Ad Hoc Communication in Teams of Mobile Robots Using ZigBee Technology

AMADEU FERNANDES, MICAEL S. COUCEIRO, DAVID PORTUGAL, JOÃO MACHADO SANTOS, RUI P. ROCHA

Institute of Systems and Robotics, University of Coimbra, 3030-290, Coimbra, Portugal

Received 16 January 2015; accepted 22 February 2015

ABSTRACT: Mobile robots developed at the Institute of Systems and Robotics (ISR) Coimbra feature ZigBee technology for inter-robot communication. The main aim of this paper is to implement and validate ad hoc wireless communication functions between robotic teammates using the ZigBee technology, thus integrating and developing the features of the XBee Original Equipment Manufacturer Radio Frequency module via standard Arduino Serial Commands. This work provides a useful instructive tool for research, enabling the interaction and cooperation of a team of mobile robots in areas such as swarm robotics, multi-robot patrolling, search and rescue, among others. In order to validate the functional requirements, several experiments were performed at the level of peer communication, namely: i) analysis of the Received Signal Strength Indication (RSSI) of messages by peers to estimate inter-robot distances for localization purposes; ii) analysis of the communication complexity in experiments with large groups of mobile robots using Zigbee communication modules and its application in a real world exploration scenario. © 2015 Wiley Periodicals, Inc. Comput Appl Eng Educ 23:733–745, 2015; View this article online at wileyonlinelibrary.com/journal/cae; DOI 10.1002/cae.21646

Keywords: ad hoc communication; multi-robot systems; ZigBee; Arduino

INTRODUCTION

Wireless communication technologies have evolved rapidly in recent years, being widely disseminated in our society [1]. Advances in micro-electromechanical systems enable integration of sensors, signal processing, and radio frequency capacity in very small devices. All types of portable applications tend to be able to communicate without using a wired connection [2]. The goal of wireless communication, in the context of multiple devices, is to exchange and gather information so as to perform a task in a given environment. This way, a typical intelligent node (e.g., sensor, robot, etc.) comprises of a unit for acquisition, processing and transmission of data. Due to the aforementioned technological advances, namely low cost, low power consumption, and low data transmission rates; wireless technologies have emerged. This has motivated research in the field of mobile robotics, especially in areas that require interaction and cooperation between mobile robots. Wireless technologies, such as Wi-Fi, Bluetooth, and ZigBee, enable mobile robots to communicate on an ad hoc basis, commonly known as mobile ad hoc networks (MANETs), allowing to cover a vast geographical area, without the need of using centralized communication in difficult environments or infrastructure deployment in emergency contexts (e.g., search and rescue scenarios). Research in the areas of multi-robot patrolling, swarm robotics and search and rescue has been carried out at the Institute of Systems and Robotics (ISR) in Coimbra [3–5], where low cost Arduino-based educational mobile platforms were built [6]. This work focuses on the implementation of MANETs in these mobile platforms, using the ZigBee technology, by integrating the OEM RF XBee Series 2 module coupled with the microcontroller in the Arduino board of the platforms. We analyze in detail the module's performance in terms of the received signal strength information (RSSI) so as to conduct localization experiments. In addition, the capabilities of the robots are extended by integrating ZigBee wireless communication in the Robot Operating System (ROS) framework, through the robots' ROS driver. By presenting the development and integration of these capabilities in our custom-assembled robots, we believe that this paper may serve as an instructive guide for researchers interested in the integration of Zigbee technology in static or mobile wireless sensor networks.

The rest of this article is organized as follows. An introduction to Zigbee technology is provided in ZigBee MANETs section. Afterwards, we focus on the implementation of wireless communication using the OEM RtF XBee Series 2 module and the

Correspondence to D. Portugal (Davidbsp@irs.uc.pt)

Arduino Uno board, inside TraxBot and StingBot mobile robots. In Experimental Study section, experimental results are presented and discussed. Finally, the article ends with conclusions and future work directions.

ZIGBEE MANETS

Mobile Ad Hoc Networks (MANETs) are a paradigm for mobile communication in which the nodes are mobile and are thus dynamically and arbitrarily located in such a way that communication between nodes is not dependent on an underlying static infrastructure [7]. Nodes in a MANET are usually mobile devices with limited resources: energy processing capability and storage capacity. Since no fixed infrastructure or centralized administration is available, these networks are self- organized [8]. In addition, the nodes in a MANET usually depend on their neighbors to deliver information to the destination nodes that they wish to communicate with (i.e., multi-hop communication). These networks are formed by an arbitrary number of devices with wireless communication capabilities, possibly heterogeneous, which associate and dissociate freely. MANETs can be applied to wireless mesh networks, military applications, mobile internet access, and emergency response networks, among others. These networks are confronted with the traditional problems inherent to wireless communication, namely security, interference, low bandwidth, etc. In spite of the design constraints associated to wireless networks [9], MANETs are highly suitable for use in situations where there is a need to quickly deploy a communicating system (e.g., in emergency situations). Due to the capacities of self-creation, self-organization and self-administration, MANETs can be implemented with minimal user intervention, and there is no needed for detailed planning and installation of connections to the base station. Mobile nodes communicate with other nodes outside their immediate range using wireless communication multi-hop [10]. In addition, the topology of ad hoc networks is highly dynamic; therefore traditional routing protocols are not viable. The routing protocol should be able to keep up with node mobility, which often leads to drastic changes of the topology of the network over time in an unpredictable way [11].

ZigBee is a global standard for wireless communication, with a focus on regulating and enabling product interoperability. It was established by IEEE [12] and the ZigBee Alliance to provide the first general standard for network applications that uses IEEE 802.15.4. The ZigBee Alliance is an industrial consortium which aims to promote and develop wireless networks for industrial monitoring and control, but also for home networking, medical sensor applications, games, and other application areas where low cost networks are needed, as well as low power and interoperability. ZigBee networks allow robust communications and offer excellent immunity against interference, and the ability to host thousands of devices in a network (theoretically 65536), with data transfer rates of up to 250 Kbps.

ZigBee is available in two feature sets [13]: ZigBee PRO and ZigBee. Both sets define how mesh networks operate. The ZigBee PRO, the specification most widely used, is optimized for low power consumption and supports large networks with thousands of devices, while ZigBee is intended for networks with lower number of devices. ZigBee networks involve two types of physical devices: the Full Function Devices (FFD) and the Reduced Function Devices (RFD). The FFDs operate in any topology [14], capable of performing the functions of Network Coordinators, Routers or End devices; they communicate with any other device and are complex to implement. The RFDs, on the other hand, are limited to the star topology. They cannot be network Coordinators, can only communicate with the Network Coordinator and have a very simple implementation.

The Coordinators form the root of the network tree and can be a bridge to other networks in order to expand it. There is exactly one Coordinator in each network. Router devices function as intermediates, retransmitting data to other nodes. Finally, the end devices are typically used to get information from multiple sensors by communicating with their parents (the Coordinator or the router). When compared to Wi-Fi, Bluetooth, and ZigBee are specially designed for ad hoc sensors networks, whose coverage can be extended by selecting an adequate topology. The ability of ZigBee to form a tree topology increases the ad hoc network coverage significantly, making it a better choice for decentralized applications that require a great number of sensors spread over a large geographic area. One can verify that Wi-Fi leads the WLAN market for home networking, small offices and access in public buildings. Bluetooth has replaced wires in telephone ad hoc networks, connections within the car and audio devices. It is expected that ZigBee will lead to a similar market for mobile robotics, sensor networks, medical data applications and others [15].

