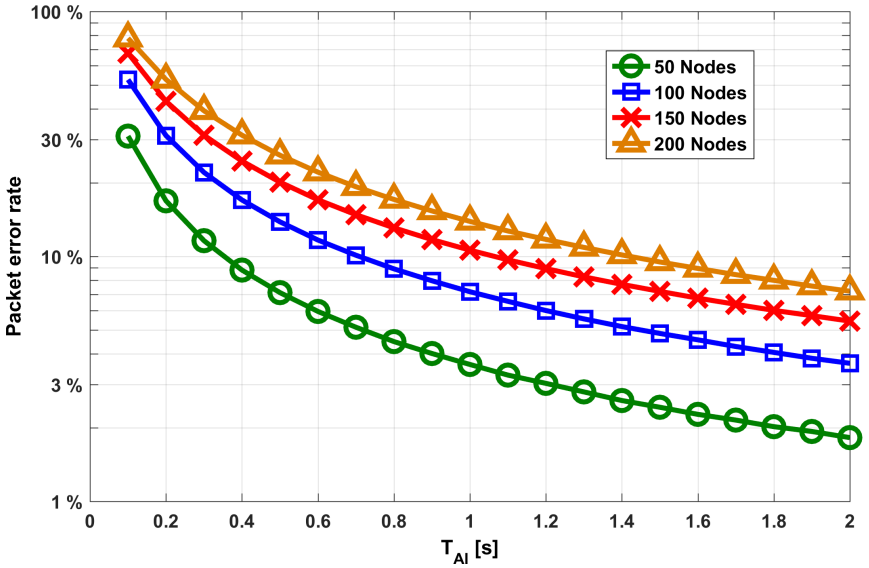


FIGURE 3.3: Percentage of corrupted packets with respect to the advertise interval (t_{AI}) for 50, 100, 150 and 200 nodes in range.

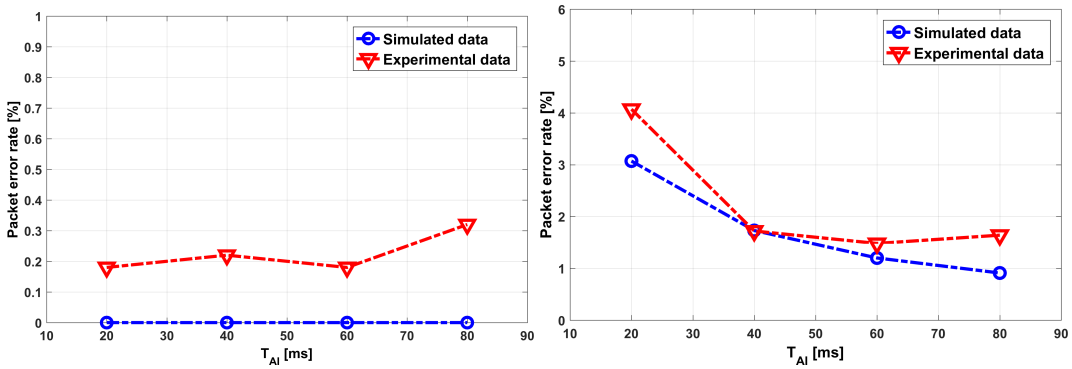
nodes are implemented using the same hardware, albeit with different firmware. The final implementation includes a Nexus 9 tablet that runs an application on the Android 5.1.1 operating system.

3.5.2 Network density and timings

As reported in the previous Sections, the advertise interval T_{AI} sets the trade-off between the network's latency and the supported node density. The latter dictates the maximum number of co-existing nodes and is limited by the loss of advertising packets due to collisions. To obtain the T_{AI} that guarantees the target performance, I simulated the BLE Link Layer to estimate the Packet Error Rate (PER) given a number of nodes and a T_{AI}. Simulations are based on a custom code developed in Matlab, whose purpose is to detect packet collisions given a network configuration (T_{AI}, number of nodes, payload length) and protocol definition (protocol parameters and rules are taken directly from the specification document [18]). A deeper description of the simulator is given in Section 5.3.

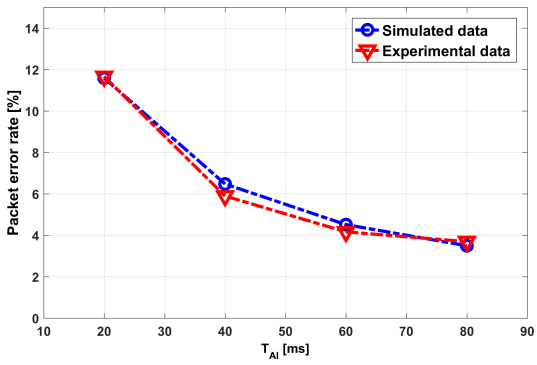
To validate the simulator, I carried on the same test, first on real hardware, then on the simulator, if the results of the two tests are coherent the simulator is validated. Because, at that time, I had 10 actual nodes in the laboratory for testing, I validated the simulator using a restricted setup with those nodes. To not stretch too much the validity of the experiment, I decided to use shorter advertising interval in order to maintain similar density for the number of packets transmitted in a given time interval. I performed the experiment in the office laboratory environment, with nodes uniformly distributed in a $0.5m^2$ area, placing them well within communication range. I experimented with 1, 2, 5 and 10 transmitting Member nodes and with 20, 40, 60 and 80 ms advertise intervals. Any experimental result given in this Section represents the average over 5000 packets, while any simulation result is based on 10000 independent and randomized trials.

The results of the tests are shown in Figure 3.4, where we observe that the simulation closely matches the experimental validation. The variations can be attributed to the fact that the simulation does not consider some hardware characteristics such as channel switching by the receiver, concurrent accesses to the radio peripheral and

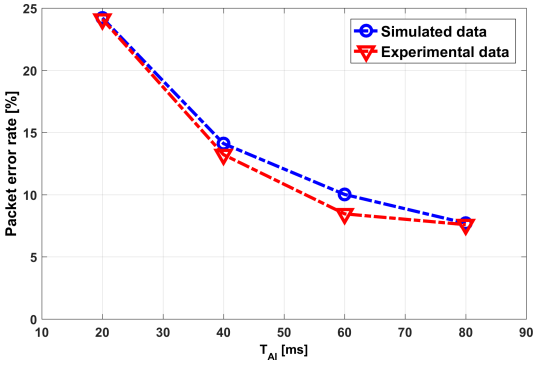


(A) 1 Member. Irregularities due to the small y-axis scale.

(B) 2 Members



(C) 5 Members



(D) 10 Members

FIGURE 3.4: Packet error rate obtained from simulation (blue/circles) and from experimentation (red/triangles) using different numbers of Member nodes. Note: scale varies across figures.

external interference, typical of an office scenario (e.g. wi-fi, mobile phones). These side effects are particularly visible in the experimental PER with only one transmitting node (Figure. 3.4-a). In this experiment, no packet collisions occur but I measured an average of 0.2% of lost packets, which can be attributed to receiver channel switching and to the interferences found in the office (such as other BLE devices or wi-fi). The simulator is indeed valid.

Considering the network simulation and its experimental validation, the Member advertising interval is set to 1 s ($T_{AI,m}$= 1s), which leads to an average packet error rate of 10.6% with 150 nodes. Since the Leader node must track and manage all the other nodes, its advertising interval is lowered to 0.625 s ($T_{AI,l}$= 0.625s). The transmission of advertising packets employs the radio for 3 ms and the remaining time the nodes are free to switch to listening for incoming packets using the BLE observer mode. To increase energy efficiency, I duty cycled the receiver with 1.25 s of activity every 2 s. This solution gives us a Member listening duty cycle of 62.5%, which reduces the power consumption of the device and leads to the timing profile illustrated in Figure 3.2.

Time multiplexing is used to manage concurrent transmission and reception accesses to the radio peripheral inside the CC2650. Therefore, even if we want the Leader's receiver to be always-on and ready to receive the advertise packets, this is not achievable due to the need to transmit advertise packets. In our setting, we measured the Leader's listening duty-cycle as 93.6%. The remainder of the time (6.4%) the Leader's radio is involved in the transmission of advertise packets or in exchanging data with the tablet through the BLE connection.

3.5.3 Discovery latency and round-trip delay

To evaluate the average time for the leader to discovery a member node, I set up one Member and I connected the Leader to the tablet via BLE. This setup recreates the test base in which the Leader is paired with a smartphone for system configuration and interaction.

I define the discovery latency (t_D) as the time from the instant a Member device enters the Leader's range until the moment a notification of its discovery reaches the smartphone app. The discovery latency can be seen as a one-directional network delay, from a Member to the smartphone. In the worst case (t_D^+) when the receiver is always on, this is given by:

$$t_D^+ = t_{CI} + 3t_{AI,m}^+ \tag{3.1}$$

where $t_{CI} = 220ms$ is the connection interval for the BLE connection between the Leader and the smartphone and $t_{AI,m}^+ = 1.01s$ is the worst case advertise interval for a Member, leading to $t_D^+ = 3.25s$.

To limit the probability of repeated packet loss due to radio artifacts, the connection and the advertise events (that occupy the Leader's radio) are executed at non-multiple time intervals ($t_{CI} = 220ms$, $T_{AI,l}$= 625ms). This ensures that the maximum number of consecutive lost packets is limited to two. In fact, the worst case is when the Member's first packet is transmitted during a Leader's connection event and the second one is transmitted during a Leader's advertise event. In such case, the subsequent packet will be received by the Leader, since it will be broadcasted in a Leader's listening interval (see Figure 3.2).

To experimentally measure the discovery latency, I used the approach of [11], in which the process of entering the node's communication range is simulated for

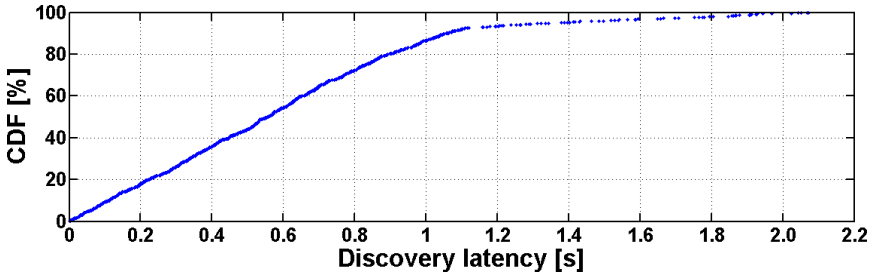


FIGURE 3.5: Cumulative distribution function (CDF) for a new Member entering in the communication range of the Leader.

practical reasons. Specifically, the Member node being examined is considered to be out of range until a randomly chosen time (t_0). Then, the time when the node is detected, denoted as the contact time t_c, is identified, and the discovery latency is calculated as $t_D = t_c - t_0$. This sequence is repeated 1000 times and the obtained cumulative distribution function (CDF) is showed in Figure 3.5. As only one Member was active, this evaluation is of a collision-free configuration. The results show that in 93% of the cases the node is discovered within 1.12s. The reminder of the cases (approximately 7%) are detected within 2.2s, which is below the calculated value of t_D^+, and represent the cases where the first packet is lost due to the Leader's radio transmission multiplexing, but the second one is correctly received. While this test never observed two consecutive packets losses, one cannot exclude pathological cases when packets continue to collide. In other words, practically speaking, I have identified a maximum discovery time that is valid in a collision free environment, but theoretically, longer discovery latencies are possible in a realistic use case environment. These measurements closely match our earlier estimates.

The round-trip delay (t_{RT}) is the time for a packet, originating in the app, to reach a Member node and return back to the app. This is important since it represents a node state change request and its acknowledgment and it strongly depends on the network timings. Summing up the time intervals and delays the notification encounters through the network, it is possible to calculate the worst (t_{RT}^+) and best (t_{RT}^-) cases:

$$t_{RT}^+ = 2t_{AI,l}^+ + t_{off,m}^+ + 2t_{AI,m}^+ + 3t_{CI} \quad (3.2)$$

$$t_{RT}^- = 2t_{AI,l}^- + t_{AI,m}^- \quad (3.3)$$

Here, $t_{off,m}^+ = 770ms$ is the worst delay caused by the sleep within the scan period ($T_{SP,m}$, see Figure 3.2), $t_{AI,m}^+ = 1.01s$ and $t_{AI,l}^+ = 0.635s$ are the worst case advertise intervals for the Member and Leader nodes and the respective best cases are $t_{AI,m}^- = 1s$ and $t_{AI,l}^- = 0.625s$. With the chosen parameters we obtain $t_{RT}^+ = 4.72s$ and $t_{RT}^- = 2.25s$. Even with the previously described Leader's receiver duty cycling, during the experimental validation I obtained $t_{RT,MAX} = 4.54s$ and $t_{RT,min} = 2.68s$, which is in between t_{RT}^- and t_{RT}^+, the average measured value is $t_{RT,avg} = 3.37s$.

3.5.4 Memory requirements

Another key factor to allow the system to manage several tens of nodes is the memory footprint on each device. The Leader node must track all of the Member nodes with their states and the Member nodes also track their neighbouring nodes. The MCU on the device is equipped with 20kB of SRAM. The application, the BLE stack and the operating system use 17.4kB, leaving 2.6kB for storing node information.

To store information about each neighbouring node it requires 35 Bytes of memory, hence each node can handle and track up to 76 devices. While this is smaller than the maximum value of 150 established in the requirements, we can imagine reaching larger group sizes by managing multiple co-existing groups with different Leaders.

3.5.5 Power consumption

To estimate the battery life of the proposed implementation, I applied the following consumption model:

$$t_{BL} = \frac{C_{batt}}{\delta I_A + (1-\delta) I_S} \tag{3.4}$$

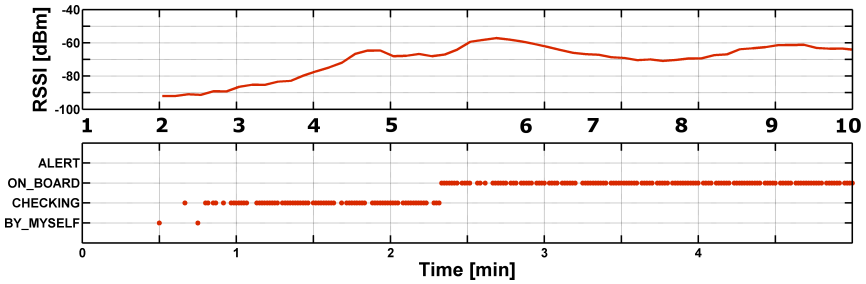
where t_{BL} is the battery life in hours, C_{batt} is the battery capacity in mAh, I_A is the average active current and I_S is the sleep current. I assume that the device is not constantly used and δ is its usage duty cycle (i.e. if the device is used only one hour per day $\delta = 1/24$).

For our platform and the chosen parameters $I_A = 3.6mA$, $I_S = 100\mu$A and $C_{batt} =$ 225mAh. The resulting expected battery life, if the device is used for one hour a day, is $t_{BL} =$ 900h (or 37.5 days). This is an optimistic estimate since it does not take into account the processing each node must perform, which slightly increases the average current absorption, and the battery's internal resistance. In an experimental test in the same conditions, I measured a battery life of 24 days. This is clearly shorter than the target of 9 months, then further work is required for the energy optimization of the devices.

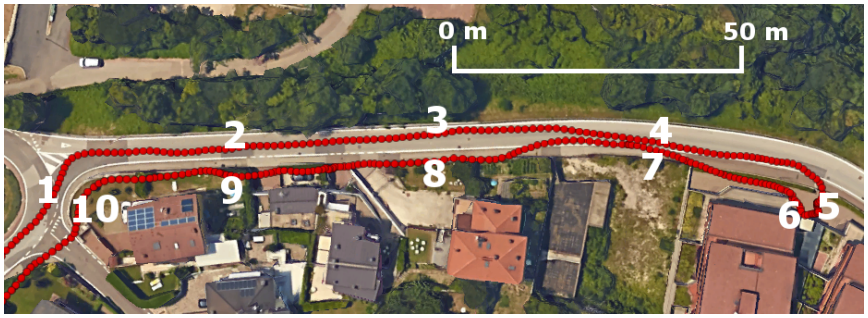
3.5.6 Case study: Walking bus

To validate the proposed system, I performed an experiment simulating a walking bus with our research staff members. For one week, every day at the same hour, 10 people met and walked together on a 400 m path using our system to manage the group. The outcome of this experiment was the log of each Member's state, the RSSIs among all node pairs and the GPS log from the leader's smartphone. As example, Figure 3.6a reports information for one Member during one of the performed walks, summarizing the Member to Leader RSSI and the associated Member state. In Figure 3.6b part of the Leader's GPS trace is reported and superimposed on the map of the route. The numbers identify the sequence of the walk and align the spatial information to the collected signals.

During this walk, the Member in focus was waiting in position 5 in Figure 3.6b (on the right side of the picture) and the Leader user approached from the left, following the numerical sequence. Once the Leader reached the Member, he checked him in (the state changes to ON_BOARD, as seen in Figure 3.6a just before minute 2.5) and they continued together following the outlined path. The Member to Leader RSSI has been filtered applying a 30s sliding window with a 25s overlap. The test was driven by the need to observe the network in an outdoor environment and to identify possible issues or bugs. Further testing will be performed focusing on the system performance and on the estimation of distances between group members.



(A) The RSSI from the Member to the Leader (upper trace) and the Member state (lower trace).



(B) Aerial view of the Leader's position recorded with GPS (red dots).

FIGURE 3.6: Results from the case study. The Member awaits the Leader at position 5. The Leader follows the path as indicated by the numbers. Once they meet, the Leader checks in the Member and they continue together.

3.6 Conclusions

I developed and evaluated the version 1 of MIGnOLO, a system for managing dynamic groups of people using the Bluetooth Low Energy technology. The main difference between the proposed system and those based on beacons is that the Member nodes are also configured to receive, unlike other systems that only transmit. This allows monitoring of Members that are not in direct contact, but are within two hops from the Leader. However, this makes the node consuming more energy because of the BLE scan procedure. I clearly see opportunities to further reduce the consumption at each node since this is the main drawback of the proposed system. I also note that this implementation today offers a clear proof of concept that such systems are feasible for Smart City scenarios, especially considering our use of the standard BLE protocol, which is supported by a majority of smartphones on the market at the time of publication.

Starting from the application requirements in 3.3, I developed the system design, tuning the parameters to achieve the wanted performance (the battery life requirement has not been met). Further, I performed extensive experimental validation to verify the effectiveness of the proposed approach. The proposed system has been tested "in-field" and the results are encouraging. The system acquires redundant information on RSSI (each node's RSSI with respect to all the others), with the idea of exploiting it to study group behaviour, e.g. in terms of proximity, regularity, density, group shape etc.

Chapter 4

Exploiting relative RSSI to estimate group shape through cooperative localization: a negative result

4.1 Introduction

Given the version 1 of the MIGnOLO system described and validated in the previous chapter, the natural follow up was to deal with node localization. As previously described, the developed network let us collect data about the relative RSSI values between group members, and since RSSI can be used as raw distance estimation I applied few state-of-the-art techniques to extract group shape information from reciprocal RSSI between nodes. This technique is defined *cooperative* because not only a minor fraction of the devices are used as reference (i.e. the anchors), but also, measurement from peer nodes are exploited. Cooperative localization is deeply reviewed here [92].

It is well known in literature that RSSI is not the most reliable metric to be used for localization purposes. However, RSSI is available for all the links within two hops from the master. This results in the availability of redundant data i.e. calculation of the same distance provided by different sources. Redundancy of independent measurements can in some cases mitigate the error and therefore can be used to compensate the unreliability of the single RSSI measurement.

In this chapter I briefly describe some of the ideas that have been tested in the context of localization using RSSI data. Unfortunately, none of them brought to a significant reduction of the error, then I decided to do a step back and change direction. However, since the exploration has been conducted with a scientific approach, it still had some value. In the following, I will present briefly the experimental study performed and I will draw some conclusion as lesson learned by analyzing what was wrong in the approach.

4.2 RSSI based localization

Regardless the used radio technology, RSSI based localization is very attractive because of two main reasons:

- RSSI is easily available on all modern CMOS radio; the hardware already embeds all the circuits for measuring it energy efficiently
- Measuring RSSI does not impact on the communication and can be done on any kind of packet, making it very suitable when scalability is a key aspect.

For this, consistent efforts have been spent in literature [10, 61, 55, 92] in analyzing techniques for estimating node position from RSSI data.

The basic idea behind this technique is that the power density of radio signals fall with a known law as the distance between the transmitter and the receiver grows (Friis equation [102]), and the RSSI represents exactly the RF power at the receiver's antenna. Given this, the node position can be calculated with two steps:

- First: distances (or ranges) between nodes in the network is estimated using the RSSI
- Second: the nodes coordinates are calculated so that all the estimated ranges are respected.

If distances are estimated with zero error this would lead to the correct results. However, if there are errors in the estimations (as it happens in real world) the second phase can only calculate the coordinates that minimizes the difference between the node distance in the calculated coordinates and the estimated ones.

4.3 Approaches

Various approaches with different complexities have been tested for the first step (range estimation), while the second (the positioning one) remains the same. A deeper analysis of the positioning algorithm is provided in Chapter 6; however, the coordinate calculation procedure can be intuitively explained with a metaphor from physics. In fact, if all the nodes are transformed into points in the reference plane, radio links can be represented with springs connecting the points and the natural (or resting) length of the springs equals the estimated range between the nodes. Furthermore, nodes' location can be calculated by letting nodes move while springs relax to the stable position. Mathematically this means that we are searching the point of minimum energy of the springs system; it can be expressed as an optimization problem where the target is to minimize the cost function that is the total elastic energy contained in all the springs.
In other words, the problem requires to solve:

$$\underset{P}{\text{minimize}}\ f_{cost}(P) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\frac{1}{2}q_{i,j}(\hat{r}_{i,j} - ||p_i - p_j||)^2 \tag{4.1}$$

where: P is a Nx2 matrix whose rows ($p_k = \{x_k, y_k\}$) represent the 2D position of the k-th node, then N is the amount of nodes and $\hat{r}_{i,j}$ is the estimated distance between node i and node j.
If we apply the metaphor if springs, $q_{i,j}$ is the elasic constant of the spring between node i and node j. However, this can also be a parameter that can be used to weight the importance of each link in the cost function. An option is to set $q_{i,j}$ to have an inverse proportionality with the expected error in the range estimation, in this way the links whose range is estimated with poor accuracy are *less trusted* then the others in the minimization process. For instance, a popular approach when RSSI is used for ranging is to use $q_{i,j} = \frac{1}{\hat{r}_{i,j}^2}$ [48, 85]. This comes from the observation that the logaritmic relationship between RSSI samples (given in dBm) and the link range makes the expected error to be proportional to the link range (Figure 4.1). The intuition given in the figure is confirmed, more rigorously and with the support of math in [96].

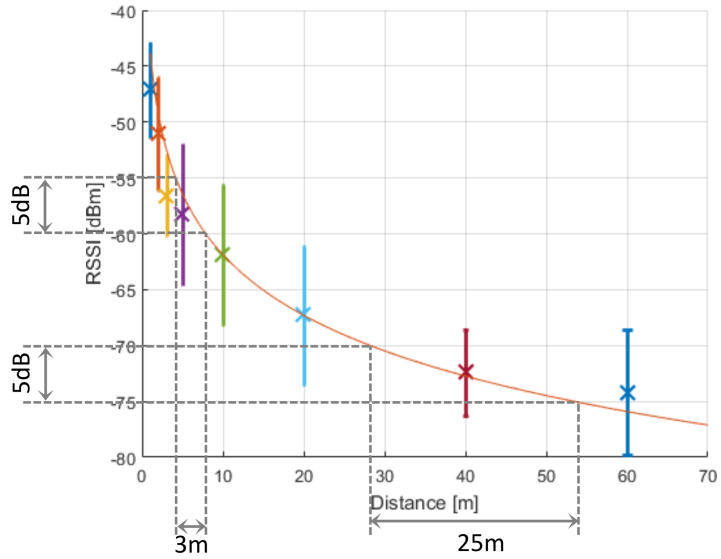


FIGURE 4.1: Simple RSSI model. A constant error interval (5dB for shake of simplicity) in the RSSI context, becomes an error that is proportional to the distance when it is passed through the logarithmic model. This is a property of the logarithmic model that describes the way signal propagates in the space and it remains valid if the logarithmic model is approximated with a polynomial one.