IMPLEMENTATION OF ZIGBEE COMMUNICATION IN MOBILE ROBOTS

The ZigBee Technology was conceived with the intent of creating a single standard for wireless communication, with low cost and low power of operation, being ideally suited for applications with low data rate transmission, security needs and long duration of execution (e.g., environmental sensing [16]). Hence, this technology is very attractive from a robotics perspective since it may enable long periods of interaction and coordination of teams of robots within large environments, such as in swarm applications operating in infrastructure-less scenarios, wherein each robot represents a node of the MANET [4]. Below, we present the ZigBee modules used in this work, which were embedded in the Arduino-based mobile robots developed.

XBee OEM RF

The XBee OEM RF modules maintain the original Zigbee requirements, which are fundamental in communications within mobile robot teams. These modules are manufactured by Digi International [17] providing wireless connection by means of Radio Frequency (RF), using the IEEE 802.15.4/ZigBee network protocol.

There are two types of operation modes: the AT mode (a.k.a. "transparent" mode) and the Application Programming Interface (API) mode. In AT mode, all data sent to the XBee module is immediately forwarded to the remote module identified by the Destination Address in memory. This is commonly used in very simple networks, or simple point-to-point communication. In API mode, all data entering and leaving the XBee module is contained in a frame that defines the operations or events. API command mode enables the configuration of the module at the application layer, which creates the respective packet with the data, the address and the identifiers necessary to establish communication with other devices.

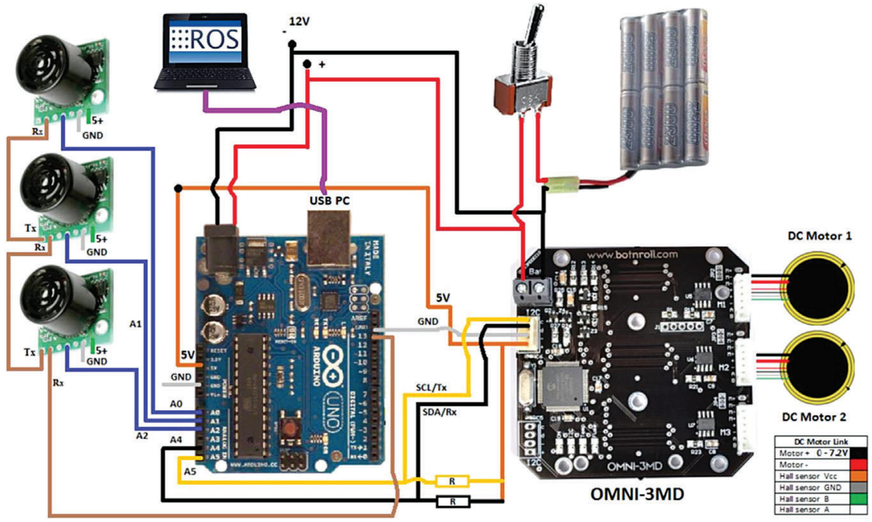


Figure 1 The main circuit of the Arduino-based educational robots developed by the authors. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Building a ZigBee network is done automatically by the XBee devices during startup. The ZigBee Coordinator starts the network, making a sweep around that particular area to discover available Personal Area Network (PAN) addresses and channels. The information in the *Scan Channels (SC)* and *Beacon Scan* parameters is used to select the channel and PAN ID. The Coordinator then sends a beacon request and waits for a response. Afterwards, routers and end devices can join the network and are assigned to a 16-bits address. The end device does not allow other devices to associate with them, and the coordinators and routers may, or may not, accept association of other devices, allowing up to a maximum of eight "children". To transmit a message to another device, it is only necessary to configure the destination address. This may be the serial number (64-bits), the 16-bits address, or the Node Identifier (NI) ASCII string [18].

Arduino-based TraxBot and StingBot robots

Both TraxBot and StingBot mobile robots [6], developed at ISR Coimbra are equipped with XBee Series 2 modules coupled to the main control board Arduino Uno. These robots were generically designed to meet the needs of swarm robotics and multi-robot tasks. A schematic of the major components of these robots, with the Arduino Uno processor board in the middle, is shown in Figure 1. The information of the remaining components of the main circuit and robots assembly can be found in [6].

The Arduino Uno is an open source development board that contains an ATmega328p 8-bit embedded microcontroller, which provides serial communication. Serving the needs of the described work, this board is ideal for compact robots, allowing optimization of space and power consumption. The Arduino enables extensible features in terms of software and hardware: it can be extended through the use of C/C++ libraries, and diverse shields may be coupled on top of the board, thus extending its capabilities. With this type of technology, one can benefit from several communication strategies for inter-robot or robot-computer interface, namely I2C, SPI, and more relevant in the context of this work, Serial. Serial communication is provided over USB, being referred as a COM port on the computer. Taking advantage of the extensible hardware characteristics of the Arduino Uno, the XBee series 2 module can be integrated into the Arduino-based robots using an interfacing XBee Shield that can be attached to the board using connections suitable for that specific purpose.

Besides the hardware integration, it was still necessary to develop the software to benefit from the XBee series 2 full functionalities so as to establish inter-robot communication within dynamic networks. The use case diagram illustrated in Figure 2 describes the functional requirements for the multi-robot communication system.

The robots, actors in this system, should be able to identify new nodes in the network (whether robots or other agents), as well as those leaving the network. In addition, they should send and receive simple messages to/from other nodes; forward data that is destined to other nodes, broadcast messages and read the Received Signal Strength Indication (RSSI) of messages sent by neighboring nodes. The software that provides these functionalities resides in the Arduino board, enabling the communication between the microcontroller and the XBee series 2 module across the XBee Shield. We freely provide the full source code of the developed software[1] in order to facilitate the integration of ZigBee communication in teams of mobile robots to the research community.

The authors have developed a library that implements some of the necessary requirements in the use case diagram of Figure 2, which are not covered in the official XBee API. Having that in mind, for the remaining ones we have used XBee API functions, namely for sending, receiving, forwarding and broadcasting messages. The identification of new nodes using discovery functions, the departure of network nodes by updating the routing tables, and the reading of the RSSI of the incoming message using an API command are provided in the library developed by the authors: the *XbeeNode* API.