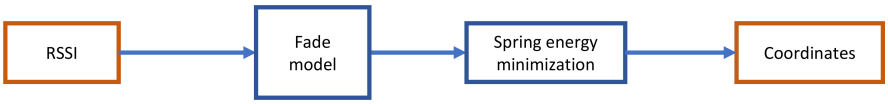


FIGURE 4.2: Schematic view of the Basic-LR algorithm.

Since in Equation 4.1, the sum is iterated on all the links of the network, and since links are between peer nodes, this kind of localization is said to be collaborative. This means that each node with its links not only contributes to localize itself but also to localize the other nodes.

4.3.1 Linear regression on basic model (Basic-LR)

The first and simplest model that is tested is mono-dimensional mono-tone. It represents how RSSI fades with distance and the exact model employs a logarithmic function [92] that can be seen in Figure 4.1. Alternatively, a polynomial approximation can be used without relevant impact on error.
A graphical representation of the model is provided in Figure 4.2. Given its simplicity this family of models is often employed when no other information rather than RSSI is available. The low dimensionality makes it unreliable, in particular in presence of reflections or in case of not pure monotonic signal fading, which often happens indoor. Nevertheless, since it is the basic and most used model we use it as baseline to compare the other solutions.

4.3.2 Linear regression on angles augmented model (Angled-LR)

The previous model hides, among many, a strong assumption that is the omnidirectionality of the antenna. Radiation pattern of the antennas are never perfectly

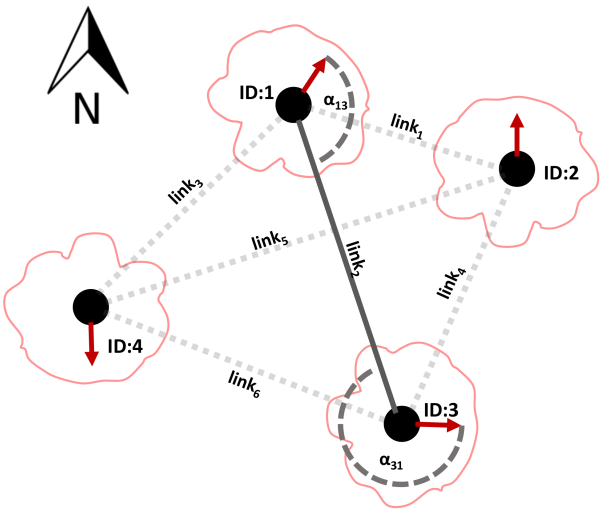


FIGURE 4.3: Example of the network where the localization algorithms are applied. All the nodes are peers and no reference anchor is present. Red lines depict an example of irregular antenna pattern, arrows represent node orientation with respect to the magnetic north (bearing) and two link angles (angle of departure at the transmitter side, angle of arrival at the receiver side) are named with α.

isotropic in the real world and this directly impacts on the RSSI value sampled.

In an attempt to solve this problem, I adopted an iterative method, where both RSSI and angles are exploited to calculate the link range. It must be noted that angles of arrival and angles of departure (I call them α_{ij}) depends on relative bearing of the nodes[1] and also on their relative position (Figure 4.3), but the nodes position is what the algorithm is supposed to calculate.
The intuition behind the proposed method is that the previous algorithm (*Basic-LR*, 4.3.1) is supposed to give a raw node map that does not consider the angles, once this raw map is available the relative bearing angles can be computed and a new set of distance estimation, which consider angle information, can be calculated. This new set of distances (or ranges) are then hopefully more accurate than the previous one, and a more precise map can now be computed. This new map will lead to new angles that will lead to a new set of distances and a new map, the procedure becomes hence iterative (Figure 4.4). This process can be iterated until some stop condition is met.
To avoid the divergence of the solution, the effect of the angle on the range estimation is slowed down by means of the *angle correction rate* (γ), which acts similarly to the learning rate in the gradient descend algorithm.

4.3.3 Neural network on decomposed network basic (Basic-NN)

Neural networks (NN) are very powerful tool to make a machine (i.e. a computer) to learn very complicated patterns, which are difficult to be detected by a human eye. I tried to apply neural networks to estimate links range given the links RSSIs. The idea is that the neural network should learn to distinguish and discard bad links by having a global view of the sensor network.

[1] Relative bearing (or relative orientation with respect to magnetic north) can be obtained by means of a magnetometer used as compass.

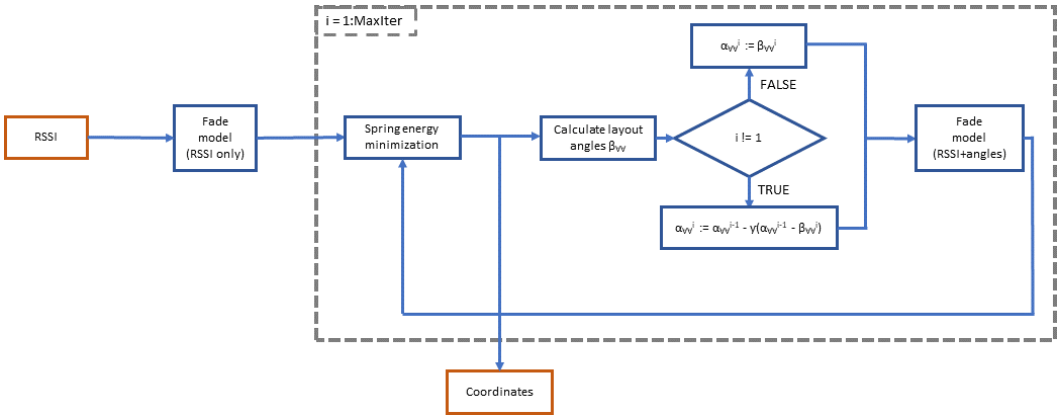


FIGURE 4.4: Schematic view of the Angle-LR algorithm. Angles are actually used only after the *Spring energy minimization* block is executed once, otherwise angles are not available and angle models cannot be applied

FIGURE 4.5: Schematic view of the Basic-NN algorithm. To use the same NN with any amount of nodes, the network is divided into triangular sub-networks and the NN model is applied on triangles. Once the length of the sides of the triangles are estimated the network is reconstructed.

Oversimplifying all the theory behind them, the NNs are machine learning techniques that have a given number of inputs (in this case one input per link, which fed with the RSSI samples of that link) and one or more outputs (the estimated links range). From a practical point of view this means that a NN will be trained for a given amount of links, and it will work only with that number of links. Then if the architecture of the sensor network change for any reason (i.e. one node goes out of range) the NN will not be able to cope with it. Having many NNs, one for each amount of links, does not seem a feasible approach. Therefore, my proposal is to decompose the network into basic building blocks (sub-networks composed by only three nodes, called triangles), run the NN on these blocks and then rebuild the network from the blocks. In this way the same NN with three inputs can be used on any network with more than two nodes.
The graphical representation of the algorithm is depicted in Figure 4.5 where the similarity with *Basic-LR* (Figure 4.2) can be easily noted.

4.3.4 Neural network on decomposed network with angles (Angled-NN)

As for the linear regression approach, also with neural networks I tried to consider the angles by means of the iterative method; employing a raw distance estimation that is then refined by embedding angle estimations. This is exactly the same intuition as in 4.3.2, however, in this case, the distance estimation is performed by the means of a neural network that decomposes the network in triangles as described in 4.3.3.
The first estimation is therefore made with *Basic-NN* algorithm, angles of arrival and angles of departure are calculated, new ranges are estimated considering also

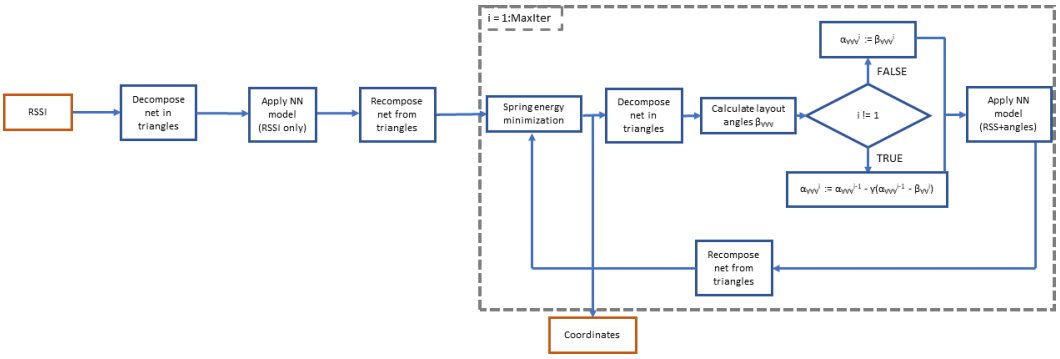


FIGURE 4.6: Schematic view of the Angled-NN algorithm. Angles and position are calculated and the iterative method is executed similarly to Angle-LR algorithm shown before.

the angles and then a new set of coordinates are calculated. As for the *Angled-LR* algorithm (4.3.2) the *angle correction rate* (γ) is employed to stabilize the output (Figure 4.6).

4.4 Experiments

4.4.1 Setup

Since the first aim of the experiments was to asses if angles information could help increasing accuracy in the context of anchor free collaborative localization, the test setup is such that the most diverse angles are present in the network. For doing so 9 nodes are employed, 8 are laid out on a circle and one stays in the centre, the I varied the radius of the circle ($r = \{1, 3, 6, 9, 12\}$) as well as nodes orientation as depicted in Figure 4.7.
The signal reflections and non-line of sight problems are reduced to the minimum by doing the test in a big open field without any obstacle in the first 50 meters in all direction from the test area. To better focus on localization the nodes behaviour have been slightly modified with respect to what described in Chapter 3. For these experiments the packets are sent with 100 *ms* and the receiver duty cycle is 1, then the BLE radio is either transmitting or receiving, it never goes to sleep. This ensure a richer data collection at price of more consumption and a reduced scalability. Network and energy optimizations are eventually left as future step.

Sometime machine learning algorithms learn too much from examples and if no precautions are taken the algorithm tend to *learn the example* rather then *learn from examples*. This effect is called overfitting and it means that the algorithm is not well generalizing the patterns it finds in the training data. When it happens the algorithm outputs very good estimations when tested with the same data used for training; however, when new and unseen data is used the algorithm totally fails in the estimation. To cope with this a popular approach is to train the algorithm using a data set (train set) but to use another data set (test set) to test it. By monitoring the error obtained with the test set during the training process it is possible to stop the training before overfitting. Ideally the train and test set should contain uncorrelated data, otherwise, the effect is similar to use the same set for training an testing. To ensure this, in the experiment train and test data comes from different sessions with

TABLE 4.1: Ranging error values, obtained before the positioning step, i.e. before the spring energy minimization.

Algorithm	Linear Regression	Neural Network
RMSE [m]	1.28 m	1.16 m
RMSNE [%]	32 %	36 %

different nodes layout. For the training the layout shown in Figure 4.7 is used, while for the test, the layout of Figure 4.8 is used.

4.4.2 Used evaluation metrics

Since the localization is purely relative, the localization error (intended as the distance between real position and estimated ones) cannot be calculated. In fact, to calculate the Euclidean distance, the two layouts (the ground truth and the estimated one) must be aligned, but since no absolute reference is considered the alignment may only be done with best fit algorithms (i.e. linear regression). Then the error would depend on how well the minimization algorithm aligns the two layouts.
To get rid of this problem, error metrics are calculated on links range rather than nodes position. In this way only nodes' relative position count, and absolute reference is no more required. Moreover, in this way error can be normalized to the link range, making it possible to give a percentage of error based on test ranges (1 meter of error in 2 meters of range is much more severe than 1 meter of error in 50 meters of range); further the error figures obtained after the positioning step can be compared to the ranging error before the positioning step.
To compare performance of the different approaches two error metrics are used: root-mean-square error (or RMSE) and root-mean-square-normalized error (or RMSNE) and they are calculated as follow:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(\hat{r}_{i,j} - r_i\right)^2}{N}} \tag{4.2}$$

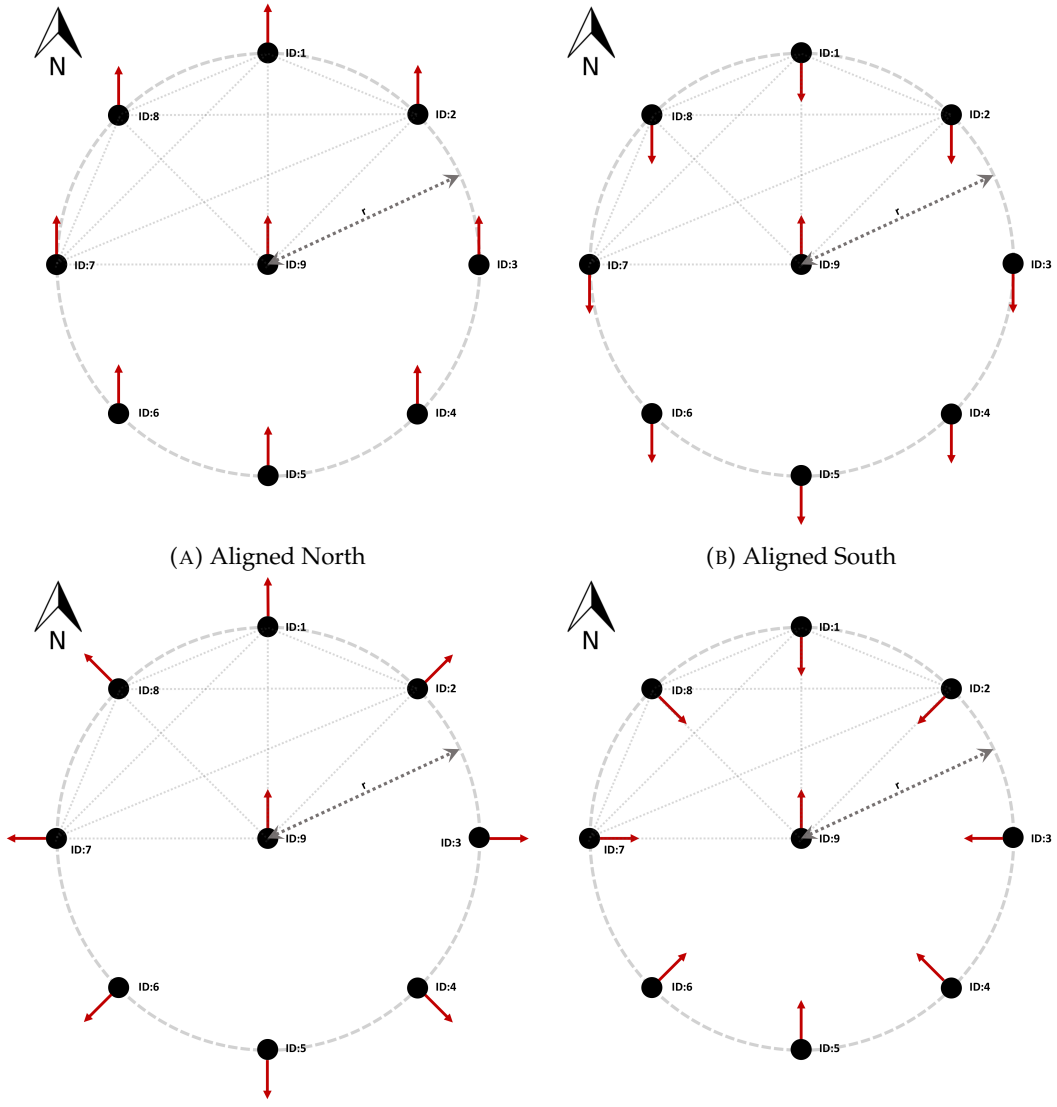
$$RMSNE = \sqrt{\frac{\sum_{i=1}^{N} \left(\frac{\hat{r}_{i,j} - r_i}{r_i}\right)^2}{N}} \tag{4.3}$$

The rationale behind these choices is that: *(i)* the mean error is not representative for range estimation since negative and positive error cancel each other, then a zero mean error could be obtained even on highly noisy data, *(ii)* large and small errors are equally weighted; these drawbacks are not found in RMSE. Further, as already pointed out with Figure 4.1 *(iii)* the error in RSSI based distance estimations tend to be proportional to the range, then providing the indication on the error as absolute value is not fair, since it should be weighted with the actual communication range.

4.4.3 Results

When only the ranging is considered, similar results are obtained with linear regression or neural network (Table 4.1):

If the positioning step is then executed (but not iterated in a loop as in the angle augmented models) and nodes' links range error is recomputed using the calculated nodes coordinates, all errors are mitigated (Table 4.2). This is an expected behaviour



(A) Aligned North

(B) Aligned South

(C) Radial external

(D) Radial internal

FIGURE 4.7: Node layout used for training the algorithms. Only a subset of links is shown with light gray, during the test the network was actually fully connected (all the nodes are in range of all the others). Nodes' absolute orientation with respect to the magnetic north is shown with red arrows.

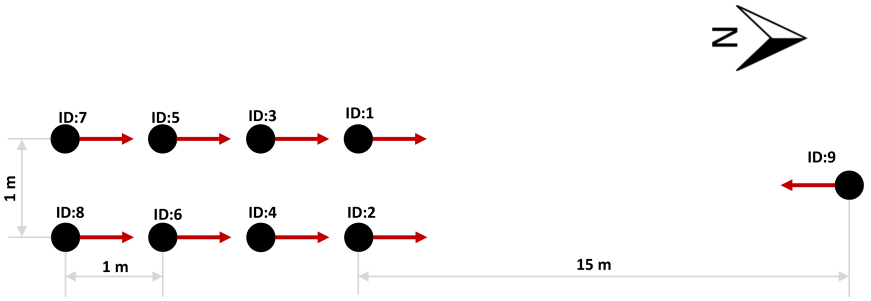


FIGURE 4.8: Node layout for the test set. It tries to mimic a compact walking bus approaching a loner child. Links are not shown, however during the test the network was fully connected (all the nodes are in range of all the others).

TABLE 4.2: Ranging error values, obtained before the positioning step, i.e. before the spring energy minimization.

Algorithm	Linear Regression	Neural Network
RMSE [m]	0.58 m	1.10 m
RMSNE [%]	27 %	33 %

as long as the ranging error is not correlated between links. However, at this point linear regression performs better than neural networks.

While if the positioning step is iterated and integrated with angles information (it is the case of the two angle augmented models: 4.3.2 and 4.3.4) two different behaviours are found: Figure 4.9 shows how the two error metrics used (RMSE and RMSNE) evolve during the Angled-LR algorithm iteration, as expected some kind of trend is visible. Even if it is not clear if the result is converging to a better solution or not (the RMSNE is reduced, but RMSE is increased by the iterations) the trend exists. However, for neural network approach (Figure 4.10) none of the two error metrics show a trend toward a better or worse solution. This seems to indicate instability of the Angled-NN algorithm: by reducing the angle correction rate to very low levels (in the order of $\gamma = 0.001$) the behaviour gets smoother. However, it is only an apparent improvement, because again there is not a stable condition. The overall errors for the algorithms that exploit angles are noted in the Table 4.3, where the neural network is missing because, as just said, no stable solution is found.

4.5 Discussion and conclusions

In general, the idea behind the proposed algorithm is that main RSSI localization errors come from known sources: one is the antenna radiation pattern and the other are signal reflections. These kinds of error are deterministic, but difficult to model in a real uncontrolled environment. To deal with them I trained propagation models

TABLE 4.3: Ranging error obtained after the positioning step is iterated several times, for Neural Network the result is not available because there is no trend and it is not possible to find a plausible stop condition (Figure 4.10).

Algorithm	Linear Regression	Neural Network
RMSE [m]	1.07 m	NA
RMSNE [%]	23 %	NA

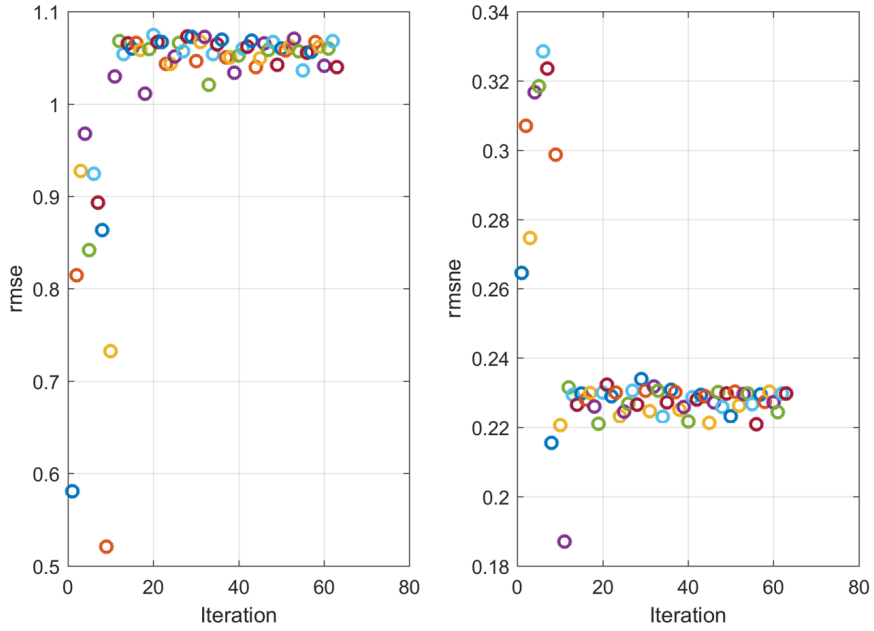


FIGURE 4.9: How error figures evolve when linear regression on angles augmented model algorithm is iterated.

that exploit angles of arrival and angles of departure (*Angled-LR* and *Angled-NN*): models have been trained with data acquired in real word experiments and tested on different datasets. RMSNE values from 23% to 36% has been found when the nodes are deployed in a near optimal configuration (nodes fixed on poles free field without obstacles). However, the effectiveness of the technique is still an open point. In fact RMSNE is reduced only by 4% from *Basic-LR* to *Angled-LR*, when the algorithm is iterated at least 15 times (Figure 4.9) and a clear stop condition for the iteration has not been detected.

Undocumented work suggests that the results are not easily replicable, because the random initialization of the node's position during algorithm initialization, seems to impact on the result. Moreover, the kind of plateau that is visible in Figure 4.9 sometimes breaks after hundreds of iterations, ending to lower or higher errors without any clear motivation. In other words, when angles were considered neither the linear regression approach nor the neural network gave stable results.

It has to be highlighted that all the tests have been conducted using Bluetooth Low Energy devices. This wireless standard has the peculiarity of being based on channel hoping, this means that the RSSI samples are not measured all on the same channel. Although at the time of writing of this thesis commercial BLE devices start to couple RSSI values with channel index, the first releases of BLE stacks lacked this information. Then, during the test described in this chapter, RSSI sample were collected but it was not possible to determine on which channel the sample have been acquired. In Chapter 6 the heavy effect of the channel on the RSSI will be shown, which justify (at least partially) the bad performance obtained in this set of experiments.

In conclusion, the ideas explored in this chapter remain valid. However, the experimental setup and the implementation of the algorithm were not adequate for the evaluation.

It can be observed, however, that the proposed angled algorithms were a sort of recurrent neural network, which is a family of algorithms used in the field of image

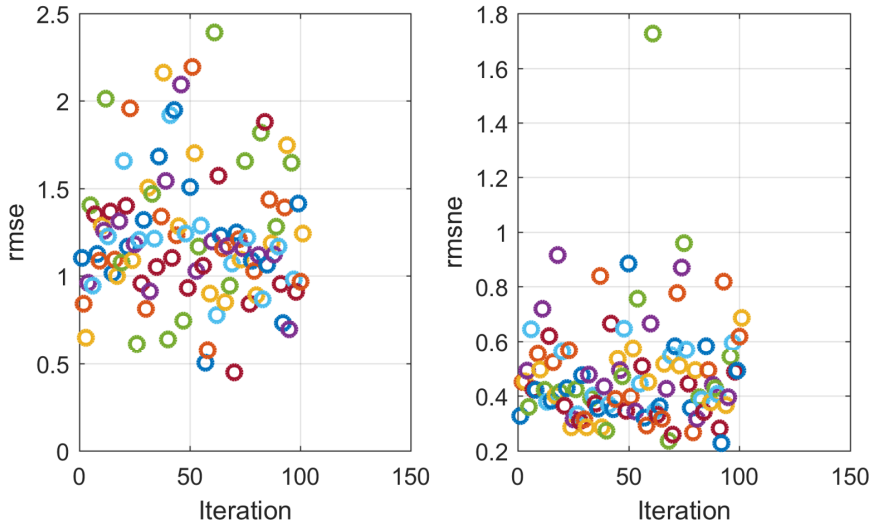


FIGURE 4.10: How error figures evolve when the neural network on decomposed network with angles algorithm is iterated.

and audio processing [30]. The theory developed in that field may be exploited for localization in a future work.