[1]https://db.tt/2JjPAtzC

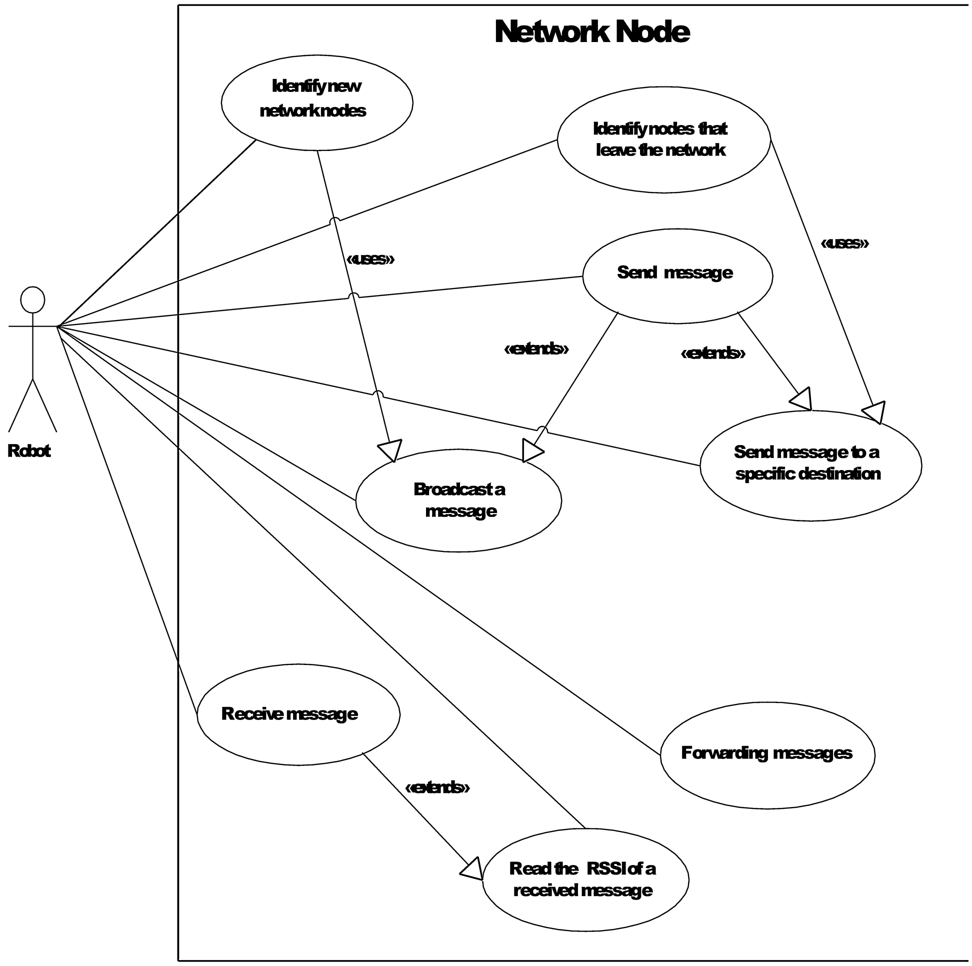


Figure 2 Use case diagram of the multi-robot communication system developed.

ZigBee Communication in the Robot's ROS driver

With the continuous growth of robotics in recent decades [19], especially with the integration of several sensors in robotic platforms, software development for robots has become arduous. Different robots can have widely different hardware, making code reuse non-trivial. Furthermore, it requires the programmer to often master the hardware used, and requires advanced knowledge on software development. To circumvent these challenges, several robotic frameworks and *middlewares* have been proposed to manage the complexity and facilitate rapid prototyping for real-world experiments (e.g., [20]).

Nowadays, ROS (Robot Operating System[2]) is the closest framework to the de facto standard that the Robotics community needed, being used worldwide. Among the many advantages of ROS [21], it promotes hardware abstraction due to its modular nature, which means that code and algorithms written in ROS, can be used in several different robots (e.g., as seen in [22]). In addition, it does not require hardware expertize, as several drivers for commonly used sensors are readily available in ROS. That being said, the authors have developed a ROS driver for the Arduino-based mobile robots developed [6]. This enabled robots that are running ROS to receive and send ZigBee messages, which was one of the main requirements, and an important contribution to the ROS community.

[2]http://www.ros.org/

The communication between the computer that executes ROS and the Arduino board is done through a USB cable. However, it uses the same port that is responsible for the communication with the Xbee Shield. Due to the limitation of the Arduino Uno, which only has one serial port for communication, it was necessary to find a solution to enable the communication to the XBee module and the computer running ROS at the same time. The adopted solution consisted on the virtualization of a serial port using the digital pins of the Arduino board. With this solution, the *SoftwareSerial* library was used to communicate with the Xbee Serial 2 module via a virtual port, and the serial port was used for communication with the computer.

Beyond the need to adapt the hardware (as shown in Fig. 3), it was necessary to modify the Xbee API library in order to operate with the *SoftwareSerial* library, thus emulating the functionalities of the *HardwareSerial* library.

With this crucial modification, it was relatively straightforward to include the Zigbee communication in the ROS driver of the robots. For this purpose, an open source *firmware* code

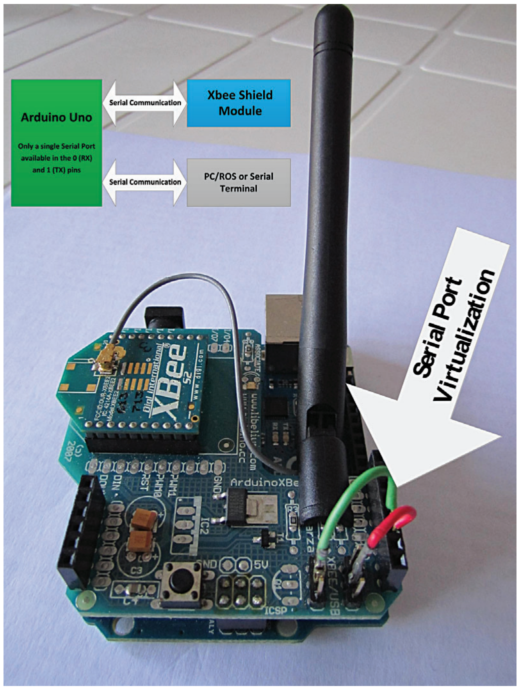


Figure 3 Communication between the Arduino Uno, xBee Shield Module, and the PC/ROS or Serial terminal, and Serial Port Virtualization. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

was developed, becoming resident in the Arduino board, and exposing the Zigbee communication as well as other robot's functionalities (e.g., odometry estimation, range sensing, move motors, etc.) to ROS. Figure 4 shows how the robot driver and standard navigation modules available in ROS exchange information.

In the PC/ROS side, communication with the Arduino board is enabled using the *serial_communication*[3] stack available for ROS and the custom protocol adopted in [6]. This way, any application developed for the robot in ROS is now capable of sending and receiving Zigbee messages, discover new robots in the network and evaluate if other robots have left the network.

In order to test the integration of the ZigBee Ad Hoc communication in ROS, a simple experiment was recorded[4]. In this video, it is possible to verify that the robot is able to discover peers in the network and communicate to them by unicast and broadcast. LEDs were used to signalize communications. All the code is available online (see footnote 1). In the next section, we conduct several experiments to evaluate our multi-robot communication system using ZigBee technology.

EXPERIMENTAL STUDY

In our experimental study, we evaluate ZigBee networks wherein each node consists of an Arduino-based mobile robot. For a successful deployment, we need to assess some basic performance parameters, such as radio signal quality, measured by the RSSI. Moreover, we use the RSSI to estimate the distance between robots and, as a consequence, we estimate the position of one of the robots using a trilateration method [23]. Additionally, we present a real-world experiment with mobile robots that make use of a MANET for coordination of their actions in a collective swarm task.

RSSI Measurement

The RSSI of received packets is used for various purposes, especially for the localization of nodes within the network and to estimate the quality of the corresponding links between them. However, the uncertainties involved in measuring the received signal strength leads to inaccuracies in the results obtained [24]. The XBee modules provide two ways of reading the signal level (in -dBm) of the last packet successfully received: (1) encoded in the Pulse Width Modulated (PWM) signal and available in the pin 6 of the XBee module. When the module receives a message, the PWM is set based on the RSSI value received and this information is related only to the quality of the last hop; and (2) using API commands (command "DB"), which also indicates the RSSI value of the last hop.

Therefore, if the transmission is multi-hop, both strategies are unable to provide any signal quality measure of all hops. The main advantage of the second method over the first one is the complete lack of synchronization required between both Arduino and XBee module. Although this could be achieved using external interruptions, those also increase the computational effort of the microcontroller. Hence, in these experiments, we considered the second method for reading the RSSI value. The RSSI values reported by the XBee series 2 modules fall within the range of -26 to -98 dBm, using the integrated Whip antennas.

In the RSSI reading experiments, two TraxBot robots containing the XBee Series 2 modules, one as Coordinator and another as Router/End device, were placed in line of sight in an indoor environment, and the distance between them was continually increased with increments of 10 cm up to a distance of 20 m. This test was conducted in order to verify the relationship between the RSSI values and the distance. All data was acquired using a notebook incorporated in one of the robots for subsequent pre-processing. For each increment of 10 cm, 6 measurements were acquired.

The relationship between the median at each distance and the RSSI is illustrated in Figure 5a. One may observe that, as the distance increases, the RSSI values tend to decrease. However, there is a high variability on the RSSI value received, especially for larger distances due to multiple reflections of electromagnetic waves on the walls.

In a subsequent experiment, a similar study was conducted in a completely open environment, as shown in Figure 8. The relationship between the median at each distance and the RSSI for this case is illustrated in Figure 5b. Notice that the RSSI values present less variability when compared to the ones retrieved in the indoor experiment. Yet, it still shows significant variations.

Estimation of the Distance Between Nodes Using RSSI

There are several methods for estimating the distance between two robots using communication modules. For this purpose, it is

[3]http://wi ki.ros.org/serial_communication
[4]http://www.youtube.com/watch?v=hy2p6AytK3o&hd=1

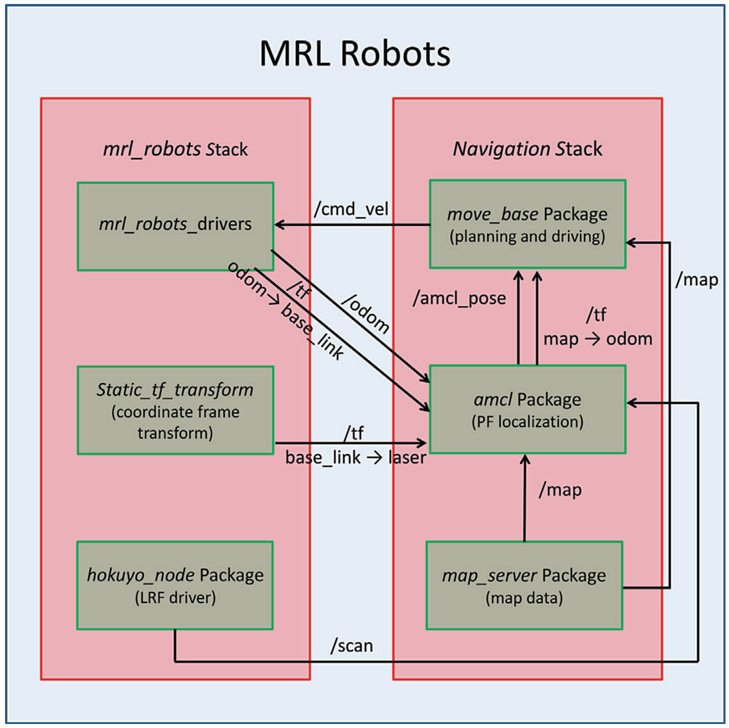


Figure 4 Flow of information between the robot driver and navigation modules available in ROS. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

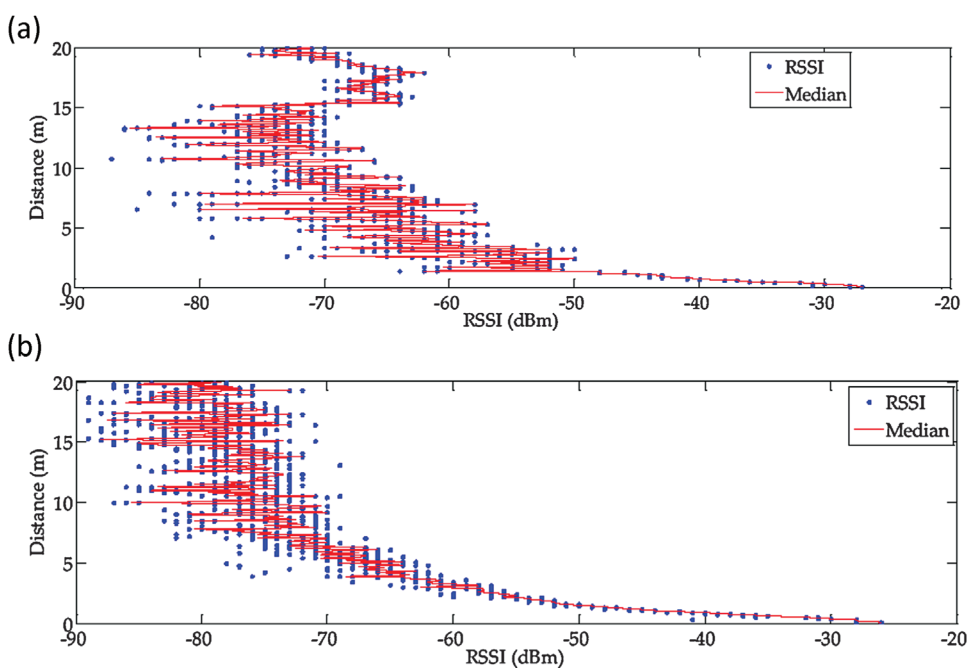


Figure 5 (a) Relationship between RSSI and distance in the RSSI indoor experiment. (b) Relationship between RSSI and distance in the RSSI outdoor experiment. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

necessary to evaluate physical magnitudes that establish a relationship with certain distances. These magnitudes usually represent the time that a signal takes to travel the distance from the receiver to the transmitter (e.g., the time of arrival (TOA) [25], the power of an acoustic signal [26], the RSSI, among others). It is important to consider that all these methods are subject to errors, and the use of each of them is related to the project requirements.