Chapter 5

Wake up radio and BLE

5.1 Introduction

Maximizing the energy efficiency is fundamental in the Internet of Things (IoT) revolution and the group management system described in this thesis is not different. As concluded in Section 3.6, the consumption of the device in the MIGnOLO system is one of the issues to face, and in this chapter, I explore an improvement that can be applied to increase the battery life and then reduce the maintenance and replacement costs.

When dealing with wireless devices, the radio interface is usually the most power hungry component of the system [78]. Almost all modern low-cost integrated wireless transceivers exhibit very similar current consumption for data transmission and for listening possible incoming communications. Hence, there is a consistent amount of power dissipated in listening mode while no data is received. To increase the node's efficiency, the radio is usually duty-cycled, and it listens only at pre-defined intervals. When a network of devices has been established and they share a common clock reference, this technique is used to effectively reduce the listening intervals of each device, while allowing for effective data exchange. This is the case of the Bluetooth Low Energy (BLE) standard: after the connection has been established, the connected devices wake up at predefined intervals to exchange data, whilst during the remaining time the radios are turned off saving a significant amount of energy [44]. However, what happens when a shared time reference is missing or when there is no continuous connection? This is the case of mobile and highly dynamic scenarios, when nodes need to continuously discover the presence or availability of other devices. In such cases, to receive data from a node or to discover new nodes in range, the radio should be in listening mode most of the time. In the worst case, the receiver will remain on continuously, hence the energy wasted can be orders of magnitude higher the one needed for the actual data transmission and the one consumed by the transmitter. There are specialized MAC protocols that optimize the radio activity exploiting transmitter and receiver activation patterns aimed at reducing the total time of activation, while keeping a short time for discovering the neighboring nodes (discovery latency) [2]. However, it is always a trade-off between contact probability or latency vs energy consumption.

To overcome these inefficiencies, the research community is developing a new class of devices, the Wake-up Radios (WuR) [51, 95]. They are radio receivers with near-zero power consumption, but without the capability of decoding complex RF modulations or manage large data payloads. These devices cannot be used for data exchange because of their low sensitivity, but they can be effectively employed to listen for incoming data transmissions and turn on the main radio only when needed. In this view, the WuR helps in avoiding useless receiver activation (idle listening or overhearing) leading to an increase in the energy efficiency of the system [78]. While

research and experimental validations have proven this approach, there are no standardized protocols for the use of WuRs and they have not been integrated in widely adopted solutions such as BLE.

In this chapter, I analyze some typical BLE beacons scenarios and architecture that are often the starting point for indoor localization based on BLE RSSI fingerprinting technique (more on this in the next chapters). I will focus on the Link Layer protocol[1] rather than on the hardware of the WuR itself, trying to propose an extension of the BLE standard that can exploit the WuR to further improve energy consumption, latency and scalability of the network. Performance of the proposed methods will be compared to the legacy ones (i.e. how the same operations are performed now, with BLE compliant devices) and results are based on a custom BLE Link Layer simulator.
Given that also legacy procedures are analyzed, an additional, and particularly relevant result of this chapter, is the analysis of the effectiveness of the collision avoidance technique implemented in the BLE standard (i.e. based on partial randomization of transmission time). The presented results will allow to draw conclusions on the scalability of the BLE standard, to give some rule for properly set the BLE advertising parameters and to provide guidelines on the integration of WuRs in such protocol stack in the context of dense IoT applications.

The reminder of this chapter is organized as follows: Section 5.2 will present the related work and Section 5.3 will describe the application scenario and the methods used for the analysis. Sections 5.4 and 5.5 will report results respectively for the BLE and for the Enhanced BLE+WuR scenarios, while Section 5.6 will discuss such results. Finally, Sections 5.7 will conclude the chapter.

5.2 Related work

The research community is very active in exploring solutions for the asynchronous wake up of wireless sensor nodes and a detailed survey on WuR hardware and protocols can be found in [94, 80, 35]. A notable approach is the use of out of band (OOB) communications exploiting non-RF medium, such as light [51] or sound [68]. For some applications this can be a very effective solution, but the type of environment and noise can limit its applicability. A further approach is the use of WuR in RF, but in a different dedicated band with respect to the main radio [28]. It is still considered OOB and this approach limits environmental problems. The main disadvantage of using RF OOB communication as the wake up signal is the need of additional hardware. Hence, there have been developed in-band solutions sharing as much as possible the main radio's hardware and focusing on small and efficient IoT devices [95, 100].

It must be noted that a great effort has been spent on WuRs for Body Area Networks (BANs) [39, 20], due to the small distances involved in BAN communications (few meters) that permit a reliable and effective use of the WuRs [82, 77, 6, 100, 110].

The only documented attempt to integrate WuR capabilities in BLE is reported in [99] and it focuses on the hardware design. The approach proposes to use backchannel modulation to make it possible for BLE compliant devices to generate the proper wake up signal the wake up radio can decode. This modulation encodes the wake up message into the advertising interval and duration (i.e. the accesses to the channel), which can be controlled through standard BLE API. Upon signal detection, the information is fed into a baseband processor that correlates the

[1] The Link Layer of Bluetooth can be seen as the equivalent to the MAC layer of 802.15.4 standard

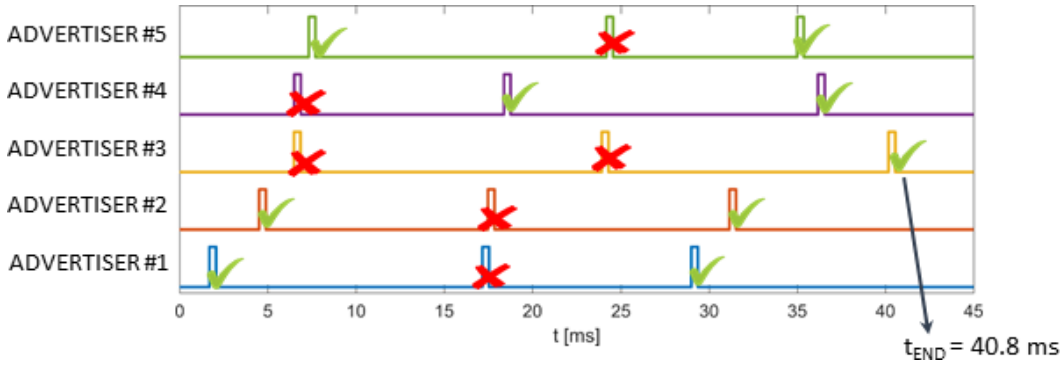


FIGURE 5.1: A graphical representation of the BLE Link Layer simulation. Tx events of each advertiser are shown, when no collision occur the packet is considered reliably sent (the behavior of the physical layer is not included in the simulator), while if events overlap the transmission is considered corrupted. The experiment is considered concluded once that at least one packet for each advertiser is sent with no collisions. The plot is only demonstrative, parameters used for the plot are not those of BLE.

energy levels with a time-based template that matches the sequence of BLE advertising packets to determine the presence of a wake up message. This CMOS based prototype achieves sensitivity of -56.5 dBm while consuming only 236 nW. Even though this implementation targets directly the BLE protocol, it will hardly work in crowded scenario, in fact, given the low symbol rate ($\sim 2\ Hz$), it is very likely that wake up message mixes in the wake up receiver with regular advertising packets sent by third node, corrupting the wake up message. Nevertheless, this solution is suitable for silent RF environment, where all the nodes are sleeping and suddenly we want to wake up a subset with a single wake up message.

The design of a low power receiver needs to be coupled with a specialized protocol to exploit all the potential of the WuR [50, 6, 59]. Unfortunately, at the time of writing there is no established standard protocol for the use of WuRs, nor support is provided for this new class of receiving devices in widely used standards such as the BLE. There have been some studies to improve existing wireless standards integrating WuR capabilities [77, 116]. In most of these cases, the wake up signal, regardless of its nature, is used as a trigger for turning on the device, hence implementing a basic low power remote trigger.

In this chapter I try to go deeper, trying to use the WuR in a more sophisticated way since it does not only turn on the nodes, instead it triggers a configuration change that optimizes the communication based on the application needs. From this point of view this is the first attempt to quantify the performance gain that a WuR can bring if strictly embedded into the BLE standard.

5.3 Scenario and methods

In this chapter I focus on the BLE beacon use case, where the BLE devices broadcast beacons providing location-based services based on the proximity to the device and enabling sensing of the environment in their proximity. Particular attention will be driven on crowded situations with hundreds of beacons within communication range. This is a typical IoT scenario where the scalability plays a fundamental role.

For the analysis, I expanded the BLE Link Layer simulator used and validated in Chapter 3.5. The simulator can emulate the BLE radio activity such as advertising events, connection requests and connection events, and it detects collisions among all the generated events. Usually BLE events are composed by multiple TX and/or RX phases on different channels, however we only consider the TX/RX phases on one advertising channel. This does not affect the simulation results, since a receiver scans only one channel at a time. Moreover, the connection packets cannot be corrupted by advertising packets since they are transmitted on different channels. Since only one connection is active at a time, collisions between two connection events are not possible. A graphical representation of the simulation is depicted in Figure 5.1, where a green 'v' marks a successful packet transmission, while a red 'x' highlights the collisions. The simulator configured in this way outputs the time to send at least one non-corrupted packet from each of the nodes, which is 40.8 *ms* in this specific simulation. To extract a general trend from this tool, the same simulation is repeated several times (in this chapter 200) randomizing the initial conditions.

The simulator instantiates the N beacons and assigns them the desired advertising interval T_{ADV}. Each beacon has a random initial offset in the range $[0; T_{ADV}]$, after which it starts broadcasting its advertising packet. Each subsequent packet is sent after $T_{ADV} + \delta$, where δ is a random delay imposed by BLE and uniformly distributed in the $[0; 10ms]$ interval.

For all the experiments the simulated setup consists of N beacons ($N = 5 \div 1000$) and all of them are supposed to run the same configuration; then they use the same advertising interval (T_{ADV}) and they always full fill the application data payload. BLE packets involved are: advertising packets that are 47 *bytes* long (with 31 *bytes* of application data), connection packets 41 *bytes* long (20 *bytes* of application data) and connection requests 44 *bytes* long (they carry only configuration data for the connection). I remark that the Bluetooth 4.2 uses 1 *Mbps* modulation, then, for instance a 47 *bytes* advertising packet is 376 μs. Moreover, I consider only one BLE central device with the receiver always on (no duty cycle is applied). The central device can perform beacon discovery (i.e. getting one advertising packet containing some minimal information about the beacon) or it can connect the beacon (one at the time) and exchange packets taking the advantages of the BLE connection. The upper layers of the protocol stack are not simulated and hence they are considered ideal without processing or memory limitations.

The central device starts the node discovery and as it receives a non-colliding advertising packet from a beacon, it marks that beacon node as *discovered*. If any packets are overlapped, a collision occurs, and the packet is discarded by the central device that restarts scanning for other beacons. If a connection must be established, as soon the beacon is discovered, the central device requests the connection to the beacon. Since advertising packets and connection requests are sent on the same channels, if they collide, the connection request is dropped because the beacon receives it corrupted. Note that, as per specification, the central device takes 6 connection intervals to realize that the connection has not been set up, then, since the minimum value for the connection interval is 7.5 ms, in the best case it takes 45 ms to discard a connection and return to scanning.

Given the number of beacons (N) and their advertising interval T_{ADV}, the simulations provide the total time needed for complete the requested operation on all the beacons.

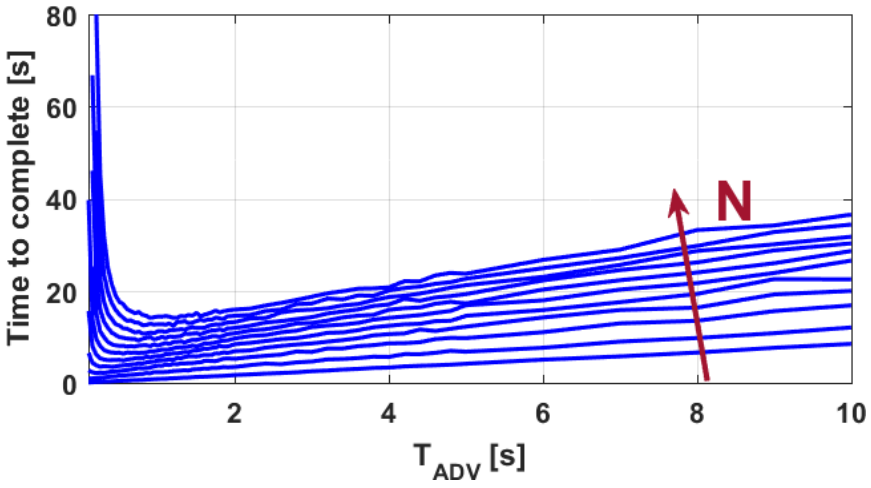


FIGURE 5.2: The average time to complete the discovery of N beacon nodes. Every trace represent the performance with different N as function of the advertise interval. N increases in the direction of the arrow and it is respectively 5, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 beacons.

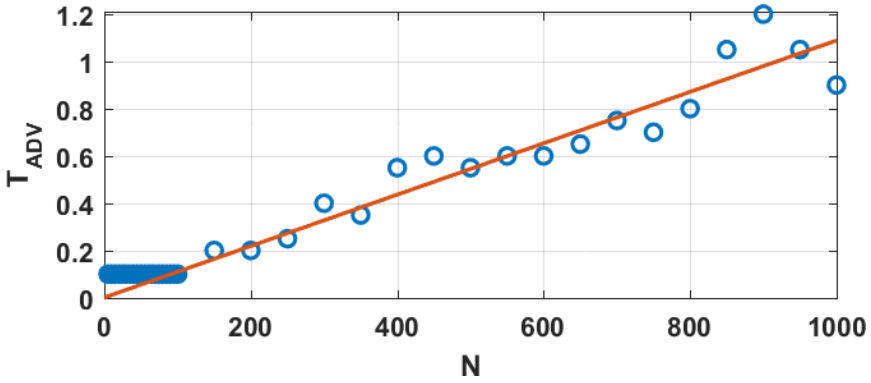


FIGURE 5.3: The optimal T_{ADV} that minimizes the time to complete the discovery of all the beacons in communication range. It represent the position of the minimum of Figure 5.2 as function of the amount of beacons N. The best fit line is also plotted.

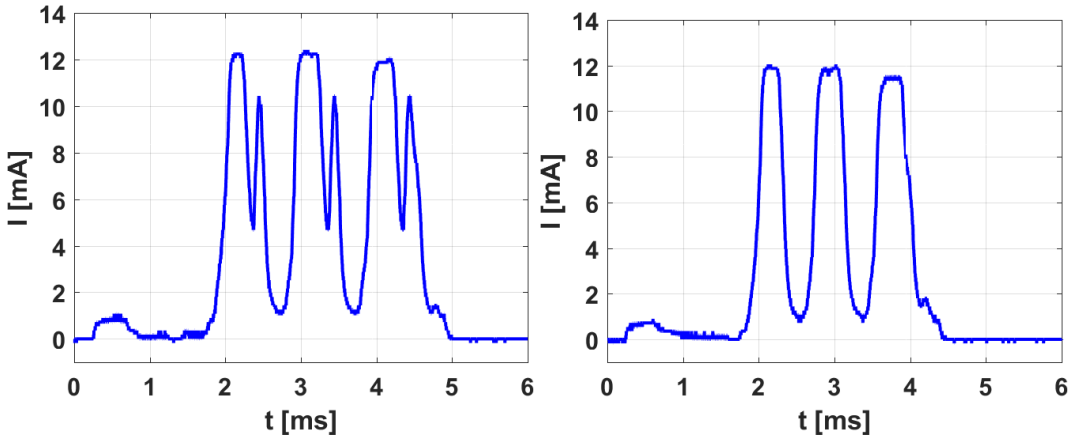
5.4 Bluetooth analysis

Firstly, I consider standard BLE devices with Bluetooth 4.2 configured as broadcasting beacons and I evaluate their behaviour in IoT applications with a high number of nodes. Given the number of beacons and their advertising interval, I used the proposed simulator to evaluate the time needed to (1) discover all the beacons, (2) discover the beacons and exchange one data packet with each of them.

Discovery time

The simulation result for the discovery time is depicted in Figure 5.2, where the horizontal axis is the advertising interval T_{ADV} (i.e. the time between two advertising packet transmissions) and the vertical axis is the time to discover all the beacons. The simulation is repeated for increasing number of beacons ($N = 5 \div 1000$).

We can observe that there are two distinct effects influencing the discovery time: the advertising interval and the packet collisions. When the collisions are few (i.e. with long T_{ADV}) the discovery time is roughly proportional with T_{ADV} and with N. That is because, if for any reason a packet is lost, the same beacon will wait one



(A) Connectable advertising. (B) Non connectable advertising.

FIGURE 5.4: Difference between connectable and non-connectable advertising in terms of current profiles. The three big spikes are the transmissions on three BLE advertising channels. In connectable advertising a small receiving window 'listen' for incoming connection requests after the transmission on each channel. The device under test is the Nordic nRF52832.

T_{ADV} before sending it again, so the longer the interval the longer the time to wait to receive the packet. Moreover, if N is high there will be more collisions, then the central device will take more time to discover all the beacons. Instead, in extremely crowded situations (i.e. with short T_{ADV}) the packets collision effect dominates. In this case, reducing T_{ADV} leads to more collisions and then the central device has less chance to receive a non-corrupted packet. This effect is the saturation of the channel.

Because of these two competing trends there is an optimal point that minimizes the discovery time, providing a guideline for setting T_{ADV} given the number of beacons. It is interesting to see that, even with 1000 beacons, this optimal point is about $T_{ADV} = 1\ s$, and for these settings we have about 53% of the packets colliding. I found that the relation between N and the optimal T_{ADV} keeps constant over the whole test range. The optimal interval is $T_{ADV} = N\ [ms]$. Using this rule of thumb for setting T_{ADV} always leads to 53% of corrupted packets. This is further illustrated in Figure 5.3, where the optimal points and their best fit is plotted.

Data exchange

To reduce the maintenance efforts by extending the battery life, the beacons use aggressive duty cycling technique: in fact, since the beacons normal operation consists of transmitting a small packet at a predefined time interval, they turn on the transmitter only for the strict amount of time needed to transmit the packet, then they return in sleep mode till the next transmission. This leads to a low duty cycle, from 1% down to 0.01% obtaining an average current consumption smaller than 10 μA.

Because of this high efficiency needs, the beacons are never listening for incoming data packets. But in almost all applications, beacons' data sometimes needs to be updated with new values. To allow this, the beacon has to be configured as connectable and it will listen for incoming connection requests during a small time window (see Figure 5.4-(a)). Once connected, a dedicated and reliable communication channel exists between the beacon node and the central device, and data can be uploaded using the connection. To be more specific, up to 20 *bytes* of data can be sent using two connection events (one for sending data, the other for receiving

the ACK - this is called *characteristic write* operation) plus one connection event for sending the disconnection request: the procedure is represented in Figure 5.5-(a).

The simulation result for the data exchange operation is reported in Figure 5.6-(a) where it is clear that the general trend is similar to the previous case, but now the values for the optimal settings of T_{ADV} are slightly different. This means that, if the beacons needs to be connected, a longer T_{ADV} should be used to avoid conflicts due to connection requests. In this case the rule of thumb to calculate the optimal advertising interval is: $T_{ADV} = 3N\ [ms]$.

5.5 BLE with WuR

As already described, in many beacon applications there is the need of uploading new data to the beacons themselves. The BLE standard permits this by connecting them (see Section 5.4), but this has two implications:

- On the beacons, the BLE standard provides for an activation of the receiver on a time window to receive connection requests (see Figure 5.4(a)).
- If the same data has to be uploaded to all beacons, doing this individually for all beacons is inefficient in particular for dense scenarios.

Here I propose the use of WuR to overcome these two implications and to improve the data uploading process to BLE beacons.

In the following the WuR is considered as a piece of hardware, mounted on all beacons, capable to generate a trigger for the BLE system-on-chip when the central device sends a specific message, such as a particular pattern in OOK modulation. The application does not require the WuR to be capable of accurate addressing (the BLE standard itself defines a reliable addressing); however since there are several WuR implementation, not all correspond to what we are looking for, in fact:

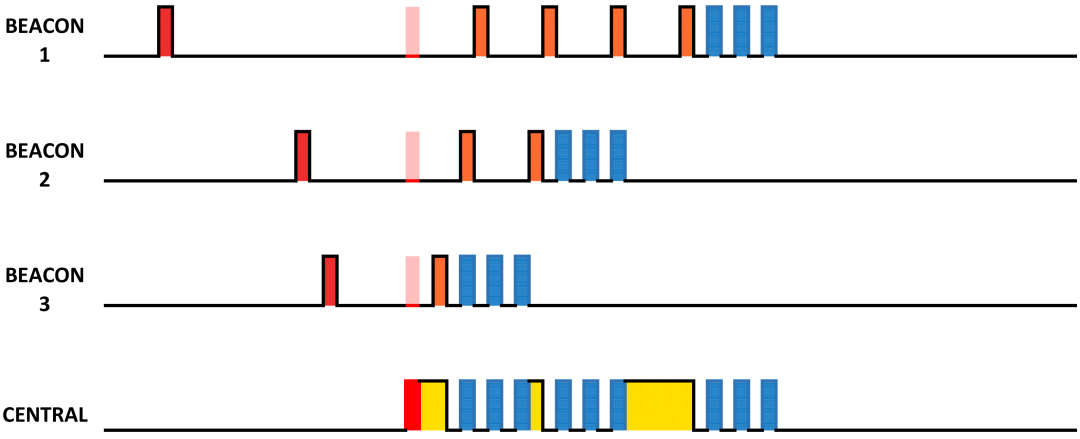
- The presence of other devices sharing the same WuR band (regardless to the approach: in band or out of band) can cause a lot of false wake-ups that frustrate the improvements brought by the WuR
- I will define two different concepts of data exchange (connection oriented and broadcast oriented) with different behaviors in response to the wake-up requests. If the beacons cannot understand the difference between the two types of request the flexibility of these solutions is low

To solve this, the wake up radio has to be smart enough to avoid false wake-ups and to distinguish between at least two different types of wake-ups requests. Anyway the design of the WuR itself is outside the scope of this chapter, and the exposed concepts can be applied regardless the type of WuR.

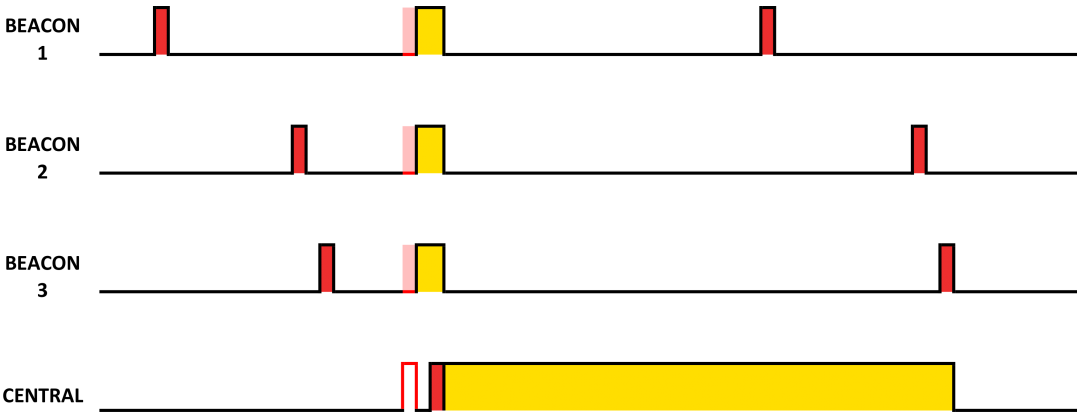
5.5.1 Connection oriented data exchange

In this case, the requested operation consists of uploading specialized data on specific beacons and because of the addressing need of this scenario the use of the BLE connection is mandatory. However, by equipping the beacon with a WuR, we can improve its efficiency and the time needed for the operation. In fact, as shown in Figure 5.4, making the beacon node connectable leads to slightly higher current consumption (30% more than non-connectable configuration) due to the receiving time window needed to receive the connection requests from the central device.