In this work, we have used the RSSI to estimate the distance between two robots, since it can be directly acquired with the XBee Series 2 modules, as described earlier. Using the data previously obtained in the outdoor experiments, an equation that relates the distance as a function of RSSI was formulated:

$$d(\mathrm{rssi}) = \mathrm{a}_1 \mathrm{e}^{-\left(\left(\frac{\mathrm{rssi}-\mathrm{b}_1}{\mathrm{c}_1}\right)^2\right)} + \mathrm{a}_2 \mathrm{e}^{-\left(\left(\frac{\mathrm{rssi}-\mathrm{b}_2}{\mathrm{c}_2}\right)^2\right)} \tag{1}$$

In order to obtain such relationship, the MATLAB Curve Fitting Tool [27] was utilized to compute the curve that approximates the distance (*d*) according to the RSSI values by a Gaussian function with $a_1 = 4.283$, $b_1 = -90.84$, $c_1 = 6.973$, $a_2 = 31.69$, $b_2 = -128.6$ and $c_2 = 41.24$.

The quality of the curve obtained, displayed in Figure 6, is evaluated by the *coefficient of determination* $R_2 = 0.9668$, wherein R_2 (R square), varies between 0.0 and 1.0. The closer R_2 is to 1.0, the better the approximation is deemed to be.

After obtaining the relationship between the distance and the RSSI values, given by Equation (1), another outdoor experiment was conducted, in which 30 RSSI samples were obtained at different distances, with the distance from the robots increasing up to a maximum of 20 m. In Figure 7 the estimated distance (dashed black line) is shown. In addition, the evolution of the absolute error (calculated using the median RSSI) versus distance is also illustrated, in red. Results clearly show that beyond distances of 6 m, the estimation becomes much less accurate due to fluctuations in the RSSI values over larger distances. The rssi variable of Equation (1) was obtained by calculating the median of the 30 samples of RSSI for each distance.

Estimating the location of a robot by trilateration

Usually in mobile robotics, it is an important requirement that the robot is capable of estimating its position in the scenario. However, this problem usually consists in equipping mobile robots with powerful sensing and processing abilities, and implementing localization filters, such as Extended Kalman Filters or Particle Filters. This often involves unbearable hardware

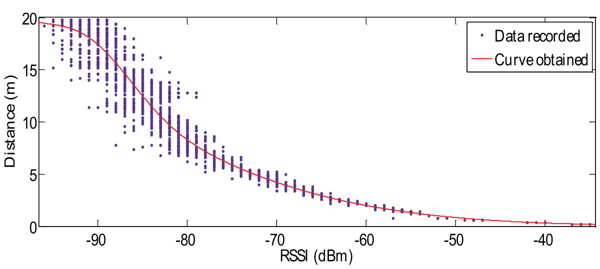


Figure 6 The resulting curve determined by Equation 1, which estimates the distance of the robots according to the RSSI. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

costs, especially when it comes to research in multi-robot systems. On the other hand, in order to compensate the lack of precision and the noisy data provided by low-cost hardware, it is necessary to develop efficient methods in terms of software, taking into account the available sensors in the robot.

Trilateration is one of the possible methods for estimating the location of robots among several others (e.g., odometry [28]). Triangulation consists in finding the position of the robot using the measurement of angles and the relationship between this and possible references available, and is commonly used in GPS technology, compasses, positioning systems using computer vision, etc. Unlike other localization methods, the trilateration uses the estimation of distances between the unknown position of a robot to others, which serve as a reference. When the estimation of location is made in the 2D plan, it is required at least three reference points for the robot to estimate accurately and uniquely its own location. This method can be generalized using *n* points of reference. In this case, the process is designated by multilateration. The derivation of the mathematical problem [29] assumes that, if the robot knows the distance between its location (unknown) and another robot whose position is known, then the robot is located at any point on a circle centered on the known robot position with radius equal to the known distance. Clearly, this information is not accurate enough. Nevertheless, if the robot knows the distance to a second robot, its location will be restricted to a point or an area between two points of intersection of both circles. The position of the robot will be perfectly set up if this robot knows the distance to a third party, then theoretically the robot's location will be restricted to the intersection point between the three circles. It is important to note that the method has associated errors which are discussed later.

Trilateration Experiment. In order to analyze the performance of a trilateration method, a new experiment was performed in an outdoor environment, without any obstruction, with three robots distributed into a triangle shape and one in the center, as shown in Figure 8. The robot at the center sends periodic messages to the robots at the vertices of the triangle, and in the reply message, the central robot extracts the RSSI value of those messages in order to estimate its own position. This procedure was repeated by increasing the distance to the robots in the vertices up to 10 m, with 1 m increments, and recording 30 samples per each increment.

In order to link the distance to the RSSI values of the different modules, we have used Equation (1), derived previously. As reported before, the solution of trilateration would be a point of intersection of three circles, but as seen in Table 1, in practical terms it does not happen that way. This is due to the distance measurements that have several associated errors. More specifically, using the estimation of distance with the RSSI, the localization process is noisy when implemented in real situations and, henceforth, the radii of the circles around the robots may vary.

To allow analysis of the experimental results as a whole, i.e. position errors involved in trilateration for each experiment, a method of least squares ellipses to 2D points was applied [30]. This allows examining the precision based on the area of the ellipse, i.e. dispersion in relation to the actual position. In addition, data accuracy can also be obtained based on the location of the center of the ellipse relative to the actual position. As shown in Figure 9, at 1 m distance the estimation exhibits superior precision and accuracy. However, from a distance of 5

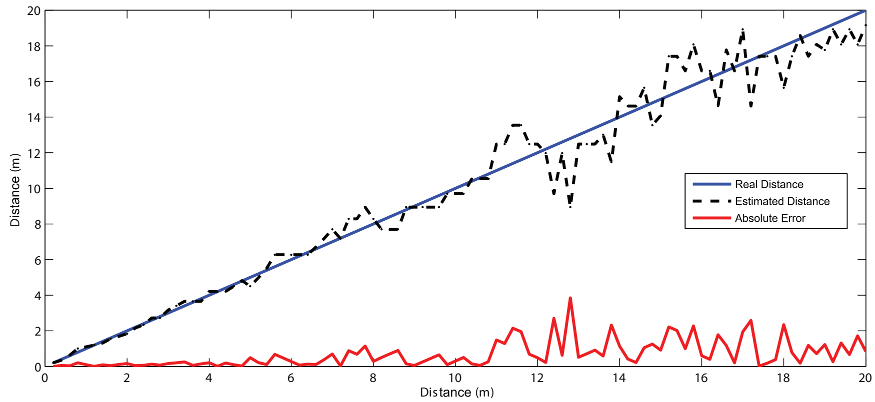


Figure 7 Error analysis in distance estimation. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

meters onwards, precision and accuracy begins to decrease, as can be seen with the increasing area of the ellipse and the distance from the center of the ellipse to the origin. The errors obtained can be caused by various sources related to the environment and the fact that signals face interference in real situations, influenced by temperature, humidity or obstructions, which has a direct impact on the signal strength [31]. Unfortunately, these interferences are beyond human control and may contribute to unexpected results. Moreover due to interference, the Xbee modules acquire different values of RSSI even when they are in similar situations. All these factors contribute to the trilateration error obtained.