(A) Procedure to upload data on three beacons using the BLE connection. This solution is used on commercial beacons where only the BLE radio is available.



(B) Proposed procedure to upload data on three beacons equipped with a wake up radio (connection oriented data exchange).



(C) Proposed procedure to upload the same data on all three beacons equipped with a wake up radio (broadcast oriented data exchange).

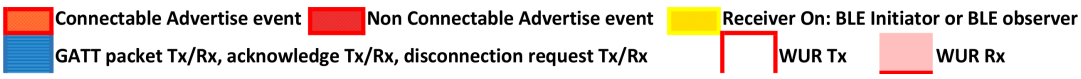
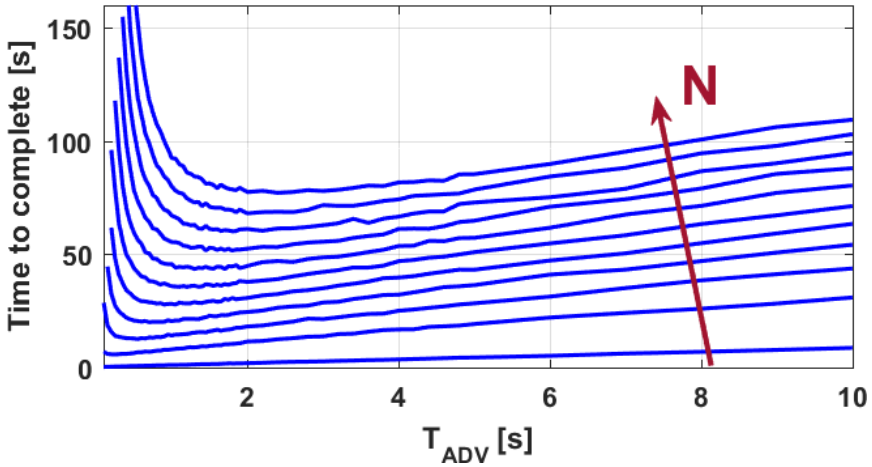
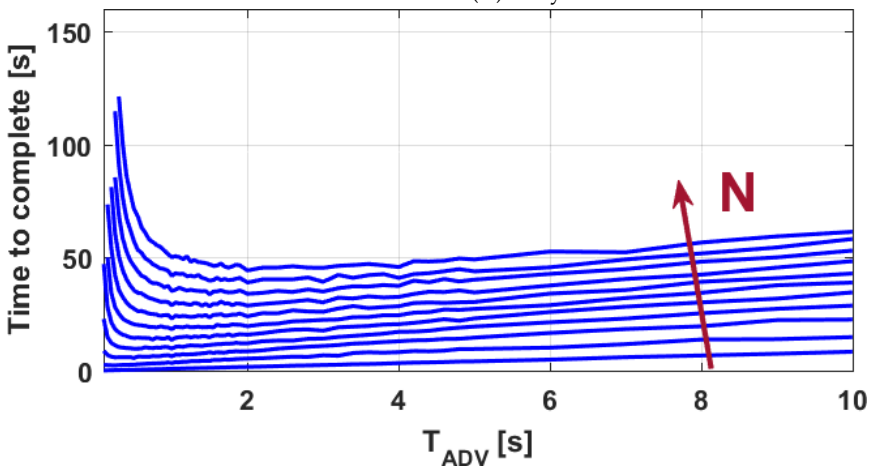


FIGURE 5.5: Description of the procedures to upload data on beacons: (a) if the beacons are not equipped with the WUR, (b) and (c) if the beacons are equipped with WUR



(A) Only BLE.



(B) BLE with WuR.

FIGURE 5.6: The average time to establish the BLE connection and to send one data packet to all N beacon. Every trace represent the performance with different N as function of the advertise interval. N increases in the direction of the arrow and it is respectively 5, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 beacons.

The proposed solution is to configure the beacons to broadcast non-connectable advertising for the regular operation. Then, the trigger from the WuR is used to revert them to connectable with a different interval T_{ADV}. Once they are connectable, the central device can connect them and send the data with *characteristic write* operations as already explained in Section 5.4; this is depicted in Figure 5.5-(b). As soon the beacon is updated and disconnected, it stops advertising for a predefined interval (from 10 s to 2 $'$, depending on N). In this way as the percentage of updated beacons grows, the level of channel congestion shrinks because less beacons keep advertising.

A feasible configuration can be to use non-connectable advertising and set T_{ADV} following the indications in 5.4 for the normal operation. Then once the beacons receive the trigger from the WuR they change to connectable advertising and set T_{ADV} following the indications in 5.4. In this way the T_{ADV} will be optimized during both phases: the normal beacon operation and the data upload process. The use of WuR reduces the energy consumption by 30% due to the use of non-connectable advertising during normal operation.

The overall results for this scenario are reported in Figure 5.6-(b), where the trend is still very similar to the cases described in the previous sections. Note that the position of minimums in Figure 5.6-(a) are very similar to the position of minimums in Figure 5.6-(b), but their values are reduced by 40%.

5.5.2 Broadcast oriented data exchange

If the same packet has to be uploaded on all the beacons, the WuR can help by doing it simultaneously, without the need of connecting all the beacons, leading to an important reduction in the time needed to complete the process, in particular with high number of beacons.

In this case, the solution implies a temporary inversion of each device role: the beacons become observers (i.e. BLE broadcast receivers) and the central device becomes an advertiser (i.e. a BLE broadcast transmitter). This permits all the beacons to receive the same broadcast packet sent by the maintenance device, without the need of establishing a connection or sending multiple identical packets; see Figure 5.5-(c).

For ensuring that all beacons received the data, as soon as the broadcast packet transmission is over, the central device starts the beacons discovery: the beacons will send the acknowledgment for instance including, in their advertising packets, a field that indicates the packet number of the last update. If the acknowledgment is not needed the whole process is shortened since the beacons discovery is not performed. Anyway, in this chapter I focus on acknowledged data transfers. Since the broadcast packet transmission last less than 1 ms, the total duration for the broadcast oriented data exchange is predominated by the discovery time needed for the acknowledgments and the transmission time can be discarded. Therefore the time needed for sending one broadcast packet and receiving all the acknowledgments is equal to the time for discovering all the beacons, as it has been already reported in Figure 5.2.

The energy efficiency gain is still 30% if non-connectable advertising are used on the beacons, while the time to complete the upload is reduced by more than 60% with respect to data exchange through the BLE connection as described in Section 5.4. This is also reflected in an efficiency gain of 60% on the central device (its consumption is roughly linear with the time it remains active). However, the focus of this chapter is the gain at beacons since the central device is supposed to have less energy constraints.

5.6 Discussion

In the last couple of years, the BLE standard is becoming very popular and in the context of the IoT the scalability is one of the key factors to consider. There are several applications requiring huge number of nodes such as goods management in big production plants, museum beacon assisted interactive guides or crowd monitoring. If this is performed with the BLE in advertising mode, the channel saturates only for collision rates over 80%. At such conditions the efficiency of the transmissions is low, only a fraction of the packets is actually received, but all the beacons are still visible.

The simulations show that there is an optimal setting for the advertise interval T_{ADV} and it depends on the number of beacons within the communication range. As rule of thumb, to achieve the shortest expected discovery latency, I measured that the value of T_{ADV} in millisecond should follow the number of beacons. Instead, if a connection has to be established, then T_{ADV} in millisecond equal to three times the number of beacons. This is caused by the transmission of the connection requests that are sent on the same channels of the advertise packets, increasing the collision rate. These rules are applicable only for rather large number of beacons (more than 100), in fact the minimum value for T_{ADV} admitted by the BLE standard in the general case is 100 ms. For reduced amounts (less than 100) a designer can freely choose the T_{ADV} without the fear of going into a saturated condition where the communication becomes impossible.

The use of the proposed WuR helps during the data upload process, while it does not affect the regular beacon operations. Since every data update is intended to happen only through the proposed protocol, the beacons can always use non-connectable advertising to save energy. I calculated the maximum gain in energy efficiency due to the use of non-connectable advertising to be 30% (based on the analysis of two widely used BLE SoCs: Nordic nRF52832 and TI CC2650). Considering the average current consumption there is a crossing point where the improvement in energy efficiency is cancelled by the consumption of the WuR. Considering $500nA$ for WuR and the nRF52832 as BLE SoC the crossing point is reached when the advertising interval is in the order of 20s: 10.24s is the maximum interval for a BLE compliant device. Then the integration of the WuR together with the proposed protocol always leads a power consumption reduction.

Nevertheless, the WuR permits to reduce the time to complete data download operations. I calculated this to decrease by 40% in the case of connection oriented communication and by 60% in the case of broadcast oriented communication. It must be highlighted that this is true only for large amount of beacons (more than 100), while in the case of few beacons the traditional approach is still preferable.

The reported design considerations come from simulations using accurate models, anyway the results needs to be confirmed by experimental tests using real hardware.

5.7 Conclusion

In this chapter I analyzed the BLE standard in advertising mode in extremely crowded environments where packet loss is remarkable because of collisions. I found that BLE is more scalable than expected; in fact, the channel saturates only with extremely

crowded situations. I gave two elementary rules that can help for setting the advertising interval of the beacons to achieve the minimum latency knowing the amount of beacons.

I also proposed the integration of a wake up receiver in the system that can help both for reducing the time to contact all the beacons and also for reducing the current consumption.

The energy saving brought by using non-connectable advertising is up to 30% and it does not reduce the BLE features, because the trigger from the wake up radio can be used to change to connectable advertising when the beacon needs to be connected. In terms of latency my technique is always better than the regular BLE in crowded situations or broadcast data dissemination.

Unfortunately, at the time of writing such WuR does not exist on the market, there are examples of discrete implementation but sensitivity and hence range is to low to be effectively applied in crowded BLE scenarios.

Chapter 6

Improving ranging accuracy with ToF over BLE

6.1 Introduction

Localization accuracy obtained using RSSI data was not satisfying (Chapter 4) in particular when the beacons are worn close to the body and out of the controlled testbed where orientations are known and nodes are fixed. This is mainly due to the scarce accuracy in the range estimation using the RSSI. However, the distance between two BLE nodes can be extracted not only from RSSI but also from Time-of-Flight (ToF). In fact, since radio signals travel with known speed through the air, if the propagation time (Time-of-Flight) is measured, the distance between the transmitter and the receiver can be estimated.
In existing BLE implementations the ToF measure is not provided and therefore a custom solution must be developed. Being based on a different physical phenomenon with respect to RSSI, it can be an alternative or a companion to RSSI as input for a localization algorithm.
Measuring ToF for distances in the order of tens of meters on BLE-like platforms can be challenging due to the clock resolution that would be required. To partially overcome this issue, the approach already proposed in literature [108, 67, 75] is to oversample the signal and than using low pass filtering (and eventually downsampling) to reduce the high measurement noise due to low clock resolution. In this work I will describe how these techniques can be efficiently ported on BLE commercial hardware, providing a general method to embed ToF measurement into BLE communication and the corresponding open source library to be used on the nordic nRF52 platform. It worth highlight that the proposed solution works in parallel with the BLE stack and does not affect the regular BLE communication.

Therefore, the contribution of this chapter is manifold: I first show how to measure ToF using a commercial out-of-the-box BLE hardware and how to embed this measurement without interfering with the existing BLE stack. Then, I collect and present measurements performed with both ToF and RSSI approach for ranging in various scenarios (i.e. indoor/outdoor and in line-of-sight/non-line-of-sight). I evaluate performance and repeatability of measurements. In section VI, I outline in particular the different behaviors of RSSI and ToF when line of sight (LoS) is occluded by the body carrying the BLE node, which is very frequent in real life group monitoring.

6.2 Background

As mentioned in the introduction, Bluetooth has been exploited both commercially [37] and in research as a suitable technology for localization.

One frequent approach is based on proximity, which requires the presence of several beacon in the surrounding. To this purpose, a building (e.g. the Gatwick airport in London in [86]) can be equipped with several Bluetooth beacons, which periodically broadcast their IDs and some other information about their status (battery level, sensor data). A smartphone application can exploit the beacons signals by activating the Bluetooth scanner; in this way, it will receive the packets sent by all the beacons that are in the proximity of the user. The packets are then decoded and used, for example, to grant the access to a location-based service to the user. The proximity is typically based on the RSSI, which can be easily recovered through the BLE API. It is measured in dBm and represents the amount of power detected by the receiver during the reception of the packet. Since the power density decreases with a known law (namely the Friis equation [102]) as the distance from the transmitter increases, it is possible to estimate if the smartphone is close to the beacon or far, just by knowing the RSSI value [54]. Therefore, by knowing the beacons' location, also the user's location can be estimated. This technique belongs to the family of techniques known as *model-based-localization*.
Under ideal conditions (line of sight, absence of reflections, accurate RSSI sampling) the RSSI-model-based-localization provides in general good results. However, in indoor environments the radio propagation is strongly affected by reflections on walls, ceiling and furnitures. This leads to achieve poor distance estimation performance [54, 10, 105] when one tries to predict the distance between the smartphone and the beacon relying only on the RSSI data and using the physical models. Moreover, since the model has a logarithmic nature, all kinds of noise/error that affect the RSSI values are reflected in the estimation error that grows linearly as the distance between devices increases.

To overcome the effects of reflections and the non-idealities of indoor propagation, *data-based* (or *RSSI-fingerprinting-based*) techniques can be applied [38, 37, 61, 55]. Instead of using a physical model for converting the RSSI to a distance and then estimate the location, these techniques rely on an off-line phase, where a number of RSSI values are collected at known positions using a scanning device (i.e. the device to be localized). Once the labelled data is collected, a machine learning algorithm is trained to output the most probable position given the previously acquired RSSI readings that acts as a fingerprint for the propagation in the environment under observation. One positive aspect of this strategy is that the reference nodes (i.e. the beacons) do not need to be at known position since the machine learning based model just requires the measure to be deterministic. Therefore, the only constraints on the beacon is that it must be at fixed position. This technique is more precise with respect to the use of a propagation model, since the trained algorithm inherently considers the effects of reflections that rarely are modelled in the physical models.

Coupled with this, there are also tracking algorithms that exploit models representing physical constraint (i.e. a human target cannot jump from one side of the room to the other in few millisecond) for filtering out bad estimations or compensate for artifacts based on the history of the positions [31, 60].

Even if the RSSI fingerprinting is widely used also in commercial products [37, 56] there are some drawbacks that limit its applicability and reliability:

- The RSSI values are strongly affected by the antenna radiation, which is potentially different on each smartphone model, and by non-line-of-sight conditions [10]
- The off-line phase for collecting RSSI at known positions requires time and is ad hoc for that place
- The off-line phase needs to be repeated if the building structure is changed (furniture, mobile walls) because also reflections change
- It cannot be applied to the class of application that rely on mobile networks [45]
- Sometime large and hard to predict errors can rise [109].

On the other side there are techniques to measure the distance travelled by a radio signal by means of the propagation time between the sender and the receiver. They are called Time-of-Fight (ToF) techniques.
Since radio signals travel at a known speed (i.e. the speed of light, approximatively 30 cm every 1 ns) it is possible to calculate the travelled distance having the travel time. As a drawback, making a time measurement in the nanosecond scale is a critical task; moreover, the clock drifts on the two sides of the link impose additional limits. For these and other reasons the main technology that is used to perform these measurements is the Ultra Wide Band (UWB) radio.

The UWB radios send information using very short burst (called chip) of electromagnetic perturbation. Because of the short duration of these chips, the reflections on walls or ceiling are received as echoes, and not as a reverberation overlaid to the direct signal, making it possible to increase the reliability of the localization through echo cancellation techniques [64]. The drawback is the need of dedicated UWB hardware that is unavailable on today's smartphones, and since market forecasts envision the Bluetooth technology to become even more pervasive than now [1] in few years, it is worth investigating the use of ToF approach as an alternative or in conjunction with RSSI methods.
It has already been shown that ToF can be exploited for ranging also on technologies similar to Bluetooth. In [108] authors use a commercial 802.15.4 compliant radio to asses two way ranging (or round trip time) performance in outdoor and indoor. When 1000 ToF measurement are averaged in outdoor the authors obtained an RMS error $RMSE = 6.7\ m$ in line-of-sight condition, that doubles ($RMSE = 15.8\ m$) when the line-of-sight is not guaranteed. They also noted that the error remains pretty constant over the whole remarkable testing range of 250 m (performance degrade only when the maximum communication range is approached). Another similar approach is described in [3], where slightly more powerful (yet commercial) hardware with 72 MHz clock shows good performance (3 m of accuracy[1] on a test range of 150 m) employing a median filter, unfortunately, the RMSE value is not provided for this work.
Another notable work in this field is reported in [67] where ToF and RSSI are compared in the context of ranging. In indoor scenario, ToF based ranging achieved better than 1 m accuracy 50 % of the time, while for RSSI based ranging 50 % of the time the error was less than 8 m. However, authors employ a custom radio (software designed radio), while our target is commercial off-the-shelf devices (COTS).

[1] The description of how *accuracy* is calculated is not given in the paper.

It is worth to highlight that, in this chapter, the aim is not to provide a final solution for indoor localization. I am instead focusing on ranging between two BLE standard nodes, measured with the approach of ToF. Today's indoor localization systems, based on BLE are in fact ignoring ToF as input to determine proximity, losing a powerful source of data for localization. This data is of course affected by noise, anyway it still provides consistent information because the environment impacts in a different manner on ToF with respect to what usually experienced with RSSI.

6.3 Time of Flight on BLE

In general, the Time-of-Flight technique can be applied to localization because the time taken by the electromagnetic wave to propagate from the transmitter to the receiver is proportional to the distance between them. In a localization scenario, once the distance between the target node and at least three reference nodes (anchors) is known, the target node can be localized with respect to the anchors on a two dimensional plane. If the target needs to be localized in a three-dimensional space one additional anchor is necessary [115].

One of the main challenges of measuring the Time-of-Flight is the needed time resolution; in fact, the propagation speed of RF signals is approximately 300000 km/s. This means that, in the ideal case, to achieve 30 cm of resolution in a single measurement, the BLE radio and the timer employed for measuring the ToF should be clocked at 1 GHz, which is almost two orders of magnitude more than what available on the typical BLE SoC. Moreover, to correctly measure the propagation time, a common clock reference should be shared between the transmitter and the receiver, otherwise at the receiver side the trigger for starting the timer would be missing. The synchronization via wireless of a common clock reference in the GHz range is unfeasible on today's low cost SoC. Moreover, the need of being compliant with the BLE further complicates it.

To face these constraints, I applied two techniques:

- **Averaging:** A distance estimation is obtained by means of averaging N consecutive ToF measurements. This helps in increasing the measurement resolution even if the reference timer used for measurement is clocked well below the GHz
- **Two-Way ToF:** I will focus on the Two-Way propagation time. It implies a bidirectional communication and it represents the time taken by the signal to propagate from node A to node B, plus the processing time on node B, plus the propagating time in the opposite direction, from B to A.

The first technique does not require a sophisticated analysis, it only needs a buffer on the SoC memory able to store N timer values. Once the values are collected they are averaged and then the distance is calculated based on the mean value.

Instead, the Two-Way Time-of-Flight measurement implies to have a deep look into how BLE standard works. As per specification, every time two BLE devices communicate over the same RF channel, the radio activity of the two nodes is coordinated to leave a non-overlapping interval of 150 us between the transmission phases of the two radios. This interval is called Inter Frame Space (IFS or T_{IFS}) [18][2] and it applies to all the Bluetooth versions starting from 4.0. Since this IFS is fixed

[2] Vol6.B.4.1

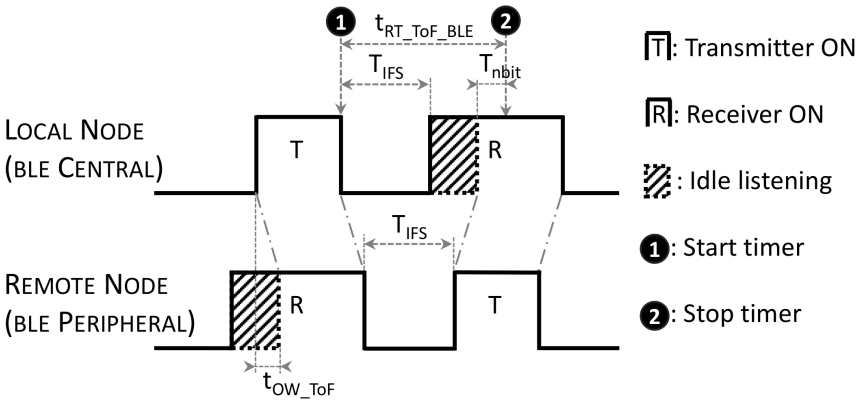


FIGURE 6.1: Connection event representation. The One-Way propagation time t_{OW_ToF} is exaggerated to highlight the details of the measurement. Figure not to scale.

by the specification, by measuring the delay between the end of transmission to the actual start of reception, it is possible to extract the information regarding the Two-Way ToF (Fig. 6.1). By halving it I calculate the One-Way ToF, which is denoted with t_{OW_ToF} in Fig. 6.1.

The *IFS* has to be respected every time the communication is bidirectional and coordinated by a central device. In the BLE world there are two main cases where this sort of coordinated communication takes place:

- at every connection event after two devices have established a BLE connection
- when a scanner device sends a scan request to an advertiser (or a beacon).

I focus on the first one, since it relies on a BLE connection and, therefore, data collection will be faster, more reliable and, since the connection data is exchanged using channel hopping over 37 BLE channels, it is more complete. The second one, instead, exploits the advertising mode of BLE, therefore it will happen only on the 3 advertising channels [18][3]. Its occurrence is less frequent and less controllable, therefore, the data collection will be slower.

6.4 Setup

6.4.1 Requirements

First, I want to test off-the-shelf hardware with its limitations. Since one of the aims of Bluetooth is to be low cost, my solution cannot rely on high-end clocks nor expensive hardware. I also want a solution that does not require the redefinition of the specifications of Bluetooth, therefore the measure will be performed on regular BLE packets, without adding any kind of proprietary transmission.

6.4.2 Hardware

I will use two nodes, one will be referred as *local* the other will be referred as *remote*. The local one will perform the measurement and will estimate its distance from the remote one, while the remote will not do anything else but behaving like any BLE

[3] Vol6.B.1.4

device (working as a BLE beacon). For the Bluetooth terminology the local nodes will take the Central role, while the remote nodes will take the Peripheral role.

As BLE radio I choose the nRF52840 by Nordic Semiconductor, which is a low cost Bluetooth compatible system on chip (SoC). It includes the radio, the processing unit (MCU) and all the typical peripherals of today's microcontrollers (timer, ADC, UART, etc). Even if the hardware and the libraries used are Bluetooth 5.0 compliant, for this chapter, I used only the features already available on the version 4.0 of the specifications.

6.4.3 Implementation

The local node will estimate its distance from the remote one by exploiting the Inter Frame Space, as explained in Sec. 6.3. Therefore, it will measure the actual time interval between the end of its transmission (denoted by ❶ in Fig. 6.1) and the reception of the first symbol transmitted by the remote node (denoted by ❷ in Fig. 6.1). The resulting time can be described with:

$$t_{TW\_ToF\_BLE} = T_{IFS} + T_{nbit} + T_D + 2\, t_{OW\_ToF} \tag{6.1}$$

Where TW in $t_{TW\_ToF\_BLE}$ stands for Two-Way. The firsts three terms of the sum are fixed and they are: $T_{IFS} = 150\ \mu s$, $T_D = 10.25\ \mu s$, $T_{nbit} = n\ \mu s$, while $2\ t_{OW\_ToF}$ is the back and forth propagation time. T_{nbit} is the on-air time for the first n *bits* of the packet, and this fixed delay is because the actual reception can be detected by the hardware only after some bit are received. Since Bluetooth 4.0 uses 1 $Mbps$ modulation, each bit is transmitted in 1 us. In this case, $n = 40$ because the MCU in use generates an event to stop the timer only after the reception of the preamble and the access address (40 *bits* in total). Instead, T_D is a further delay, which is due to the electronics (symbol detection delay, timer start delay), this value is retrieved from the datasheet of the component. For details on the BLE packet format the reader can refer to the specification document [18][4].

The time counting task is carried out using one of the digital timers included in the SoC. This timer is configured to increment the counter with a 16 MHz cadence and it is automatically started and stopped by the radio peripheral without the intervention of the MCU (this connection exploits the PPI - Programmable Peripheral Interconnect of the nRF52840). This ensures the minimum processing delay on the timer management since the radio and the timer are connected by hardware. The source code of the developed software module used for measuring the ToF can be found in [33].

For the measurement I make use of one timer that is active just for the time necessary for collecting one single ToF measurement ($t_{TW\_ToF\_BLE} \approx 200\ \mu s$). Since its maximum current consumption is 120 μA and the maximum duty cycle is 0.027[5], the measure adds, in the worst case, 3.2 μA to the average current consumption. This value is comparable with the current consumption of the SoC when it is in sleep mode. Nevertheless, the measure can be carried out only when the device is active and connected to another BLE device: in this state the average current consumption of a BLE central (again at the maximum duty cycle) is lower than 1 mA [87]. Therefore, ToF measure is negligible in terms of consumption compared to the

[4] Vol6.B.2.1.2

[5] This value comes from: $\frac{200\ \mu s}{7.5\ ms}$, where 200 μs is the time the timer is used and 7.5 ms is the minimum connection interval of the BLE standard [18, Vol6.B.4.5.1].

average current for maintaining a BLE connection, being three orders of magnitude lower.

In other words, we are piggybacking the distance measurement on a regular BLE communication, with very low overhead. Of course, if no communication is taking place between the BLE devices, the connection must be established first and some hundreds of packets have to be exchanged before having some ToF result. The average current for his procedure may consistently vary because, before establishing the connection, the device is requested to go in *scanning* to discover the remote node. Excluding extreme cases with very unfavourable settings, the overall average consumption will be below 1.5 mA, with a duration that depends on the number of packets to be collected.

6.5 Experimental procedure

For each connection event, which includes one Tx phase and one Rx phase happening on the same RF channel (Fig. 6.1), the node stores the $t_{TW\_ToF\_BLE}$, the RSSI and the carrier frequency. In fact, the Bluetooth frequency hopping scheme requires a channel change (channel hop) at every connection event.

During the tests the nodes where fixed on a stand at 40 cm above the ground and the distance between the two nodes has been varied to investigate how the ToF compares to RSSI for estimating the distance. I collected data from 1000 connection events at: 20 cm, 50 cm, 1 m, 2 m and so on with 1 m step up to 20 m.

The tests were repeated indoor and outdoor and, for both environments, I tested the Line-of-Sight (LoS) and the Non-Line-of-Sight (NLoS) conditions. The non-line-of-sight condition is created interrupting the line-of-sight between the two nodes using a man standing with its torso near (30 cm) the local node. This is for simulating the situation where the smartphone's owner stands in between the smartphone itself and the anchor.

Therefore, I collected a dataset that represents four cases: Outdoor LoS, Indoor LoS, Outdoor NLoS, Indoor NLoS. Moreover, the whole dataset has been collected twice, one is for tuning the algorithm parameters (training set) and the other is for testing the performance (test set).

Each test data, which consists of the triplet (*ToF, RSSI, Frequency*), is collected for the same 1000 connection events. If a connection events contains CRC errors, its triplet is replaced by the following one. At the end of the test, data is transferred to a PC as text file for the analysis.

6.6 Results & Discussion

6.6.1 Physical Model

Once the data has been collected and stored in the PC, I used Matlab for visualizing the results and numerically compare ToF and RSSI performance.

First, I plotted the mean and the standard deviation with respect to the distance (Fig. 6.2 shows the RSSI and ToF data for the outdoor line-of-sight test). Then, I used the mean values at each distance to estimate the range based on the acquired data, therefore I trained two physical models that are supposed to represent the behaviour of the two quantities. The models are necessary if I want to compare test data since the RSSI and ToF data is in two different measurement units (respectively dBm and ns) then I cannot directly compare the two results, thus I convert both in meters

using the aforementioned models. In this chapter I do not want to focus on the model itself, instead the purpose is to evaluate the quality of the acquired data in a fair way (that is the reason why I collected the RSSI and ToF for the same packets). In fact, regardless the used technique (fingerprinting or model-based), it is quite oblivious that the less noisy and more repeatable is the algorithm's input data, the more accurate will be the localization.

I used the log-normal model for the RSSI and a linear model for the ToF [114, 54, 75], which are described by the two equations:

$$d^{RSSI} = d_0^{RSSI} 10^{\frac{RSSI_0 - rssi}{10\alpha^{RSSI}}} \tag{6.2}$$

$$d^{ToF} = d_0^{ToF} + c^{ToF} ToF \tag{6.3}$$

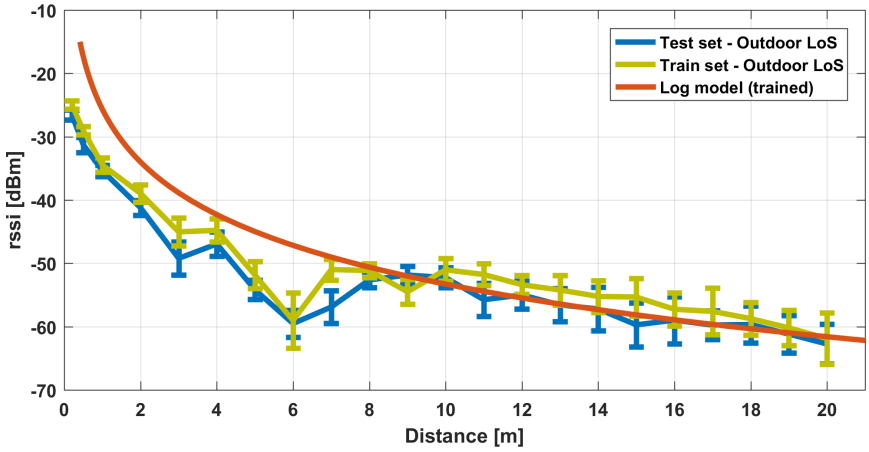
Eq: 6.2 describes the mapping from the $RSSI$ sample to distance, and Eq: 6.3 equivalently from ToF data to distance. The models parameters ($d_0^{RSSI}, RSSI_0, \alpha^{RSSI}$ for Eq: 6.2 and c^{ToF}, d_0^{ToF} for Eq: 6.3) are trained with linear regression on the training set and the trained models are visible in Fig. 6.2 as red traces. The objective of the training is to minimize the sum of the squared error over the training set, therefore I will carry on the analysis comparing results using the RMSE (Root Mean Squared Error). In this regard I want to highlight that the RMSE value tend to be larger than the mean error value. In fact, positive and negative errors will compensate each other when the mean is calculated, while they won't compensate in the RMSE calculation. For this I believe the RMSE is more relevant than mean error for evaluating ranging performance.

From Fig. 6.2 it is clear that the ToF measurements have a large standard deviation; this was expected and it is because the timer runs at 16 MHz, which means that for every clock cycle the radio waves travels approximately for 18.8 m. Since I am measuring the Two-Way ToF the distance to travel is doubled (back and forth) then every timer tick represent 9.4 m of distance travelled by the radio waves. This 9.4 m is hence the granularity of the distance estimation based on a single ToF sample. Anyway, since the two devices are not clock-cycle synchronized, the drift and the phase noise of the clocks adds a measurement noise that can be considered uncorrelated among different ToF samples. Therefore, the clock noise act as dithering noise, which permits to increase the resolution of result over 9.4 m if multiple samples are averaged.

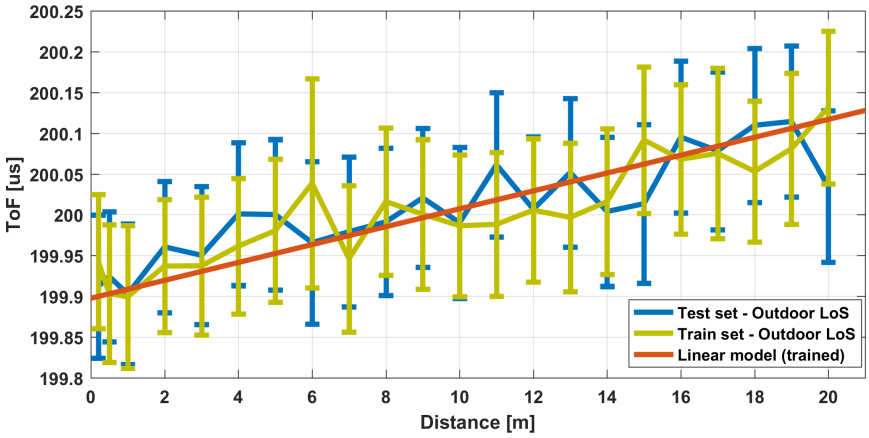
Another important observation is that the slope of linear model used for ToF (red line in Fig. 6.2-(b)) should be the inverse of the speed-of-light per definition, but actually it is the inverse of a smaller value (approximately 180000 $\frac{km}{s}$). I still miss a satisfactory and complete explanation for this, but I believe it is a second order effect of reflections.

6.6.2 Models performance

With the purpose of investigating how the training condition impacts on the estimation error, I trained a different model for each training set (indoor/outdoor, LoS/NLoS) and then I tested them all on all test sets (note that the training and test sets have been collected on different days). Moreover, I created an outdoor generic model, which is trained with the merge of the outdoor LoS and the outdoor NLoS datasets, and I did the same also for the indoor condition. Recapping, I trained six different models (outdoor LoS, indoor LoS, outdoor NLoS, indoor NLoS, outdoor generic, indoor generic) and I tested them in the four cases (outdoor LoS, indoor LoS, outdoor NLoS, indoor NLoS). The resulting test error are reported in Table 6.1



(A) RSSI data.



(B) ToF data.

FIGURE 6.2: Acquired data: RSSI values (a) and Time of flight (b), plotted with respect to nodes distance in the Outdoor LoS condition. The mean value over 1000 ToF samples is reported, error bars are one standard deviation high. The two colors depict the train and test dataset. The fitted model is in red.

RMSE		TRAIN CONDITION											
		outdoor LoS		indoor LoS		outdoor NLoS		indoor NLoS		outdoor generic		indoor generic generic	
		RSSI	ToF	RSSI	ToF	RSSI	ToF	RSSI	ToF	RSSI	ToF	RSSI	ToF
TEST CONDITION	outdoor LoS	mean: -1.77 rmse: **3.26**	mean: -0.46 rmse: **3.02**	mean: -4.07 rmse: 4.97	mean: **1.55** rmse: 3.50	mean: 6.17 rmse: 7.65	mean: 3.66 rmse: 4.90	mean: 4.11 rmse: 5.30	mean: 5.19 rmse: 6.44	mean: 3.02 rmse: 4.51	mean: **1.87** rmse: **3.55**	mean: **-1.11** rmse: **4.23**	mean: 4.23 rmse: 5.39
	indoor LoS	mean: **0.57** rmse: 3.79	mean: **0.23** rmse: 3.35	mean: **-1.69** rmse: **4.01**	mean: 2.06 rmse: **3.48**	mean: 7.04 rmse: 8.80	mean: 3.89 rmse: 4.78	mean: 5.94 rmse: 7.58	mean: 5.37 rmse: 6.23	mean: 4.57 rmse: 6.43	mean: 2.20 rmse: 3.76	mean: 3.09 rmse: 4.85	mean: 4.39 rmse: 5.22
	outdoor NLoS	mean: -20.20 rmse: 20.92	mean: -6.00 rmse: 7.40	mean: -21.97 rmse: 22.49	mean: -3.01 rmse: 5.18	mean: **-1.03** rmse: **3.25**	mean: **-0.99** rmse: 4.31	mean: -14.37 rmse: 16.19	mean: 1.98 rmse: 4.76	mean: -9.62 rmse: 10.24	mean: -3.95 rmse: 5.89	mean: -50.84 rmse: 58.91	mean: **-0.32** rmse: 4.20
	indoor NLoS	mean: -6.49 rmse: 7.28	mean: -6.79 rmse: 8.70	mean: -9.09 rmse: 9.67	mean: -3.67 rmse: 5.45	mean: 4.79 rmse: 6.69	mean: -1.60 rmse: **4.09**	mean: **0.91** rmse: **3.64**	mean: **1.53** rmse: **2.89**	mean: **0.16** rmse: **3.85**	mean: -4.72 rmse: 7.21	mean: -8.83 rmse: 10.22	mean: -0.89 rmse: **3.66**
	average	mean: 1.30 rmse: 8.81	mean: 1.42 rmse: 5.62	mean: 1.20 rmse: 10.28	mean: 1.86 rmse: 4.40	mean: 5.31 rmse: 6.60	mean: 2.77 rmse: 4.52	mean: 3.60 rmse: 8.18	mean: 4.17 rmse: 5.08	mean: 2.88 rmse: 6.26	mean: 1.99 rmse: 5.10	mean: 3.09 rmse: 19.55	mean: 3.09 rmse: 4.62

TABLE 6.1: Mean and RMSE (root mean squared error) values obtained training and testing the models with various combinations of the data sets. The minimum Mean and RMSE of RSSI and ToF in each column is highlighted in bold. Unit is m.

where the mean error and RMSE (Root Mean Squared Error) is reported for all combinations of train and test.

The error values in this chapter must be considered in the context of ranging and not of localization. In fact, for localizing a target on a plane, the distances from at least three known points are needed. If the noise in the three distance estimations is not correlated, the localization error will be reduced with respect to the ranging error from one single beacon.

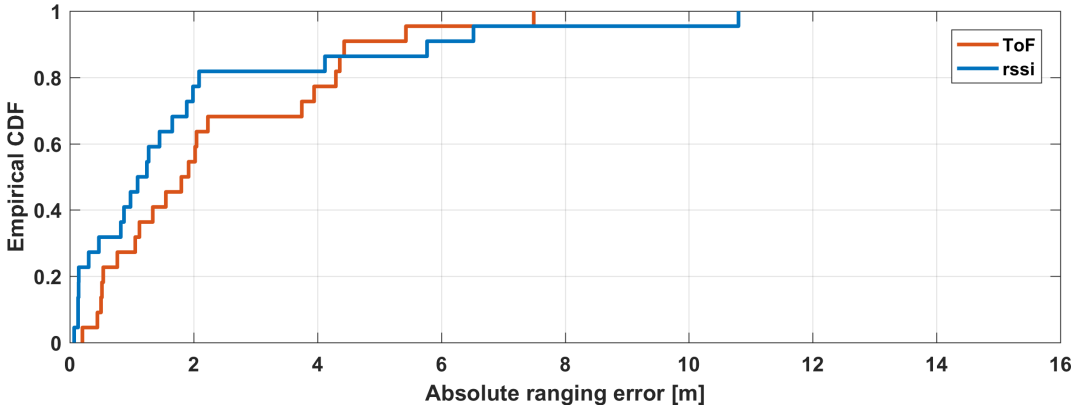
There are some other important observations regarding Table 6.1:

- (almost) all models have their minimum error when the test data has been collected in the same conditions of train data; this was expected, and it means that different datasets collected with the same conditions are similar (i.e. the experiment is repeatable)
- the RMS error obtained using ToF is always less than the one obtained with RSSI. This is highlighted also by the mean values (bottom line of Table 6.1) that represent how a specific model behaves in the general case
- the impact of the LoS or NLoS condition is more severe on the RSSI data
- in some cases, the RSSI error tends to diverge (almost up to 60 m) while the ToF error remains under control

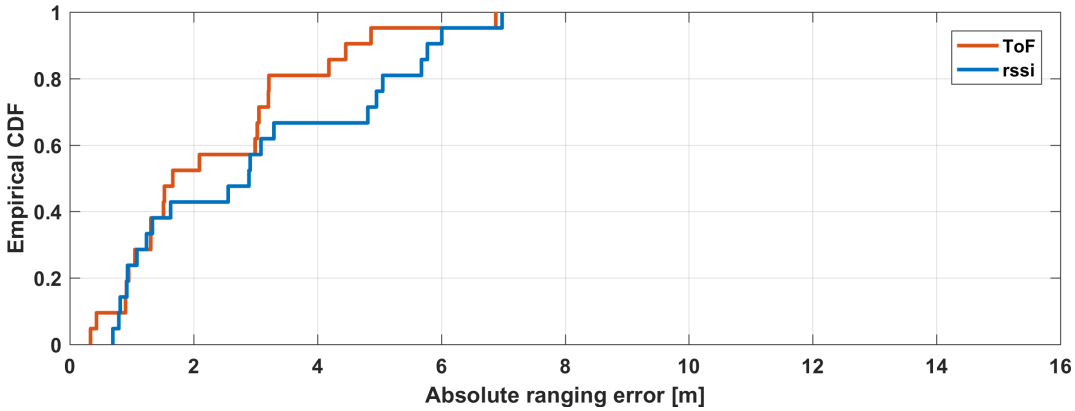
Very similar considerations can be derived from the CDF (Cumulative Distribution Function) of the (absolute) ranging error that is show in Figure 6.3. In favorable situations the CDF error shows similar behaviour for ToF and RSSI (the error remains below 2 m in the 50 % of the cases), while in unfavorable conditions both ranging techniques degrade in accuracy, however the RSSI is almost 4 meters less accurate than ToF.

6.6.3 Dependency on the number of packets

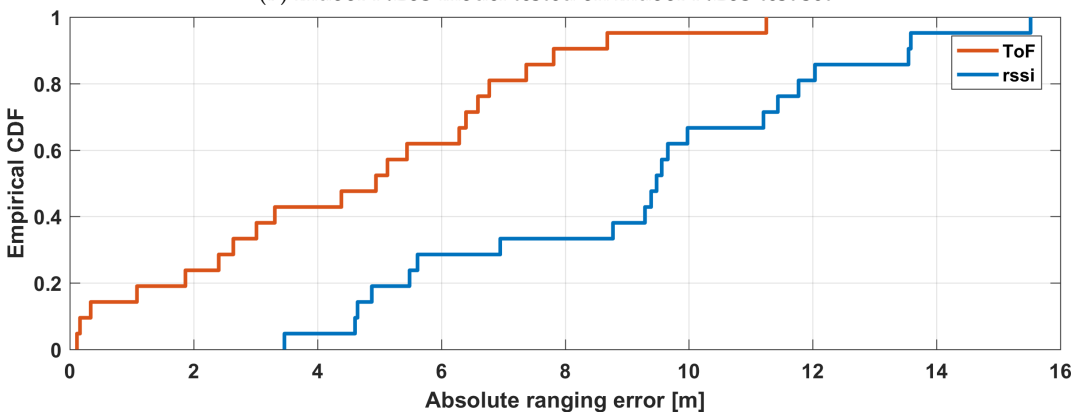
As already mentioned, the resolution is increased by the means of averaging data over 1000 ToF samples, but this implies that an application would have a response latency that is at least the time needed to collect all the samples. Since the node collects one sample per connection event and since, as per BLE specification, the minimum time between connection events is 7.5 ms, the minimum expected latency is 7.5 s, which may be too high for some application. There are workarounds for



(A) Outdoor LoS model tested on Outdoor LoS test set

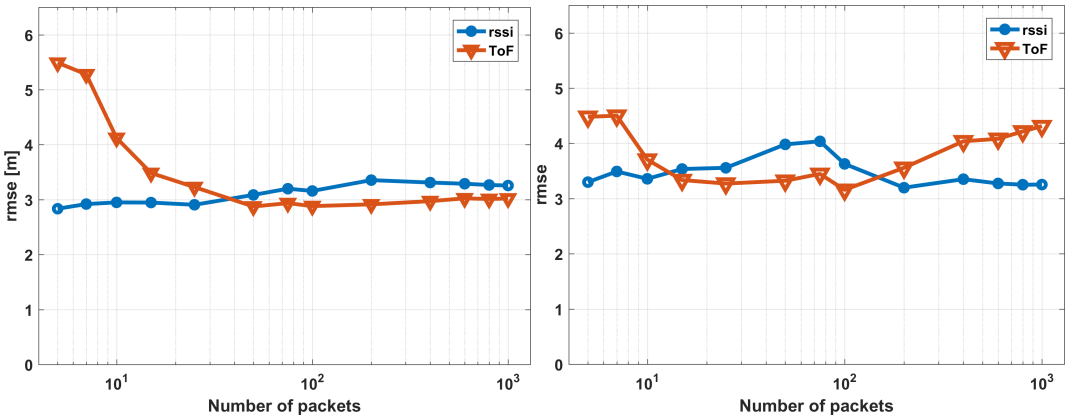


(B) Indoor NLoS model tested on Indoor NLoS test set



(C) Indoor LoS model tested on Indoor NLoS test set

FIGURE 6.3: Cumulative Distribution Function of absolute ranging error for three conbinations of train/test set.



(A) Outdoor LoS (B) Outdoor NLoS

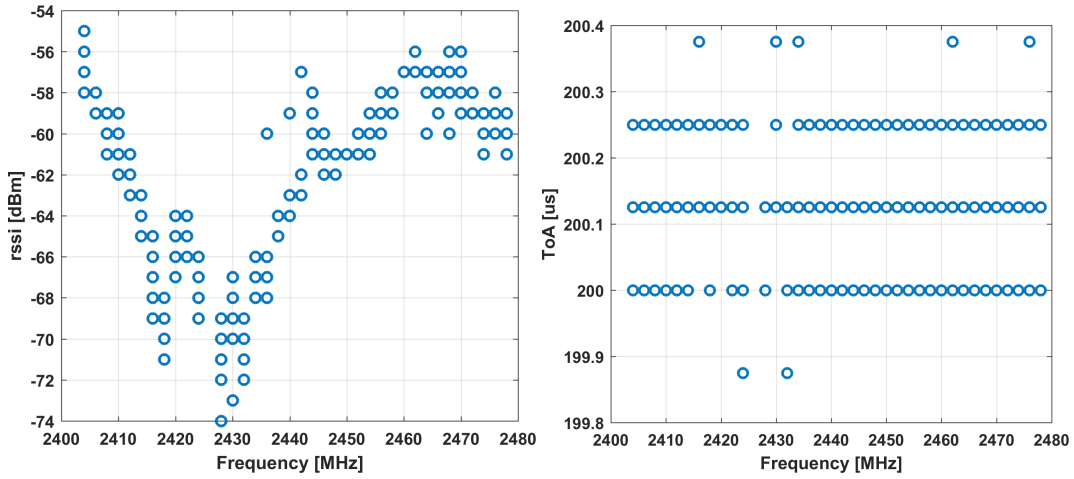
FIGURE 6.4: RMSE (Root mean squared error) in function of the number of packets averaged for the outdoor LoS (a) and NLoS (b). Note that X axis is log scale.

reducing the overall latency without violating the 7.5 *ms*. Anyway, it is important to analyze how the error changes as the number of samples for the average is reduced.

For doing this I repeated the model testing, using the average of only the first N samples at each position for calculating the distance. The models are the same as previous experiments, then there is no need to train them again. I used this technique instead of random sampling over the 1000 samples for obtaining coherent results, since the frequency hopping pattern built in the BLE cannot be randomized. Two representative results are reported in Fig. 6.4, which shows how the error changes with respect to the number of averaged samples. Both the errors for ToF and RSSI are reported; the data of the two figures refers to Outdoor LoS and NLoS conditions.

I repeated this analysis for all the cases I collected (the 24 combinations in Table 6.1) and the behaviour is not uniform. In the majority of the cases (roughly 70 %) I obtained a benefit of at least 1 m on the RMSE after averaging, anyway for some case like outdoor NLoS condition reported in Fig. 6.4-(b), the RMSE is pretty constant after the average, no matter how many packets are considered. For the test cases where the average is effective, I did not have relevant decrease of the error over 100 samples averaged. In an optimal, and yet realistic, configuration (which is sampling period of 7.5 *ms*, see above) it takes 0.75 s to fill the average window. This data latency can be considered more than acceptable for most of human navigation and localization applications.

Moreover, the RSSI performance is almost unaffected by the amount of samples to average; this is explained by the measurement granularity. As said the single measurement of the ToF has a granularity of $\frac{1}{16\ MHz} = 62.5\ ns$, which is converted in 9.4 m. While the RSSI granularity is 1 dB; this is less straight forward to convert to meters, but using the model described by Eq.6.3, within the testing range (0.2 m-20 m), the estimation granularity spans approximatively from 7.5 cm to 1.7 m. Now it is clear that, even in the worst case, the single sample granularity of RSSI is smaller than ToF; therefore, adding samples to average is an effective technique for ToF data, while it is less effective for the RSSI data. These considerations can be taken into account when the localization latency is a critical aspect of the application.



(A) RSSI (B) ToF

FIGURE 6.5: Effect of channel hopping on raw (not averaged) RSSI and ToF data. Each figure contains 1000 data points, then each circle may represent multiple overlapped points.