Experiments with Mobile Nodes

The work developed and described in this paper has been used to study and evaluate important issues in swarm robotics, such as the deployment problem or to optimize the communication procedure between robots to perform collective intelligent behavior. Couceiro et al. [32] proposed the Robotic Darwinian Particle Swarm Optimization (RDPSO), an exploration algorithm for robotic swarms in unknown environments, mainly for post-catastrophic search and rescue events. In brief, the RDPSO approach is an adaptation of the PSO algorithm [33] to real mobile robots, in which five general features are considered: (i) an improved inertial influence based on fractional calculus concepts taking into account convergence dynamics; (ii) an obstacle avoidance behavior to avoid collisions; (iii) an algorithm to ensure that the MANET remains connected throughout the mission; (iv) a novel methodology to establish the initial planar deployment of robots preserving the connectivity of the MANET, while spreading out the robots as most as possible; and (v) a novel punish-reward mechanism to emulate the deletion and creation of robots.

Figure 8 Open outdoor environment used throughout the outdoor experiments, and robots disposition during the triangulation experiment. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Following this, an analysis of the architecture and characteristics of the RDPSO communication system has been conducted in order to improve the behavior and achieve a more scalable swarm system. Such improvements have been motivated by the need to use large teams of robots without significantly increase the communication overhead. As will be seen in this section, the communication overhead within a swarm of robots has been decreased just by adapting the Ad hoc On-demand Distance Vector (AODV) routing protocol [34], which is one of the mostly used reactive MANET routing protocol. In general, AODV exhibits good performance on MANETs, accomplishing its goal of eliminating source routing overhead. However, at considerably high rates of nodes mobility, it requires the transmission of many overhead packets.

To explore and compare the properties of the "regular" version of the RDPSO with the "optimized" RDPSO, several experiments were conducted in a real world scenario using a group of 15 eSwarBots. The eSwarBot consists of a small differential platform with an Arduino Uno processing unit, whose communication is ensured by an Xbee Series 2 module, described previously.

In the following paragraphs the focus is placed in the communication complexity, while the convergence of the RDPSO exploration algorithm is analyzed afterwards. For more details on the RDPSO algorithm, refer to [32].

Although the XBee *Series* 2 modules allow a maximum communication range of approximately 30 m in indoor environments, preliminary tests show that the connectivity starts failing above 10 m. To ensure the connectivity between robots,

Table 1 Estimated Robot Position by Trilateration With Increasing Distanced to Three References

Estimated Position (m)				Estimated Distance			
X	Y	Error (m)	Error (%)	Robot 1	Robot 2	Robot 3	Real Distance (m)
−0.06	0.15	0.16	16	0.67	1.03	0.93	1
0.12	0.34	0.36	18	1.48	2.04	2.21	2
−0.35	0.45	0.57	19	2.38	3.44	2.97	3
−0.69	0.68	0.97	24.50	3.2	4.85	3.96	4
−0.73	0.75	0.93	18.6	4.24	5.9	5.18	5
−1.73	1.34	1.52	25.33	4.85	7.56	6.69	6
−1.67	1.22	2.07	29.57	5.18	8.52	6.28	7
−2.22	1.64	2.75	34.37	6.69	10.71	8.03	8
−1.64	1.13	1.98	22	8.52	11.31	9.3	9
−3.01	1.25	3.26	32.6	9.56	13.29	9.57	10

The Real Position of the Robot was (0,0).

the received signal quality was used, as described in RSSI Measurement section.

We have conducted two groups of experiments in a 20×10 m indoor scenario to test two deployment algorithms (explained later on), each group with 20 trials. The exploration algorithm consists of dividing the whole team of robots in different swarms (i.e. sub-teams) identified by their RGB-LEDs colors, enabling an external viewer to identify each different sub-team. A minimum, initial and maximum number of 0, 3, and 4 swarms were used, thus representing an initial swarm size of $N_S = 5$ robots.

Figure 10a depicts the average packet delivery ratio (y axis) when swarms are formed by specific number of robots (x axis). As one may observe, there is a sharp decrease on the packet delivery ratio for the "regular" RDPSO when a swarm is formed by more than 10 robots, dropping down to approximately 65% for a maximum network load of 15 robots. By adopting the AODV routing protocol, the "optimized" RDPSO significantly decreases the number of exchanged messages, and robots are still capable of receiving more than 90% of the data even within a swarm of 15 robots. On the other hand, Figure 10b illustrates the routing overhead when swarms are formed by a specific number of robots. The routing overhead is represented by the ratio between the number of route discovery messages and the number of data packets. Once again, the "optimized" RDPSO clearly overcomes the "regular" one for larger population of robots. Even though the number of data packets is reduced due to the efficient way to share information between robots, the number of route discovery messages decreases more significantly, thus resulting in a smaller routing overhead for a larger number of robots.

The experiments clearly show the advantages of such an optimized strategy regarding the scalability of the algorithm, thus paving the way for future swarm applications of hundreds or thousands of robots.

Besides the communication complexity, we also study the convergence of the RDPSO algorithm by evaluating two autonomous and marsupial strategies for initial deployment in unknown scenarios in the context of swarm exploration: *Randomized Initial Deployment* (RID) and *Extended Spiral of Theodorus* (EST). These are based on a hierarchical approach, in which exploring agents, named *scouts*, are autonomously deployed through explicit cooperation with supporting agents, denoted as *rangers*. Such cooperation is once again enabled by the use of the existing Xbee modules on each different robot.

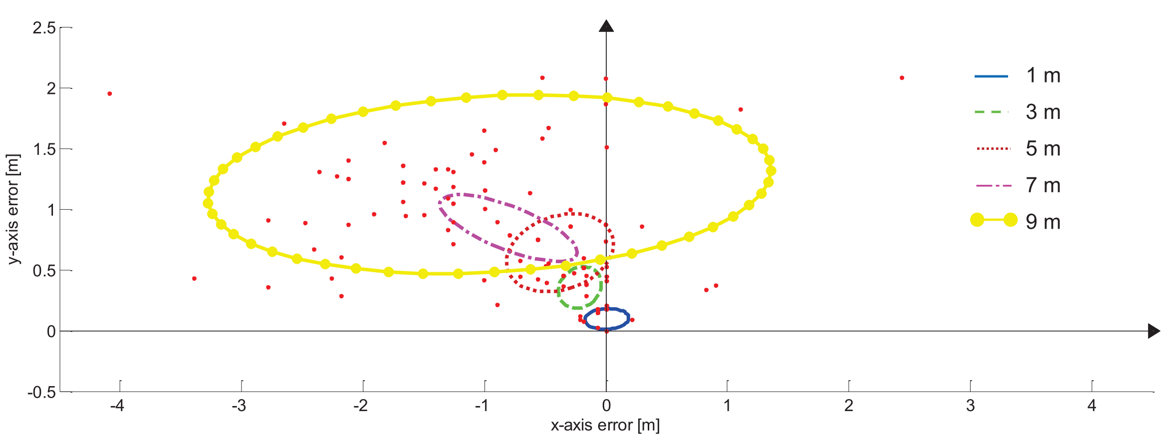


Figure 9 Evolution of the Trilateration Error with Distance. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