6.6.4 Dependency on the channel

The last observation I want to bring to the reader is the effect of the channel hopping on the acquired data. In Fig. 6.5 the raw samples collected with the two nodes at 20 m of distance in outdoor LoS conditions are plotted with respect to the used channel. It is immediate to see (Fig. 6.5-(a)) that the RSSI data is strongly affected by the channel. These results where expected from the two-ray model of propagation [43, 105], since during the outdoor test I had the reflections of the ground and also of a neighboring building. The reported figure has been chosen as representative but at the other distances the channel effect has a different behaviour, which is dependent on the surrounding environment. At the same time the ToF shows more constant results (Fig. 6.5-(b)).

From this result, it is clear that using the RSSI for calculating the distance between two nodes without considering which channel each sample refers to is limiting. Modelling the effects of reflections at different frequencies and in the general case (without prior information on the environment) is a complex task. In contrast, including the channel information in a machine learning algorithm that automatically takes it into account does not add a relevant overhead. In particular, for the fingerprinting techniques giving more data to the algorithm is likely to give better estimation result. Unfortunately, the channel number information is not exposed through the standard Bluetooth API of the most popular mobile devices (Android/iOs smartphones) and on many embedded platforms, then it is much more challenging for the developer to get that information[6].

6.6.5 Repeatability of the test

Although it is clear from the results that the collected ToF data, even if noisy, can be used as an indicator of distance even for narrow band protocols such as BLE, its usefulness may be questionable if the measure isn't repeatable over different beacon

[6] These BLE technologies are continuously evolving, and recently (1 year after the study reported in this chapter) major BLE device manufacturers started coupling the channel index information to each RSSI sample acquired. This further confirms the observation just discussed.

models or over different specimens of the same beacon. With this regard I did a preliminary evaluation testing both the nRF52840 and the nRF52832 with two different SDK (respectively softdevice 140 v5.0.0-2alpha and softdevice 132 v5.0.0) and also another SoC (Texas Instruments cc2650, with BLE stack v2.2) used as BLE beacon, the test has been repeated on multiple specimens of the selected SoC.

What I found is that choosing a different BLE SoC manufacturer has potentially the same impact to changing the SDK, and the effect is a small change in the offset of the measure. This offset should be nominally $T_{IFS} + T_{nbit} + T_D = 199.65\ us$ that is Eq: 6.1 when the distance between the two devices is zero. As visible from the figures, obtained results are slightly different (the measured offset is few hundreds of nanosecond longer) and I justify this with the hardware/software implementation of the BLE stack. In fact, we are dealing with time resolutions that are in the same order of magnitude of clock interval of the SoC, therefore the hardware/software architecture plays a fundamental role for this kind of measure. Unfortunately, this can hardly be calculated analytically since most of the BLE SoC manufacturers give part of the SDK as pre-compiled binaries, then source code cannot be analyzed. Nevertheless the results are stable among specimens of the same device/SDK configuration, then it is reasonable to speculate that the offset can be a constant value statically associated with a specific configuration of SDK-SoC model and broadcasted as part of advertising packet payload, similarly to what already happen for the *calibrated Tx power* field in the Eddystone UID packet [36].

This evaluation is preliminary and is only qualitative, anyway it has been useful to verify that the experiment can be brought in the real world outside the small, controlled and homogeneous testbed I used.

6.7 Conclusion

In this work, I investigated if RSSI is the only indicator that can be used for localization based on Bluetooth Low Energy or if ToF can be a viable additional one. Using low-cost off-the-self hardware, I have shown that with BLE the ToF can be measured if a sufficient number of packets is averaged. Obtained results are satisfying, first of all because this seems to be one of the first attempt to measure the ToF over the BLE at low level (i.e. excluding any overhead due to the stack processing), moreover results are in line with similar works employing less constrained hardware and particularly less constrained protocols. Even though the communication range in my experiments is relatively short (0.2 m to 20 m)[7], [108] argued that error on ToF remains constant in most of the communication range. Then the ToF ranging error reported in this chapter are similar to what found in [108], for instance in outdoor LoS scenario they obtained $RMSE = 6.7\ m$ while in this chapter, with very similar conditions, I halved it obtaining $RMSE = 3.02\ m$. When the LoS is occluded, the RMSE roughly doubles in both cases.
My results are instead worse (approximately by a factor of two) than the other similar work [67]: the reason is probably due to the limitation and constraints imposed by the hardware (this work employs commercial hardware, the other a custom software defined radio) and by the BLE standard. Nevertheless, the proposed approach, being based on BLE, has much more potential. In fact, at the time of writing, the pervasiveness of Bluetooth connectivity in commercial products is several orders of magnitude higher than any other standard for short range low power communication such as Zigbee, Thread or anything else based on 802.15.4 MAC.

[7] This choice was made to keep the same range for indoor and outdoor scenarios.

I calculated the energy overhead due to the ToF measurement finding that in case the BLE connection is already established (e.g. for data transfer) the overhead for measuring the ToF is negligible. While, if the connection has to be established for the purpose of localization the average current consumption is higher (1 mA to 1.5 mA), which corresponds to the consumption of a BLE central device sending and receiving empty packets.

Experiments showed that ToF and RSSI have comparable performance for distance estimation in the range $0 - 20$ m; however, ToF is slightly better in terms of RMSE. ToF performance starts to degrade when the average is calculated on windows below 100 samples. While for RSSI, averaging over multiple samples does not have a particular effect, making it a good choice when the latency is a key point.

As expected, LoS/NLoS condition had an impact on distance estimation with RSSI; this should be considered when the device estimating its position is a Smartphone. In such case, the possible proximity to owner's body makes difficult to predict if beacons and smartphone are in LoS. At the same time, the LoS/NLoS has a lighter effect on distance estimation when the ToF is used for ranging.

Another kind of data that is often discarded for the RSSI based localization is the channel (or the career frequency). The Bluetooth standard is based on channel hopping for mitigating the effects of interference and fading, and in this work I have shown the high impact of channels on raw RSSI values. Therefore, to increase algorithms accuracy RSSI data should be always coupled with the channel where available. This observation may be considered by Bluetooth radio manufacturers or by Bluetooth SIG when defining APIs used for controlling the radio.

Results reported in this chapter show that ToF alone is not fully reliable for localization; however, it furnishes valuable data at low price, useful to increase the accuracy of a model-based or fingerprint-based algorithm.

Recently, the new version of Bluetooth specification has been released. Version 5.0 includes some interesting features, such as the long range support (up to 1 km in LoS), which is achieved using coded modulations. With such a kind of long range wireless communication, the RSSI based localization will lose reliability because it is well known that the ranging accuracy of RSSI degrades as the range increases [96] (refer also to Section 4.3, in particular to Figure 4.1), therefore I expect ToF measurement on Bluetooth 5.0 to gain interest soon.

Chapter 7

From raw data to position tracking

7.1 Introduction

Previously in this thesis, the Time-of-Flight (ToF) measurement technique has been applied to the ranging problem on Bluetooth Low Energy showing that RSSI is not the only metric that can be exploited for localization in a BLE network. Because of hardware constraints, ToF data is very noisy, then some filtering technique must be applied before using it. In the previous chapter, a simple average window has been applied with the purpose of comparing RSSI and ToF performance. Now that we know that ToF and RSSI on BLE have similar performance in the context of ranging, I want to do a step forward providing a localization framework that exploits both sources of data.

The technique proposed in this chapter stems from the general idea described in [75] and [40] (i.e. RSSI and Time-of-Flight (ToF) data fusion), but it relies on BLE technology and on a different methodology. Various examples of data fusion algorithms for indoor localization based on a Bayesian approach already exist [41]. However, the solution proposed in this chapter is computationally simple and it is tailored for BLE implementation. In general, unlike the case of Ultra-Wide Band (UWB) where ToF can be generally measured with high accuracy [4], the ToF measurement of BLE messages (as well as of other narrow-band wireless protocols) is subject to large fluctuations due to: the jitter affecting message detection on the receiver side, the limited resolution of the available clocks to measure time intervals, or a combination thereof. However, such uncertainty contributions exhibit approximately a white spectral power density. Therefore, they can be effectively reduced through averaging [75, 40, 44]. Moreover, their relative impact on range measurements tends to decrease with distance, whereas, on the contrary, the uncertainty of RSSI measurements tends to grow with distance. This observation suggests that RSSI and ToF data may play a complementary role for wireless positioning, which justifies the approach described in this chapter.

In the following, first, in Section 7.2, the ranging and positioning steps of the proposed localization algorithm are explained. Some meaningful experimental results of both ranging and planar positioning are presented in Section 7.3. Finally, Section 7.4 draws the conclusions and propose future directions.

7.2 Localization Algorithm

In contrast to fingerprint techniques, which strongly rely on acquired data, there are model-based techniques that are more related to radio signal propagation models. Regardless of the quantity to be measured (e.g., RSSI, ToF or time difference of arrival), model-based localization is typically a twofold process consisting of a *ranging*

step followed by a *positioning step*. In the *ranging step*, the target node to be localized performs some measurements to obtain the estimates of its distance from a number of reference nodes (anchors), while the *positioning step* transforms the distance estimates into coordinates in a given reference frame. In the following sections, the algorithm developed to implement both steps is described.

7.2.1 Ranging Step

In general, the *ranging step* requires to measure a physical quantity that is a monotonic function of the distance from one of the anchors. Given the wireless nature of the problem, the most popular quantities to be measured are: *signal power* and *propagation time*. As already widely discussed in the thesis, the former quantity (that is natively measured by the RSSI circuitry available in any wireless chip) tends to decrease with the inverse of the square of the distance from the transmitter. However, in indoor environments it may exhibit a non-monotonic behavior as well as a path loss no longer proportional to the square of the distance, due to both multi-path fading phenomena and the presence of obstacles. On the other hand, the propagation time grows linearly with the distance between transmitter and receiver, as the signal propagation speed is constant and depends on the type of signal employed (e.g. radio or acoustic signals). Of course, in the case of radio (e.g. BLE) signals, the signal propagation speed coincides with the speed of light. The main problem of propagation time measurements is that typically the time intervals to be measured over short distances are very small (i.e. in the order of tens of nanoseconds). Therefore, high-resolution time interval measurements are needed. Moreover, the propagation time intervals are inherently subject to large fluctuations caused by the circuitry used to send and to receive the radio messages. A solution to mitigate this problem is to use acoustic signals, since their propagation speed is much smaller. However, this approach as well as other ranging techniques requiring dedicated hardware (e.g. for Doppler shift measurements) are not suitable if standard commercial off-the-shelf (COTS) BLE components have to be employed. For this reason, the ranging approach adopted in this chapter relies on the fusion of the distance values obtained from both RSSI data and the ToF of BLE messages, since such quantities can be measured using standard Bluetooth platforms without additional hardware.

As known, the distance values based on RSSI result simply from [92]

$$d_{RSSI}(s) = d_0 10^{\frac{s_0 - s}{10\alpha}}, \tag{7.1}$$

where s is the RSSI value, s_0 is the RSSI at a reference distance d_0 and α is referred to as the path loss coefficient. In ideal conditions $\alpha = 2$, whereas in indoor environments it depends on the surrounding ambient and hence should be experimentally evaluated. For what concerns the distance based on the propagation time of BLE messages, in this work it is estimated from the round trip time (RTT) of a pair of messages, i.e.

$$d_{RTT}(t_{RTT}) = c\frac{t_{RTT} - t_{off}}{2}, \tag{7.2}$$

where c is the speed of light, t_{RTT} is the measured round-trip-time and t_{off} is an offset including the processing latency on the responding node (namely the anchor) as well as the timestamping delays on both the transmitter and the receiver side. The ranging data given by (7.1) and (7.2) are both used in the update step of a Kalman

Filter (KF), which is described in details below. In particular, the distance measurement value $r_{m,i}(k)$ from anchor i at time $k\Delta_t$ (where Δ_t is the constant sampling time) are based on (7.2) and result from

$$r_{m,i}(k) = d_{RTT}(k) = r_i(k) + \xi_r(k), \tag{7.3}$$

where $\xi_r(k) \sim \mathcal{N}(0, \sigma_r^2(k))$ is the ranging measurement uncertainty. The noise process is assumed to be non-stationary, with variance approximated with the sample variance in the time instants $k - i_v, \ldots, k$, with i_v being the sample window.

Even though the RSSI distance measurements (7.1) could be injected directly in the update step of the KF, in this here they will be used to estimate the target's velocity $v_i(k) = \dot{r}_i(k)$ in the direction towards anchor i at time $k\Delta_t$, i.e.

$$v_{m,i}(k) = \frac{d_{RSSI}(k) - d_{RSSI}(k-1)}{\Delta_t} = v_i(k) + \xi_v(k), \tag{7.4}$$

where $\xi_v(k) \sim \mathcal{N}(0, \sigma_v^2(k))$ is the velocity measurement noise. As for ranging, the noise process is not stationary and it is given by

$$\sigma_v^2(k) = 2\left(\left.\frac{\partial d_{RSSI}(s)}{\partial s}\right|_{s=s_m} \frac{\sigma_{RSSI}}{\Delta_t}\right)^2 \tag{7.5}$$

with σ_{RSSI} is the standard deviation of (7.1) obtained from experimental data and s_m is the measured RSSI. The rationale of this choice is that the accuracy of the KF using velocity data is globally better, which we argue is related to the system observability. In fact, using ranging data only leads to a dynamic observability of the system, whereas using both distance and velocity values leads to a static observable system: the presence of a relatively high measurement noise is indeed the responsible of the loss of performance in the former case.

It worth noticing that the derivative in the equation 7.5 makes the KF to weight more the velocity data when the range between devices is short, which is the wanted behaviour since it is known that RSSI based estimations are more reliable when the communication distance is short [75].

The adopted KF relies on a second-order discrete-time, linear and time-invariant dynamic model excited by a random walk, as briefly explained below.

- In the **prediction stage** the predicted state results from

$$\hat{\mathbf{x}}_i(k+1)^- = A\hat{\mathbf{x}}_i(k) + B\varepsilon(k), \tag{7.6}$$

where $\hat{\mathbf{x}}_i(k) = [\hat{r}_i(k), \hat{\dot{r}}_i(k)]^T \in \mathbb{R}_{\geq 0} \times \mathbb{R}$ is the estimated state vector, the superscript $\cdot^-$ denotes the predicted quantities, $\varepsilon(k) \sim \mathcal{N}(0, \sigma_\varepsilon^2)$ is the unknown velocity increment, and

$$A = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix}, \text{ and } B = \begin{bmatrix} \Delta_t \\ 1 \end{bmatrix},$$

are the dynamic matrix and the velocity increment input matrix, respectively. It turns then out that the prediction covariance matrix at time $(k+1)\Delta_t$ results from

$$P_i(k+1)^- = AP_i(k)A^T + \sigma_\varepsilon^2 BB^T.$$

- In the **update stage** both the raging and velocity measures given by (7.3) and (7.4) are used. As a consequence, the Kalman gain is given by

$$K(k+1) = P_i(k+1)^- H^T S(k+1)^{-1},$$

where H is a bi-dimensional identity matrix, i.e. $H = I_2$,

$$S(k+1) = HP_i(k+1)^- H^T + R(k+1)$$

is the innovation covariance and $R(k+1) = \text{diag}(\sigma_r(k+1)^2, \sigma_v(k+1)^2)$. Therefore, the estimated state and its covariance matrix at time $(k+1)\Delta_t$ result respectively from

$$\begin{aligned}\hat{\mathbf{x}}_i(k+1) =& \hat{\mathbf{x}}_i(k+1)^- + \\ &+ K(k+1)\left(z(k+1) - \hat{\mathbf{x}}_i(k+1)^-\right), \\ P_i(k+1) =& \left(I_2 - K(k+1)\right) P_i(k+1)^-,\end{aligned}$$

where $z(k+1) = [r_{m,i}(k+1), v_{m,i}(k+1)]^T$.

From a computational view-point the proposed solution is very light. Indeed, the highest complexity comes from the *update stage*, where a matrix inversion is present. However, being the system bi-dimensional, a closed-form solution can be used, hence reducing to a trivial computation of some additions and multiplications. It is worth noticing that $\xi_v(k)$ in (7.4) is not white by definition. Therefore, formally the KF is not optimal. Nevertheless, the noise introduced by the derivative operator in (7.4) is so relevant in practice, that the correlation between noise samples has just a second-order effect. The dynamic model underlying the *prediction step* of the KF together with the model noise power σ_ε^2 is crucial for the filtering capabilities of the algorithm. By lowering σ_ε^2, a superior filtering effect is obtained, to the detriment of the actual ranging dynamic that is only approximately modeled by the matrix A. Indeed when the target is close to an anchor, its dynamic changes rapidly (i.e., the velocity changes sign, but not amplitude, within few samples). Hence, a higher value of σ_ε^2 is needed. However, in such a case the relevant noise variance of measurements comes into play and the estimation accuracy decreases. To account for this trade-off, an experiment-based tuning has been performed, as described in Section 7.3.

As a last remark, the constraint $\hat{r}_i(k) \geq 0$ should be always guaranteed. In fact, $\hat{r}_i(k)$ can occasionally become negative due to the measurement uncertainties. To enforce this condition, the estimate projection with inequality constraints presented in [104] can be applied. More precisely, after each prediction step, the condition

$$C\hat{\mathbf{x}}_i(k) \geq r_{\min} \geq 0,$$

with $C = [1, 0]$, is checked. If it is violated, then the estimated quantity is modified according to a Weighted Least Squares (WLS) approach as follows:

$$\hat{\mathbf{x}}_i^c(k) = \hat{\mathbf{x}}_i(k) + W_i(k)C^T\left(CW_i(k)C^T\right)^{-1}\left(r_{\min} - C\hat{\mathbf{x}}_i(k)\right), \tag{7.7}$$

where the superscript c in $\hat{\mathbf{x}}_i^c$ means *constrained* and the weighting matrix

$$W_i(k) = \begin{bmatrix} w_{i,1} & w_{i,2} \\ w_{i,2} & w_{i,3} \end{bmatrix} \tag{7.8}$$

is positive definite. As a result, (7.7) becomes simply

$$\hat{\mathbf{x}}_i^c(k) = \hat{\mathbf{x}}_i(k) + \begin{bmatrix} r_{\min} \\ \hat{r}_i(k) + \frac{w_{i,2}}{w_{i,1}}\left(r_{\min} - \hat{r}_i(k)\right) \end{bmatrix}. \tag{7.9}$$

It has to be noted that after the constrained solution is obtained, the **prediction stage** for the estimated states (7.6) modifies to

$$\hat{\mathbf{x}}_i(k+1)^- = A\hat{\mathbf{x}}_i^c(k) + B\varepsilon(k).$$

Furthermore, if $W_i(k) = P_i(k)$, the constrained solution has minimum variance, while if $W_i(k) = I_2$, the constrained solution is closer (in norm-2 sense) to the actual value [104]. Both solutions have been implemented and compared in Section 7.3.2.

7.2.2 Positioning Step

Once the distance values from different anchors are estimated by the KFs described in Section 7.2.1, in the *positioning step* the location $p(k) \in \mathbb{R}^2$ of the target on a plane at time $k\Delta_t$ (for the sake of brevity the time index k will be omitted in the rest of the description) results from the minimization of a cost function $f_{cost}(p, \hat{r}_i)\ \forall i \in \mathcal{A}$, where $\mathcal{A}$ is the set of anchor nodes from which the distance of the target is estimated. Even though such a cost function can be defined in a variety of ways, in the rest of this chapter a point-spring parallel model will be used to this purpose [92]. According to this model, all nodes are regarded as points and the links between the target node and each anchor $i \in \mathcal{A}$ are modeled as springs, whose resting length is a function of $\hat{r}_i$. Since the estimated target's position corresponds to the equilibrium point of the mass-point system [48], the cost function to minimize is the total elastic energy, i.e.

$$f_{cost}(p, \hat{r}_i) = \sum_{i \in \mathcal{A}} \frac{1}{2} q_i (\hat{r}_i - r_i(p))^2, \tag{7.10}$$

where $r_i(p) = \|p - a_i\|$ is the actual distance between the target and the $i-$th anchor located at a known position a_i in the reference frame considered and q_i is the elastic constant of the spring associated to the $i-$th link. In the case considered, two options are viable, i.e.

$$q_i = 1, \quad \forall i \in \mathcal{A}, \text{ or} \tag{7.11}$$

$$q_i = \frac{1}{\hat{r}_i^2}, \quad \forall i \in \mathcal{A}. \tag{7.12}$$

If (7.11) is used, the strength of all the springs is constant and this is a good choice when the noise level is constant over the measurement range. Conversely, (7.12) is more suitable when the noise level grows with the distance, e.g. when just RSSI data are used [75]. The solution to this problem is obtained using a standard solver for unconstrained minimization problems.

However, the problem can be regarded as a classic nonlinear regression problem (also known as nonlinear WLS). In fact, (7.10) can be equivalently rewritten as

$$f_{cost}(p, \hat{r}_i) = \frac{1}{2}\left(\hat{\mathbf{r}} - \mathbf{r}(p)\right)^T Q \left(\hat{\mathbf{r}} - \mathbf{r}(p)\right), \tag{7.13}$$

where $\hat{\mathbf{r}} = [\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_n]^T$, $\mathbf{r}(p) = [r_1(p), r_2(p), \ldots, r_n(p)]^T$ and $n = \#\mathcal{A}$. Therefore, an approximate solution to the problem can be computed through the following algorithm:

1. Initialize p_j to a dummy value, with $j = 0$;
2. Compute the Jacobian $H_j = \frac{d\mathbf{r}(p)}{dp}\Big|_{p=p_j}$ and $\mathbf{r}(p_j)$;
3. Compute the residuals $G_j = H_j^T Q^{-1} H_j$;
4. Update $p_{j+1} = p_j + G_j^{-1} H_j^T Q^{-1} \left(\hat{\mathbf{r}} - \mathbf{r}(p_j)\right)$ and increment j, $j = j + 1$;
5. If $j = j_{\max}$, p_j is the wanted solution, otherwise the algorithm should restart from step 2).

In the depicted problem, it is sufficient to have $j_{\max} = 3$ to reach a stable solution. The algorithm is extremely fast, not only because the number of iterations is very small, but also because G_j and G_j^{-1} can be computed in a closed form, since $p \in \mathbb{R}^2$. Therefore, no specific solvers are needed. Notice that if $Q = I_n$ (i.e. the weighting in (7.11) is adopted) a nonlinear Least Squares (LS) solution is obtained. However, such a solution tends to be approximated, as shown in Table 7.2 of Section 7.3.2.

7.3 Experimental Results

In this section the experimental setup and the results obtained for the *ranging step* and the *positioning step* are summarized and discussed.

7.3.1 Experimental Setup

Experiments have been conducted in the hall of the third FBK pavilion, Trento, Italy, over an area of about 15×50 m^2. The hall is higher than 10 m and the walls of the building are made of concrete, metal and glass. Therefore, the indoor environment considered is similar to a train station hall or a shopping mall. Six SensorTag devices based on the wireless module Texas Instruments CC2650 were used as anchor nodes at known positions. The anchors were placed onto some internal walls, along the perimeter of the building at about 1.2 m off the floor, while the target node collecting RSSI and ToF data was connected to a laptop via a USB link. During the experiments the laptop with the target was carried by a single user as shown in Fig. 7.1 and held approximately at the same height as the anchors. Even though the testing area was almost free of obstacles, no pure LOS conditions could be guaranteed, since some of the anchors were temporarily, but unavoidably concealed by the moving user's body.

I performed two types of repeated experiments. First, the accuracy of the *ranging step* of the algorithm was analyzed in the best possible (i.e. LOS) conditions. To this end, the value of the path loss coefficient α in (7.1) has been tuned measuring the RSSI between the target node and each one of the six anchors at different known reference distances d_0, i.e. 5 m, 10 m, 20 m, 30 m. Afterwards, the root mean square ranging errors (RMSE) were computed by changing the value of α in a range consistent with the characteristics of radio propagation in indoor environments and in LOS conditions [16]. Fig. 7.2 shows clearly that the minimum RMSE is achieved when $\alpha = 1.6$, which is in line with other results found in the literature [16]. The

FIGURE 7.1: Experimental setup: a CC2650 sensorTag (reference node) is installed on an internal wall. The PCA10056 development board equipped with the nRF52840 module (target node) is connected via USB to a laptop PC that is used to collect and log data.

RSSI-only ranging is then compared with those achieved with the KF described in Section 7.2.1.