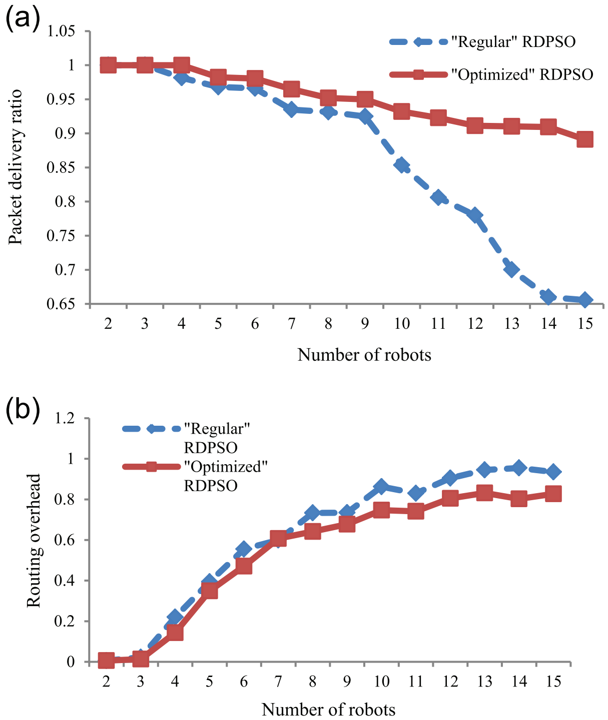


Figure 10 (a) Packet delivery ratio within robots belonging to the same swarm. (b) Routing overhead within robots from the same swarm. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

The *scouts* used in the experiments were *eSwarBots* (described previously) and the *Traxbot* platform [6] was adopted as *ranger*. In order to support 5 *eSwarBots* on the top of the platform, a conveyor kit has been built (Fig. 11). The marsupial deployment process is simple: First of all, *scouts* are manually loaded and equally distributed on the conveyor belt, i.e. *ranger* carrier system. After the ranger reaches the desired position to deploy a *scout* (according to one of the algorithms presented), the stepper motor conveyor is controlled by the *ranger* robot, to place the *scout* robot on the ground. After deploying each *scout*, the *ranger* informs the *scout* of its position by sending a message to it via the Zigbee MANET.

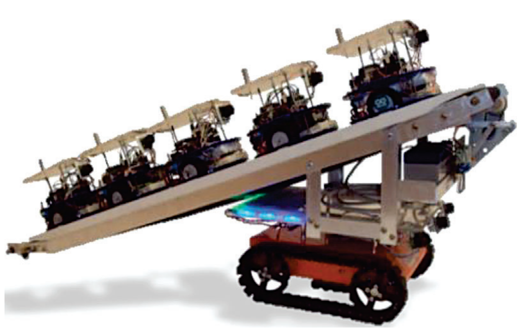


Figure 11 The TraxBot Conveyor Kit loaded with 5 eSwarBots. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

In the RID approach, *scouts* are successively deployed, one after another, by the same *ranger*, such that the pose of the *n*th robot always depends on the pose of the *(n-1)*th robot and the existence of obstacles in the path between them.

In the case of the EST, after deploying the second *scout*, the *ranger* will use the poses of both *scouts* to define a spiral center and determine the desired location for the next *scout*. The *EST* will not have a fixed central point x_0 of deployment. Instead, the central point will vary over time depending on the *scouts* previously deployed, *i.e.* number of deployed *scouts* and distance between them.

In both strategies, after deploying the whole team, the *ranger* broadcasts a message to start the mission. The message will be replicated by *scouts* inside its communication range, thus reaching all robots within the same sub-team. When the message is received, all *scouts* become aware that their teammates are already deployed in the environment and, consequently, they can start their mission. To illustrate the initial deployment process, Figure 12 presents a sequence of frames wherein 3 rangers deploy the whole population of 15 scouts using the EST strategy[5].

The effectiveness of the deployment strategies was evaluated using the RDPSO on swarms of *eSwarBots*, while performing a collective foraging task in a real-world scenario. Since the RDPSO is a stochastic algorithm, it may lead to a different trajectory convergence whenever it is executed. Therefore, the impact of the deployment strategy on the convergence of the algorithm was also evaluated by comparing the distributed spiral approach EST, with the random distribution RID.

The experimental environment contained two sites represented by an illuminated spot with different levels of light brightness. The main objective of the *scout* robots was to find the brighter site (optimal solution). All *eSwarBots* were equipped with LDR light sensors that allow finding the candidate sites. Figure 13 depicts the performance of the algorithm, with different initial deployment strategies. The colored zones between the solid lines represent the interquartile range (i.e. midspread) of the best solution considering the 20 trials for each different deployment strategy.

The EST deployment allowed a faster convergence of *scouts* towards the optimal solution. This is due to the larger distribution obtained with the EST approach that grants a larger diversity of solutions, thus yielding better results. On the other hand, such diversity is also responsible for having a larger interquartile range than the random deployment. Experimental results show that the exploration strategy converges sooner when using the EST deployment approach, demonstrating the importance of an informed choice of an initial deployment strategy in exploration tasks in unknown scenarios.

The experiments described in this section were crucial to understand how the communication between different members of the team of mobile robots by means of the ZigBee technology can be leveraged in real-world applications.

[5]https://www.youtube.com/watch?v=xs9dRPe6AuM

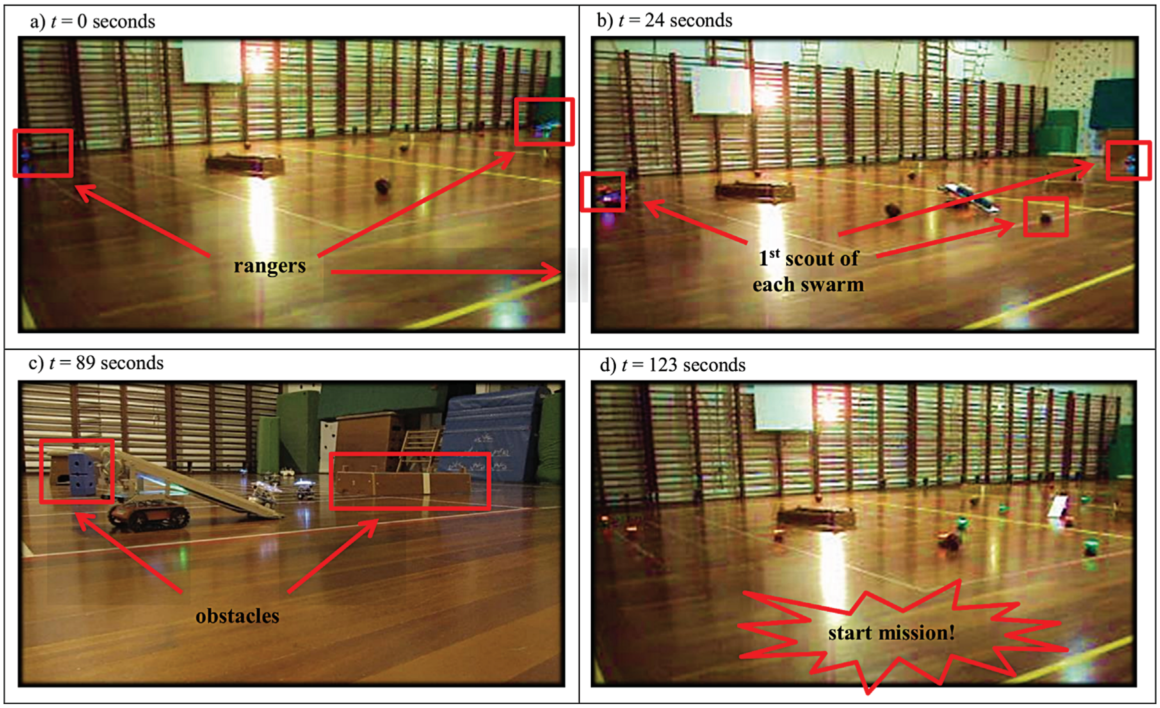


Figure 12 Frame sequence showing the EST deployment strategy on a population of 15 scouts and 3 rangers. (a) The population of scouts is initially divided into three groups – red, green and blue – each group loaded on top of a different ranger (one of the rangers is outside camera's field-of-view); (b) Each ranger randomly chooses the first position to deploy the first scout of each group; (c) The rangers will deploy the other successive scouts considering the previously deployed ones while avoiding obstacles; (d) After deploying all the scouts from one group, the ranger in charge of such deployment broadcasts a message to start the mission. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