In the second group of experiments, the positioning accuracy of the algorithm described in Section 7.2.2 was evaluated. To this purpose, the ground truth positions were determined by dividing the trajectories into segments over which the user walked approximately at a constant speed. Start and end of each segment were tagged by timestamps set by the user himself. Once the length of each segment is known, the average user's speed and his position can be reconstructed. I evaluated the accuracy of this method to be approximately 20 cm, that is one order of magnitude smaller than the expected localization error and, hence, negligible.

7.3.2 Description of results

In the following, the results of the experiments related to the *ranging step* and *positioning step* of the proposed algorithm are shortly described.

Ranging step

The outcomes of the first type of experiments confirm that the data fusion algorithm based on the KF described in Section 7.2.1 improves the ranging accuracy with respect to the case when just the RSSI distance values are considered. This is clearly visible in Fig. 7.3(a)-(b), where the distances between the user moving along straight paths and six different anchors are estimated by (7.1) (i.e. using RSSI data only) and the KF described in Section 7.2.1, respectively. In both cases, the mean distance values (blue lines) and the corresponding (gray) uncertainty band (centered in such mean values $\pm$ the sample standard deviations) are plotted as a function of the actual distance.

Observe that, over short distances (i.e. below 10 m), the ranging based only on RSSI data clearly returns overestimated results. This is probably due to the fact that

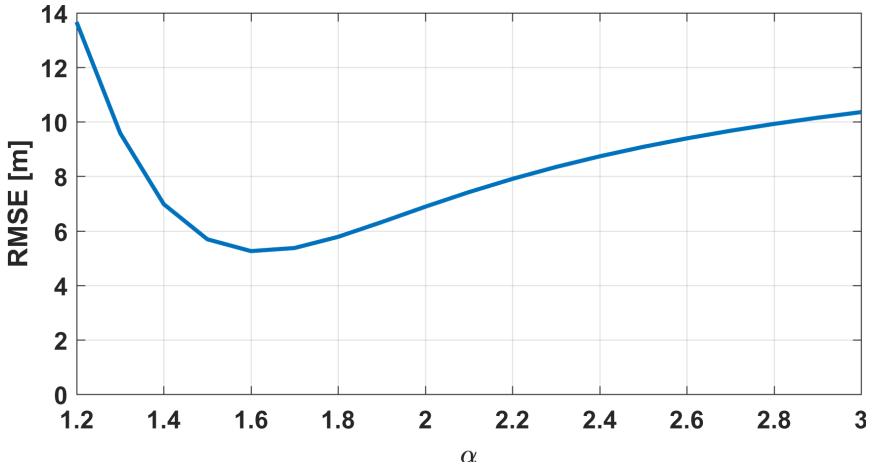
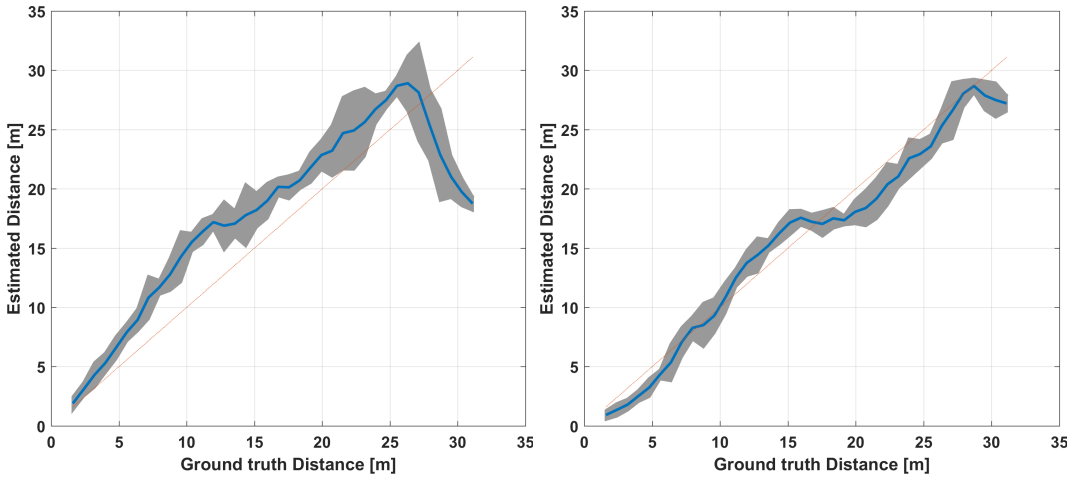


FIGURE 7.2: RSSI ranging RMSE as a function of the tuning parameter α.



(A) (a) (B) (b)

FIGURE 7.3: Ranging performance comparison between the RSSI-only approach (a) and the proposed KF (b). The blue lines refer to the estimated mean distance values, while the gray uncertainty band is given by the mean values $\pm$ the corresponding sample standard deviations.

the impact of multipath is less significant in the short-range. Therefore, even if the path loss coefficient $\alpha = 1.6$ minimizes the overall RMSE, it is too low in the short range. However, such overestimation almost disappears when the KF is used. In both cases, the range of variability due to random contributions is comparable. Notice that, over longer distances, not only the RSSI-based ranging uncertainty tends to grow, but larger deviations from the ideal behavior are also quite evident. Both phenomena are strongly mitigated by the KF.

Tab. 7.1 reports the mean and the RMSE of the distances of the experiments shown in Fig. 7.3. The corresponding grand mean values are also shown, for the sake of comparison. From Tab. 7.1, it results that the average mean ranging error associated with the KF is slightly larger than the value obtained using RSSI data only. However, the mean RMSE value (that is more important as it includes the effect of both systematic and random contributions) decreases by about 50%.

The good performance of the KF is confirmed also in the case of random (i.e.

TABLE 7.1: Mean and RMS ranging errors between the target and six anchors over straight-line paths, when the RSSI-only technique and the KF are used. Resolution is 5 cm.

	RSSI-only		KF	
	RMS [m]	Mean [m]	RMS [m]	Mean [m]
Anchor 1	5.75	-0.55	2.16	0.42
Anchor 2	8.62	-5.02	2.05	-0.36
Anchor 3	6.14	-2.21	3.20	-1.28
Anchor 4	4.17	-1.62	4.25	1.86
Anchor 5	5.32	-2.05	2.68	1.95
Anchor 6	4.13	2.55	2.34	1.29
Grand mean	5.69	-1.48	2.78	0.64

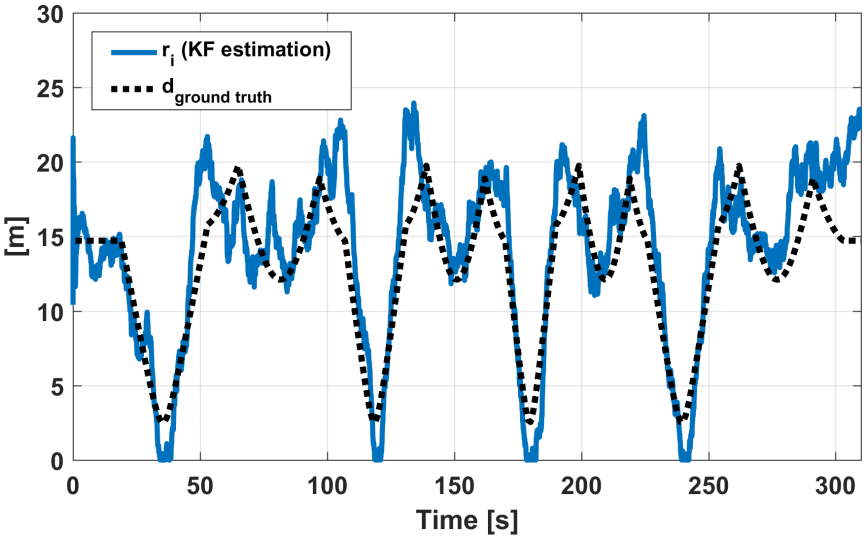


FIGURE 7.4: Distance estimated by the KF from anchor 6, when the user follows a random trajectory in the room.

non-straight) paths. This is visible in Fig. 7.4, which depicts the comparison, as a function of time, between the distance estimated by the KF from anchor no. 6 and the corresponding ground truth pattern. Again, the mean RMS ranging error is about 3.85 m, which is consistent with the results shown in Tab. 7.1.

Positioning step

As described in Section 7.2.2, the position estimation algorithm performs the fusion of the distance values returned by multiple KFs (one for each anchor). Fig. 7.5 shows the positioning results when the user moved over a rectangular path (black dotted lines) four times, starting from the position with coordinates (0,0). The black circle markers represent the position of the anchor nodes. Observe that Fig. 7.5(a) refers to the fusion algorithm that minimizes the cost function (7.10) by using the Quasi-Newton Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [101]. Fig. 7.5(b) depicts instead the path estimated by the approximated, low complexity WLS algorithm described at the end of Section 7.2.2. While the former solution is slightly more accurate, as confirmed by the empirical Cumulative Density Functions (CDFs) of positioning errors (namely the Euclidean distances between estimated and ground truth positions along the given path) shown in Fig. 7.6, the latter algorithm is computationally lighter and can be easily implemented in a low-cost, low-power embedded platform, such as the target node. It is worthwhile to note that both algorithms are executed with equal weights, i.e. (7.11) has been selected. This implies that Q in (7.13) is the identity matrix and, hence, the nonlinear WLS problem turns into a nonlinear LS one. Even though, from a theoretical viewpoint, the best weights should be given by the inverse of the ranging uncertainties estimated by the KFs in the update stage, this is not the case in practice since the KF implementation trades between filtering ability and model limitations, as already explained in Section 7.2.1. Since the estimated uncertainty turns to be slightly optimistic (i.e. better than it actually is), the choice of giving equal importance to different ranging measures is sensible and ensures better performance.

To provide a final quantitative comparison between different solutions, Tab. 7.2 shows the average mean and the average RMS values of the positioning errors when the user moves along the same trajectories shown in Fig. 7.5. Two different choices for the weighting matrix $W_i(k)$ in (7.7) are considered, i.e. $W_i(k) = P_i(k)$ and $W_i(k) = I_2$. It has to be noted that the results obtained with $W_i(k) = I_2$ are slightly more accurate than those based on $W_i(k) = P_i(k)$, $\forall i \in \mathcal{A}$. Again, this is due to the aforementioned effect on the estimated covariance matrices of the KFs. Furthermore, it is also evident that the nonlinear LS solution is slightly less accurate than using a numerical optimization of (7.10).

7.4 Conclusion

In this work I described a technique for indoor positioning based on Bluetooth Low Energy, therefore able to exploit the benefits, in terms of pervasiveness, of this energy-efficient communication standard. Obtained results (RMS value of positioning error in the order of 2.5 m, mean error is slightly higher than 2 m) are comparable with other reference works such as [40] where the average positioning error is 3.8 m. However, this reference result is based on a mix of real experiment and simulations, while my results are based only on real measurement campaign. Nevertheless, the authors addressed the scalability by considering the ToF ranging (that is more bandwidth demanding with respect to RSSI ranging) only to a subset of the available

TABLE 7.2: Average RMS and mean positioning errors when the target moves repeatedly over a rectangular path. The reported results refer to different solvers of the positioning optimization problem described in Section 7.2.2 and to different weighting matrices (7.8).

	$W(k) = P_i(k)$		$W(k) = I_2$	
	RMS [m]	Mean [m]	RMS [m]	Mean [m]
Numerical optimization of (7.10)	2.40	2.07	2.39	2.06
Nonlinear LS solution	2.70	2.34	2.54	2.19

nodes. They propose to use the geometric dilution of precision algorithm (GDOP) to select only the relevant links to perform the ToF measurements. This last consideration is definitely relevant also in a BLE based localization system that needs to guarantee the scalability. Results are also in line with another similar work [75], where RSSI and ToF where fused using two independent KF, in that case ranging uncertainty$< 2\ m$[1].

The proposed technique can be considered a proof of work for a fully mobile network where nodes are carried by group members, and localization is only relative to the other peers. Here I considered the case with some fixed node and one mobile target to be easier to implement and to validate rather than directly focusing on a completely dynamic network. Extending the solution proposed in this chapter to the group management problem is relatively straight forward since it just requires to change the set of links ($\mathcal{A}$) used to calculate the f_{cost} (Eq: 7.10), from:

- $\mathcal{A}$ = the links between the target node and each anchor

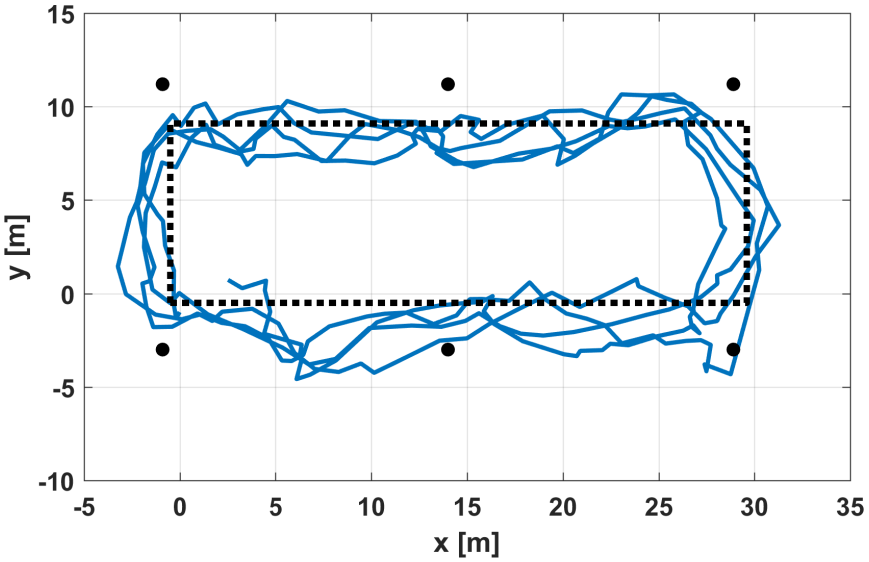
to:

- $\mathcal{A}$ = all the available links between peer nodes (i.e. group members)

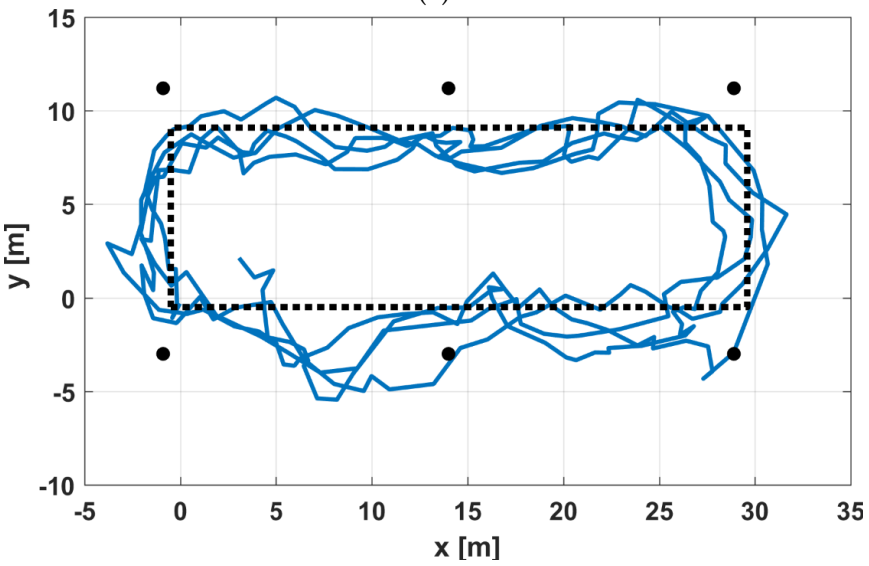
This will turn the algorithm to be *collaborative* (or *cooperative*), because location information is obtained using distance estimations from peers rather than using only reference nodes at known position. This positioning technique has already been exploited in Chapter 4 where it proved to work (even if the focus of that chapter was on RSSI based ranging).

I presented results from experiments conducted in the field using six anchors and a mobile target node. The performance comparison with or without data fusion clearly shows the benefits of the proposed approach. The mean RMS ranging error is indeed about 50% lower than in the case when just RSSI data are used (the same result is obtained in [40]). From the distance values estimated by the KF between the mobile node and different anchors the position of the target in a planar frame can be determined. Two approaches, both based on a mass-point model and on the solution of a nonlinear optimization problem, exhibit similar performance with a positioning uncertainty in the order of 2.50 m, which, as already stated, is in line with state of art. The two approaches mainly differ for the fact that one is conceived for resource-constrained devices and, therefore, it is computationally lightweight. I showed that, tailoring this algorithm for embedded platforms, positioning uncertainty grows by no more than 6%.

[1] RMSE value in dynamic conditions are not given in the paper, only uncertainty is reported.



(a)



(b)

FIGURE 7.5: Planar position of the target (blue lines) estimated using a numerical unconstrained optimization of (7.10) (a) and the algorithm described at the end of Section 7.2.2 (b), when the user moved over a rectangular path (black dotted lines) four times. The black circle markers represent the position of the anchor nodes.

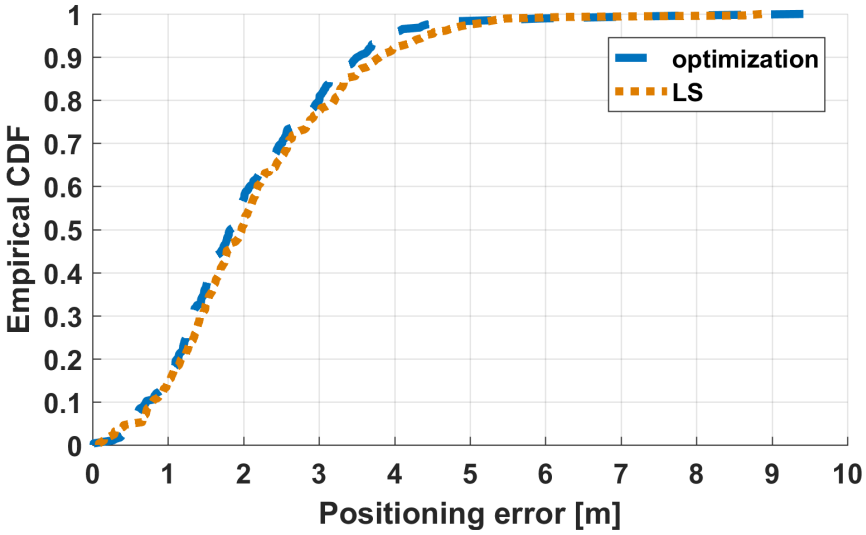


FIGURE 7.6: Empirical cumulative distribution function (CDF) of positioning errors.

Chapter 8

Conclusions

8.1 Managing groups of people in smart city

Managing groups of people moving together can be considered a trivial task since, in its basic form, it just requires to collect and maintain the list of neighbours. Moreover, standardization permits to developers to use many complex technologies without the need of understanding all the technical details of the lower layers, this substantially ease the development. However, by considering such system to be a tool of a smart city, through which services are provided to the citizens, it should be flexible enough to serve as many applications as possible. If on one side IoT experts forecast to have billions of devices connected to internet by the end of 2010's, on the other hand people won't like to carry many application specific devices with comparable characteristics. For this reason, a new technology, optimized for the purpose of identification and group management should be developed because those already available (such as Bluetooth, widely discussed in this thesis) can be tuned, optimized and employed for a specific application, but they fail in covering the needs of others. At E3DA group of FBK we focused on the walking bus scenario, however others interesting use cases can benefit from technologies here presented, therefore it is clear that the concept of radio identification and group management can be reused many fold in the context of smart city and communities in general. In some of the proposed use cases, it is challenging, if not unfeasible, to provide tags to all the group members. At least for those involving people, it might be more realistic to think about some hardware/software module (such as the Time-of-Flight library described in Chapter 6) in the own devices like smartphones or smartwatches that serves the purpose of radio identification. Anyway, this module should not be application specific otherwise it would require an explicit action from the user (like the installation of an app) and then it falls in the same problem of having to tag all the people. A standardization of radio identification with the purpose of group monitoring is therefore mandatory to make a single radio identification device usable by many applications; and if this single solution will be found, it will be easier to make people adopt it in place of application specific devices. The thesis mainly focuses on lower layers, however even higher ones require standardization: for instance, to resolve IDs a single way should be developed, otherwise it will be impossible to map unknown IDs to real objects/people.

8.2 Summary of contributions

The results and contribution included in this thesis can be grouped in two main topics: *Bluetooth (Low Energy)*, and *Radio based localization*. For the former, main contributions are related to the analysis of the communication and its optimization. Instead, for what regards the localization, novel contributions are mainly at the algorithm side. More specifically they are:

- Throughput and consumption analysis of Bluetooth 4.0 *Connection mode*. In Chapter 2 I have shown the promising performance of BLE in terms of energy efficiency, the analysis also demonstrates the mid-low throughput available over the BLE connection (compared to other standards such as wi-fi).
- The design of the first basic system based on BLE with two hops communication range is described and validated in Chapter 3.
- Analysis of the scalability constraints in *BLE Advertising mode*. Unexpected scalability performance of *BLE Advertising mode* have been demonstrated in Chapter 5.
- Wake-up radio (WuR) embedding into BLE standard have been described in Chapter 5 together with expected performance: this brings new opportunities for speeding up the maintenance of beacons and to reduce energy consumption when beacons are requested to scan for neighbours.
- The innovative technique for measuring the Time-of-Flight over a Bluetooth link is described for the first time in Chapter 6 and Chapter 7: ranging error have been reduced by more than 50% when Time-of-Flight is used to complement the more popular RSSI.
- In Chapter 4 I tested various approaches for ranging with RSSI by exploiting the redundancy of acquired data, also non-homogeneity of antenna radiation is considered in the algorithms.
- Time-of-Flight measurement on narrow band radio (such as those involved in BLE communication) has low resolution. However, since sample rate can be higher than the actual requirement, oversampling and filtering can be effectively applied to increase resolution and reduce the error. In Chapter 7 a filtering technique tailored to constrained platform is described and experimentally evaluated.
- Finally, in Chapter 7, the comparison between a high end optimizer and the lighter Least Square algorithm shown very minor performance degradation when the latter is preferred for the positioning step of the localization algorithm.

In summary, the scientific contribution of this thesis started from the validation of the Bluetooth Low Energy as communication standard for the IoT. The protocol has been characterized and optimized in terms of throughput, power and scalability, validating and stressing it even in complex networks such as mobile mesh. Studies and experiments have been carried on in the applicative scenario of the walking bus. Within this scenario, the major contribution is in the development of an innovative technique for measuring the distance between two BLE nodes, exploiting the combined effect of two known techniques (Time-of-Flight and RSSI) that are fused to obtain a single distance estimation that is used for locating the nodes.

8.3 Ongoing work

8.3.1 From anchor based indoor localization to anchor free outdoor localization

The description and discussions carried on in this thesis cannot be considered concluded here, in fact the discussion stops after showing a localization framework, which do not target directly the group management problem. Nonetheless a solution is not far, in fact, as briefly described in Section 7.4 expanding the positioning algorithm to work on anchor free dynamic network only requires some changes in the positioning step of the localization algorithm.

The results obtained with the localization solution presented in Chapter 7 will be tailored to run in a real setting where group management is required. This is ongoing work, as briefly described in 7.4. Expanding the positioning algorithm to work on anchor free dynamic network only requires some changes in the positioning step of the localization algorithm.

8.3.2 The network management

As already mentioned, the localization solution proposed in the last chapters exploits the Connection mode of the Bluetooth Low Energy standard, whose scalability performance is quite limited (every node can handle tens of concurrent connections[1]) while moving groups can reach hundreds of users in some situations.
I consider this problem similar to the network formation process in self organizing mesh networks. In general, once a mesh node boots up, it tries to join the mesh network, this means that it will try to create one or more links with other nodes (let's call them routers), which are already part of the network. When many routers are available, some policies will be applied to choose the ones to use. A policy may be to connect to the router with the highest link quality, in this case RSSI may be used to decide (the higher the RSSI the higher the link quality), but other policies for instance based on network status (i.e. on how many connections are handled by a single anchor) may be preferred. Alternatively, a state-of-the art algorithm such as *geometric dilution of precision* employed in [40] could be used.
The topic has not been studied yet, then speculations on this are not supported by experiments nor deep investigation, however the idea is to use the BLE scanning process to get a raw view of the network status. This could give a snapshot that can be used to tune the decision policy, for instance in the case where some nodes are moving away from the main group, they should be subjected to higher priority monitoring (i.e. they should maximize the number of links with the main group) so that there is more probability of keeping the link for longer time. This decision should be repeated periodically to account for scene change, how often and how long the scanning should be scheduled depends on requirements.

8.4 Final remark: what if every person have an RF beacon?

Airplanes have radio beacons that permits to anybody to detect the plane if you have a proper receiver. They transmit an ID and their GPS position, this is accepted by all of us because of safety reasons. Autonomous cars will likely have something

[1] The actual limit depends on connection setting and on hardware/software constraint; it is potentially different on every family of devices, and also given the same kind of BLE chip, the maximum number of concurrent connection may vary depending on the software version being used.

similar to be detectable by the other cars to avoid crashes: this is likely to be accepted for safety reasons and for services that will be associated. But will people accept to carry a personal beacon all the time? At present, possibly not, since many of us might be worried our privacy will be violated. However, we already carry our smartphone, which in many cases can provide our position to several services. In fact, we probably provided our consent when installing some applications trusting the provider and since in exchange we benefit of some valued service. Therefore, the answer to our initial question is that: yes, people will accept to wear a beacon if the benefit of carrying it and the level of trust is sufficient.

In conclusion, we are going towards a future where each of us will be identified by some kind of digital signature. Such technology will be the enabler for a number of applications, not only those discussed in this thesis: it will likely be used to check-in into protected building or in place of id card/passport, it may also be used to localize people in case of natural disaster such as it is already done with avalanche transceivers (also known as avalanche beacon). Localization aspects such a technology are discussed in this thesis, however, if the aforementioned personal beacon will take place in our daily life, security and privacy will be of primary importance at all the stack layers to make people accept the technology.

Bibliography

[1] ABI Research. https://www.abiresearch.com/. Accessed: 2017-09-22.

[2] Affoua Thérèse Aby, Alexandre Guitton, and Michel Misson. "Asynchronous blind MAC protocol for wireless sensor networks". In: *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*. IEEE. 2014, pp. 712–717.

[3] Stephan Adler et al. "Measuring the distance between wireless sensor nodes with standard hardware". In: *Positioning Navigation and Communication (WPNC), 2012 9th Workshop on*. IEEE. 2012, pp. 114–119.

[4] Abdulrahman Alarifi et al. "Ultra Wideband Indoor Positioning Technologies: Analysis and Recent Advances". In: *Sensors* 16.5 (2016).

[5] Marco Altini et al. "An ECG Patch Combining a Customized Ultra-low-power ECG SoC with Bluetooth Low Energy for Long Term Ambulatory Monitoring". In: *Conf. on Wireless Health*. 2011.

[6] Al Ameen et al. "A MAC protocol for body area networks using out-of-band radio". In: *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European*. VDE, pp. 1–6.

[7] Oliver Amft and Paul Lukowicz. "From Backpacks to Smartphones: Past, Present, and Future of Wearable Computers". In: *IEEE Pervasive Computing* 8.3 (2009), pp. 8–13. ISSN: 1536-1268.

[8] Giuseppe Anastasi et al. "Urban and social sensing for sustainable mobility in smart cities". In: *Sustainable Internet and ICT for Sustainability (SustainIT)*. IEEE. 2013, pp. 1–4.

[9] Ch Antonopoulos et al. "Experimental evaluation of a WSN platform power consumption". In: *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE. 2009, pp. 1–8.

[10] Abdalkarim Awad, Thorsten Frunzke, and Falko Dressler. "Adaptive distance estimation and localization in WSN using RSSI measures". In: *Digital System Design Architectures, Methods and Tools . DSD 2007. 10th Euromicro Conf. on*. IEEE. 2007, pp. 471–478.

[11] Mehedi Bakht, Matt Trower, and Robin Kravets. "Searchlight: helping mobile devices find their neighbors". In: *ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds*. 2011, pp. 1–6.

[12] Paolo Barsocchi et al. "A novel approach to indoor RSSI localization by automatic calibration of the wireless propagation model". In: *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*. IEEE. 2009, pp. 1–5.

[13] Anas Basalamah. "Sensing the crowds using bluetooth low energy tags". In: *IEEE Access* 4 (2016), pp. 4225–4233.

[14] Thomas Bell. "Automatic tractor guidance using carrier-phase differential GPS". In: *Computers and electronics in agriculture* 25.1-2 (2000), pp. 53–66.

[15] M Benocci et al. "Optimizing ZigBee for data streaming in body-area biofeedback applications". In: *Intl. Workshop on Advances in Sensors and Interfaces (IWASI)*. 2009, pp. 150–155.

[16] Alan Bensky. "Chapter 2 - Radio Propagation". In: *Short-range Wireless Communication (Second Edition)*. Ed. by Alan Bensky. Burlington: Newnes, 2004, pp. 11 –37.

[17] Pascal Bihler, Paul Imhoff, and Armin B Cremers. "SmartGuide–A smartphone museum guide with ultrasound control". In: *Procedia Computer Science* 5 (2011), pp. 586–592.

[18] Bluetooth SIG. "Specification of the Bluetooth system v4.0". In: (2010).

[19] Michael S Braasch and AJ Van Dierendonck. "GPS receiver architectures and measurements". In: *Proceedings of the IEEE* 87.1 (1999), pp. 48–64.

[20] D. Brunelli et al. "Bio-feedback system for rehabilitation based on a wireless body area network". In: *Fourth IEEE Int. Conf. on Pervasive Computing and Communications Workshops (PERCOMW'06)*. 2006. DOI: 10.1109/PERCOMW.2006.27.

[21] Davide Brunelli et al. "Design considerations for wireless acquisition of multichannel semg signals in prosthetic hand control". In: *IEEE Sensors Journal* 16.23 (2016), pp. 8338–8347.

[22] Chiara Buratti et al. "Design of a Body Area Network for Medical Applications: The WiserBAN Project". In: *Intl. Symp. on Applied Sciences in Biomedical and Comm. Technologies*. ISABEL '11. 2011, 164:1–164:5.

[23] Wolfram Burgard et al. "The interactive museum tour-guide robot". In: *Aaai/iaai*. 1998, pp. 11–18.

[24] B.H. Calhoun et al. "Body Sensor Networks: A Holistic Approach From Silicon to Users". In: *Proceedings of the IEEE* 100.1 (2012), pp. 91–106.

[25] M. Cattani, S. Guna, and G. P. Picco. "Group monitoring in mobile wireless sensor networks". In: *Intl. Conf. on Distributed Computing in Sensor Systems and Workshops (DCOSS)*. 2011, pp. 1–8.

[26] R. Cavallari et al. "A Survey on Wireless Body Area Networks: Technologies and Design Challenges". In: *IEEE Communications Surveys Tutorials* 16.3 (2014), pp. 1635–1657.

[27] *CC2650*. http://www.ti.com/product/CC2650. accessed 2016-4-20.

[28] Li Chen et al. "Range extension of passive wake-up radio systems through energy harvesting". In: *2013 IEEE Int. Conf. on Communications (ICC)*, pp. 1549–1554.

[29] Po Yu Chen et al. "A group tour guide system with RFIDs and wireless sensor networks". In: *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM. 2007, pp. 561–562.

[30] Junyoung Chung et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555* (2014).

[31] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. "Kernel-based object tracking". In: *IEEE Transactions on pattern analysis and machine intelligence* 25.5 (2003), pp. 564–577.

[32] D. Craven et al. "Potential for extended battery life in mobile healthcare with Bluetooth low energy and signal compression". In: *Signals and Systems Conference (ISSC 2012), IET Irish*. 2012, pp. 1–6. DOI: 10.1049/ic.2012.0209.

[33] Davide Giovanelli. https://bitbucket.org/dgiovane/time_of_flight_ble/. Accessed: 2018-07-06.

[34] A. Dementyev et al. "Power consumption analysis of Bluetooth Low Energy, ZigBee and ANT sensor nodes in a cyclic sleep scenario". In: *IEEE Int. Wireless Symposium (IWS)*. 2013, pp. 1–4. DOI: 10.1109/IEEE-IWS.2013.6616827.

[35] Ilker Demirkol, Cem Ersoy, and Ertan Onur. "Wake-up receivers for wireless sensor networks: benefits and challenges". In: *IEEE Wireless Communications* 16.4 (2009), pp. 88–96.

[36] *Eddystone*. https://developers.google.com/beacons/. accessed 2016-4-20.

[37] Estimote. https://estimote.com/. Accessed: 2017-08-03.

[38] Ramsey Faragher and Robert Harle. "Location fingerprinting with bluetooth low energy beacons". In: *IEEE journal on Selected Areas in Communications* 33.11 (2015), pp. 2418–2428.

[39] E. Farella et al. "Design and implementation of WiMoCA node for a body area wireless sensor network". In: *2005 Systems Communications (ICW'05, ICHSN'05, ICMCS'05, SENET'05)*. 2005, pp. 342–347. DOI: 10.1109/ICW.2005.39.

[40] T. Gädeke et al. "A bi-modal ad-hoc localization scheme for wireless networks based on RSS and ToF fusion". In: *Proc. 10th Workshop on Positioning, Navigation and Communication (WPNC)*. Dresden, Germany, 2013, pp. 1–6.

[41] A. Galov and A. Moschevikin. "Bayesian filters for ToF and RSS measurements for indoor positioning of a mobile object". In: *Proc. International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Montbéliard-Belfort, France, 2013, pp. 1–8.

[42] Arnab Ghosh, Prashant Kumar Gajar, and Shashikant Rai. "Bring your own device (BYOD): Security risks and mitigating strategies". In: *Journal of Global Research in Computer Science* 4.4 (2013), pp. 62–70.

[43] João C Giacomin et al. "Radio channel model of wireless sensor networks operating in 2.4 ghz ism band". In: *INFOCOMP Journal of Computer Science* 9.1 (2010), pp. 98–106.

[44] D Giovanelli, B Milosevic, and E Farella. "Bluetooth Low Energy for data streaming: Application-level analysis and recommendation". In: *Intl. Workshop on Advances in Sensors and Interfaces (IWASI)*. 2015, pp. 216–221.

[45] D. Giovanelli et al. "Dynamic group management with Bluetooth Low Energy". In: *2016 IEEE International Smart Cities Conference (ISC2)*, pp. 1–6. DOI: 10.1109/ISC2.2016.7580822.

[46] C. Gomez, I. Demirkol, and J. Paradells. "Modeling the Maximum Throughput of Bluetooth Low Energy in an Error-Prone Link". In: *IEEE Communications Letters* 15.11 (2011), pp. 1187–1189.

[47] Carles Gomez, Joaquim Oller, and Josep Paradells. "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology". In: *Sensors* 12.9 (2012), pp. 11734–11753. ISSN: 1424-8220. DOI: 10.3390/s120911734.

[48] Craig Gotsman and Yehuda Koren. "Distributed graph layout for sensor networks". In: *International Symposium on Graph Drawing*. Springer. 2004, pp. 273–284.

[49] Jayavardhana Gubbi et al. "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future Generation Computer Systems* 29.7 (2013), pp. 1645–1660.

[50] Chunlong Guo, Lizhi Charlie Zhong, and Jan M Rabaey. "Low power distributed MAC for ad hoc sensor radio networks". In: *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*. Vol. 5, pp. 2944–2948.

[51] Tero Hakkinen and Jukka Vanhala. "Ultra-low power wake-up circuit for short-range wireless communication". In: *Intelligent Environments, 2008 IET 4th International Conference on*, pp. 1–4.

[52] Mark Hedley and Jian Zhang. "Accurate wireless localization in sports". In: *Computer* 45.10 (2012), pp. 64–70.

[53] Mark Hedley et al. "Accurate tracking using TOA in sensor networks". In: *Intelligent Sensors, Sensor Networks and Information Processing, 2008. ISSNIP 2008. International Conference on*. IEEE. 2008, pp. 73–78.

[54] Karel Heurtefeux and Fabrice Valois. "Is RSSI a good choice for localization in wireless sensor network?" In: *IEEE 26th Int. Conf. on Advanced Information Networking and Applications*. IEEE. 2012, pp. 732–739.

[55] Héctor José Pérez Iglesias, Valentín Barral, and Carlos J Escudero. "Indoor person localization system through RSSI Bluetooth fingerprinting". In: *Systems, Signals and Image Processing (IWSSIP), 19th Int. Conf. on*. IEEE. 2012, pp. 40–43.

[56] IndoorAtlas. http://www.indooratlas.com/. Accessed: 2017-08-03.

[57] Hung-Chin Jang, Yao-Nan Lien, and Tzu-Chieh Tsai. "Rescue information system for earthquake disasters based on MANET emergency communication platform". In: *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*. ACM. 2009, pp. 623–627.

[58] A.J. Jara et al. "Evaluation of Bluetooth Low Energy Capabilities for Continuous Data Transmission from a Wearable Electrocardiogram". In: *Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. 2012, pp. 912–917. DOI: 10.1109/IMIS.2012.201.

[59] Vana Jelicic et al. "Benefits of wake-up radio in energy-efficient multimodal surveillance wireless sensor network". In: *IEEE Sensors J.* 14.9 (2014), pp. 3210–3220.

[60] Yunye Jin et al. "SparseTrack: Enhancing indoor pedestrian tracking with sparse infrastructure support". In: *INFOCOM, 2010 Proceedings IEEE*. IEEE. 2010, pp. 1–9.

[61] Shinsuke Kajioka et al. "Experiment of indoor position presumption based on RSSI of Bluetooth LE beacon". In: *Consumer Electronics (GCCE), IEEE 3rd Global Conf. on*. IEEE. 2014, pp. 337–339.

[62] S. Kamath. "Measuring Bluetooth Low Energy Power Consumption". In: *Application Note AN092 Texas Instruments*. 2010.

[63] Raman Kazhamiakin et al. "A gamification framework for the long-term engagement of smart citizens". In: *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE. 2016, pp. 1–7.

[64] Benjamin Kempke et al. "SurePoint: Exploiting Ultra Wideband Flooding and Diversity to Provide Robust, Scalable, High-Fidelity Indoor Localization". In: *14th ACM Conf. on Embedded Network Sensor Systems*. ACM. 2016, pp. 137–149.

[65] Philipp Kindt et al. "Precise Energy Modeling for the Bluetooth Low Energy Protocol". In: *CoRR*. 2014.

[66] Tong Kun Lai et al. "An 8x8 mm2 Bluetooth Low Energy wireless motion-sensing platform". In: *Int. Symp. on Information Processing in Sensor Networks*. 2014, pp. 341–342.

[67] Steven Lanzisera, David Zats, and Kristofer SJ Pister. "Radio frequency time-of-flight distance measurement for low-cost wireless sensor localization". In: *IEEE Sensors Journal* 11.3 (2011), pp. 837–845.

[68] Emanuele Lattanzi et al. "A sub-a ultrasonic wake-up trigger with addressing capability for wireless sensor nodes". In: *ISRN Sensor Networks* (2013).

[69] HyungJune Lee et al. "Localization of mobile users using trajectory matching". In: *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. ACM. 2008, pp. 123–128.

[70] Seung Joon Lee et al. "The group management system based on wireless sensor network and application on android platform". In: *Advanced Computer Science and Information Technology*. 2011, pp. 152–159.

[71] Yao-Nan Lien, Hung-Chin Jang, and Tzu-Chieh Tsai. "A MANET based emergency communication and information system for catastrophic natural disasters". In: *Distributed Computing Systems Workshops, 2009. ICDCS Workshops' 09. 29th IEEE International Conference on*. IEEE. 2009, pp. 412–417.

[72] Zhe-Min Lin et al. "Bluetooth Low Energy based blood pressure monitoring system". In: *Intelligent Green Building and Smart Grid (IGBSG)*. 2014, pp. 1–4.

[73] *Lineable*. http://lineable.net/. Accessed: 2016-04-29.

[74] Xiaowei Luo, William J O'Brien, and Christine L Julien. "Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites". In: *Advanced Engineering Informatics* 25.2 (2011), pp. 355–363.

[75] D. Macii et al. "A Data Fusion Technique for Wireless Ranging Performance Improvement". In: *IEEE Transactions on Instrumentation and Measurement* 62.1 (2013), pp. 27–37.

[76] E. Mackensen, M. Lai, and T.M. Wendt. "Bluetooth Low Energy (BLE) based wireless sensors". In: *IEEE Sensors*. 2012, pp. 1–4.

[77] M Magno et al. "A versatile biomedical wireless sensor node with novel dry-surface sensors and energy efficient power management". In: *IEEE Int. Work. on Advances in Sensors and Interfaces (IWASI)*. 2013, pp. 217–222.

[78] Michele Magno et al. "Combined methods to extend the lifetime of power hungry WSN with multimodal sensors and nanopower wakeups". In: *Wireless Communications and Mobile Computing Conference (IWCMC), 2012 8th International*. IEEE. 2012, pp. 112–117.

[79] Michele Magno et al. “Long-term monitoring of small-sized birds using a miniaturized bluetooth-low-energy sensor node”. In: *SENSORS, 2017 IEEE*. IEEE. 2017, pp. 1–3.

[80] Michele Magno et al. “Wake-up radio receiver based power minimization techniques for wireless sensor networks: A review”. In: *Microelectronics Journal* 45.12 (2014), pp. 1627–1633.

[81] Annapola Marconi et al. “Exploring the world through small green steps: improving sustainable school transportation with a game-based learning interface”. In: *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*. ACM. 2018, p. 24.

[82] Stevan Marinkovic and Emanuel Popovici. “Nano-power wake-up radio circuit for wireless body area networks”. In: *2011 IEEE Radio and Wireless Symposium*, pp. 398–401.

[83] Konstantin Mikhaylov, Nikolaos Plevritakis, and Jouni Tervonen. “Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI”. In: *Journal of Sensor and Actuator Networks* 2.3 (2013), pp. 589–613.

[84] Yuichiro Mori et al. “A self-configurable new generation children tracking system based on mobile ad hoc networks consisting of Android mobile terminals”. In: *Autonomous Decentralized Systems (ISADS), 2011 10th International Symposium on*. IEEE. 2011, pp. 339–342.

[85] Sarfraz Nawaz and Sanjay Jha. “A graph drawing approach to sensor network localization”. In: *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*. IEEE. 2007, pp. 1–12.

[86] Nordic Semiconductor. `http://blog.nordicsemi.com/getconnected/beacons-go-live-at-londons-gatwick-airport`. Accessed: 2017-09-01.

[87] Nordic Semiconductor. `https://devzone.nordicsemi.com/power/`. Accessed: 2018-02-20.

[88] Nordic Semiconductor. “S110 nRF51, SoftDevice Specification v2.0”. In: (2014).

[89] Federico Ossi et al. “Biologging in service of wildlife management: a system for bear dissuasion and reeducation”. In: *The 6th international bio-logging science symposium*. 2017, p. 217.

[90] Kaveh Pahlavan, Prashant Krishnamurthy, and Yishuang Geng. “Localization challenges for the emergence of the smart world”. In: *IEEE Access* 3 (2015), pp. 3058–3067.

[91] Ambili Thottam Parameswaran, Mohammad Iftekhar Husain, Shambhu Upadhyaya, et al. “Is RSSI a reliable parameter in sensor localization algorithms: An experimental study”. In: *Field failure data analysis workshop (F2DA09)*. Vol. 5. IEEE. 2009.

[92] Neal Patwari et al. “Locating the nodes: cooperative localization in wireless sensor networks”. In: *IEEE Signal processing magazine* 22.4 (2005), pp. 54–69.

[93] Gian Pietro Picco et al. “Geo-referenced proximity detection of wildlife with WildScope: design and characterization”. In: *Proceedings of the 14th international conference on information processing in sensor networks*. ACM. 2015, pp. 238–249.

[94] Rajeev Piyare et al. "Ultra low power wake-up radios: A hardware and networking survey". In: *IEEE Communications Surveys & Tutorials* 19.4 (2017), pp. 2117–2157.

[95] Nathan M Pletcher et al. "A 2GHz 52 μW wake-up receiver with-72dBm sensitivity using uncertain-IF architecture". In: *2008 IEEE Int. Solid-State Circuits Conference-Digest of Technical Papers*, pp. 524–633.

[96] Yihong Qi and Hisashi Kobayashi. "On relation among time delay and signal strength based geolocation methods". In: *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*. Vol. 7. IEEE. 2003, pp. 4079–4083.

[97] Cliff Randell, Ted Phelps, and Yvonne Rogers. "Ambient Wood: Demonstration of a digitally enhanced field trip for schoolchildren". In: *Adjunct Proc. IEEE UbiComp 2003* (2003), pp. 100–104.

[98] MA Razzaque and Siobhan Clarke. "Smart management of next generation bike sharing systems using Internet of Things". In: *IEEE Intl. Smart Cities Conference (ISC2)*. IEEE. 2015, pp. 1–8.

[99] N. E. Roberts et al. "A 236nW -56.5dBm-sensitivity bluetooth low-energy wakeup receiver with energy harvesting in 65nm CMOS". In: *2016 IEEE Int. Solid-State Circuits Conf. (ISSCC)*, pp. 450–451. DOI: 10.1109/ISSCC.2016.7418101.

[100] Nathan E Roberts and David D Wentzloff. "A 98nW wake-up radio for wireless body area networks". In: *2012 IEEE Radio Frequency Integrated Circuits Symposium*, pp. 373–376.

[101] David F Shanno. "Conditioning of quasi-Newton methods for function minimization". In: *Mathematics of computation* 24.111 (1970), pp. 647–656.

[102] Joseph A Shaw. "Radiometry and the Friis transmission equation". In: *American journal of physics* 81.1 (2013), pp. 33–37.

[103] M. Siekkinen et al. "How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4". In: *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. 2012, pp. 232–237.

[104] D. Simon. "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms". In: *IET Control Theory & Applications* 4 (8 2010), pp. 1303–1318. ISSN: 1751-8644.

[105] Christoph Sommer and Falko Dressler. "Using the right two-ray model? A measurement based evaluation of PHY models in VANETs". In: *Proc. ACM MobiCom*. 2011, pp. 1–3.

[106] Dora Spenza et al. "Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems". In: *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE. 2015, pp. 522–530.

[107] H. Strey et al. "Bluetooth low energy technologies for applications in health care: proximity and physiological signals monitors". In: *Int. Conf. on Emerging Technologies for a Smarter World (CEWIT)*. 2013, pp. 1–4. DOI: 10.1109/CEWIT.2013.6851347.

[108] Bjorn Thorbjornsen et al. "Radio frequency (RF) time-of-flight ranging for wireless sensor networks". In: *Measurement Science and Technology* 21.3 (2010), p. 035202.

[109] Joaquín Torres-Sospedra and Adriano Moreira. "Analysis of Sources of Large Positioning Errors in Deterministic Fingerprinting". In: *Sensors* 17.12 (2017), p. 2736.

[110] Sana Ullah and Kyung Sup Kwak. "An ultra low-power and traffic-adaptive medium access control protocol for wireless body area network". In: *Journal of medical systems* 36.3 (2012), pp. 1021–1030.

[111] Asimina Vasalou, Anne-Marie Oostveen, and Adam N Joinson. "A case study of non-adoption: the values of location tracking in the family". In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM. 2012, pp. 779–788.

[112] Morris Williams et al. "Children and emerging wireless technologies: Investigating the potential for spatial practice". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2005, pp. 819–828.

[113] Henk Wymeersch, Jaime Lien, and Moe Z Win. "Cooperative localization in wireless networks". In: *Proceedings of the IEEE* 97.2 (2009), pp. 427–450.

[114] Jiuqiang Xu et al. "Distance measurement model based on RSSI in WSN". In: *Wireless Sensor Network* 2.08 (2010), p. 606.

[115] Zheng Yang, Yunhao Liu, and X-Y Li. "Beyond trilateration: On the localizability of wireless ad-hoc networks". In: *INFOCOM 2009, IEEE*. IEEE. 2009, pp. 2392–2400.

[116] Hiroyuki Yomo, Yoshihisa Kondo, Noboru Miyamoto, et al. "Receiver design for realizing on-demand WiFi wake-up using WLAN signals". In: *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 5206–5211.

[117] Menghan Zhang, Weiwei Xia, and Lianfeng Shen. "Bluetooth Low Energy based motion sensing system". In: *Int. Conf. on Wireless Communications and Signal Processing (WCSP)*. 2014, pp. 1–5. DOI: 10.1109/WCSP.2014.6992068.

[118] Bing Zhou et al. "A Bluetooth low energy approach for monitoring electrocardiography and respiration". In: *Intl. Conf. on e-Health Networking, Applications Services (Healthcom)*. 2013, pp. 130–134.