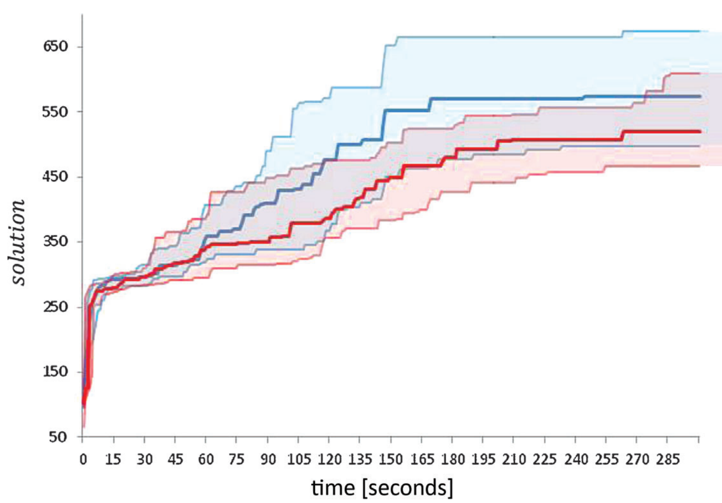


Figure 13 Performance of the RDPSO algorithm along time (x axis), under different deployment strategies. The y axis represents the intensity of the solutions. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

CONCLUSION

This article presented the implementation of ad hoc communication in Arduino- based educational mobile platforms using the ZigBee technology, through the integration and development of the features of the XBee Original Equipment Manufacturer Radio Frequency module.

The work developed was included and exposed to ROS, providing a very useful tool for one of the most known and widely used robotic frameworks. Furthermore, the relationship between the RSSI signal and distance was shown, as well as localization experiments based on the RSSI to estimate robots' positions through triangulation. Finally, real world experiments with mobile nodes were conducted, more specifically with robotic swarms, proving the potential of this technology in the coordination of large teams of mobile robots, where one of the biggest challenges is to provide a reliable and easy to implement communication.

It is the authors' wish that this paper may inspire other researchers, and serve as a guideline for the development and proliferation of mobile robot teams equipped with Zigbee communication modules.

ACKNOWLEDGMENTS

This work was supported by the CHOPIN research project (PTDC/EEA- CRO/119000/2010) and by ISR-Institute of Systems and Robotics (project PEst-C/EEI/UI0048/2011), all of them funded by the Portuguese Science Agency "Fundação para a Ciência e a Tecnologia" (FCT).

REFERENCES

[1] Doletech Corporation, D. R. Senik and Associates Inc, Wireless Technology Roadmap: 2006–2016 Mapping the Crucial Skills required to make Canada a Global Wireless Leader, Technical Report, 2007.
[2] C. Ramya, M. Shanmugaraj, and R. Prabakaram, Study on ZigBee technology. In Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT), vol. 6 (2011).
[3] D. Portugal and R. P. Rocha, Decision methods for distributed multi-robot patrol. In Proceedings of the 2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR'2012) (2012).
[4] M. S. Couceiro, D. Portugal, R. P. Rocha, and N. M. F. Ferreira, Marsupial teams of robots: deployment of miniature robots for swarm exploration under communication constraints, Robotica 32 (2014), pp 1017–1038. Cambridge University Press, October 2014.
[5] R. P. Rocha, D. Portugal, M. Couceiro, F. Araújo, P. Menezes, and J. Lobo, The CHOPIN project: Cooperation between Human and rObotic teams in catastroPhic INcidents. In Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR'2013).
[6] A. Araújo, D. Portugal, M. S. Couceiro, and R. P. Rocha, Integrating arduino-based educational mobile robots in ROS, J Intell Robot Syst 77 (2015), 281–298.
[7] J. Hu, Cooperation in mobile ad hoc networks, Wirel Ad Hoc Ne (2009).
[8] T. Ohta, T. Hashimoto, and Y. Kakuda, Self-organizing real-time service dissemination and collection using mobile agents for mobile ad hoc networks. In Proceedings of the 14th IEEE International Symposium on Object/Component/Service- Oriented Real-Time Distributed Computing Workshops, 2011, pp 199–206.
[9] J. Misic and V. B. Misic, Wireless personal area network: performance, interconnections and security with IEEE 802.15.4 Wiley, 2008.
[10] C. M. Chao and K. H. Lu, Load Awareness Multi-Channel MAC Protocol Design for Ad Hoc Networks. In Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous (2008) pp 36–43.
[11] Z. Wang, M. Zhou, and N. Ansari, Ad-hoc robot wireless communication. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics Vol. 4, pp 4045–4050, Washington D.C., October, 2003.
[12] R. J. Kennelly, IEEE standards for physical and data communications, Biomed Instrum Techn 30 (2003), 172–175.
[13] A. Wheeler, Commercial applications of wireless sensor networks using ZigBee, in IEEE communications magazine, Ad Hoc Sens Ne 45 (2007), 70–77.
[14] S. Safaric and K. Malaric, ZigBee wireless standard. In proceedings of the 48th International Symposium ELMAR - Multimedia Signal Processing and Communications pp. (2006), 259–262.
[15] K. Pahlavan and P. Krishnamurthy, Networking Fundamentals: wide, local and personal area communications, John Wiley and Sons, July (2009).
[16] ZigBee Alliance Website: http://www.zigbee.org/.
[17] Digi International Website: http://www.digi.com/.
[18] Digi International, XBee TM Series 2 OEM RF Modules - Product Manual v1.x.2x - ZigBee Protocol for OEM RF Module Part Numbers: XB24-BxIT-00x, (2007).
[19] E. Garcia, M. A. Jimenez, P. Gonzalez de Santos, and M. Armada, The evolution of robotics research: from industrial robotics to field and service robotics, In IEEE Robotics and Automation Magazine Vol. 14, pp 90–103, 2007.
[20] A. Pérez and J. Rossel, A roadmap to robot planning software development, Comput Appl Eng Educ 18 (2009), 651–660.
[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, ROS: an open-source Robot Operating System. In Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Workshop on Open Source Software (2009).
[22] J. Machado Santos, D. Portugal, and R. P. Rocha, An Evaluation of 2D SLAM Techniques Available in Robot Operating System. In Proceedings of the 2013 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR'2013).
[23] F. Thomas and L. Ros, Revisiting trilateration for robot localization, IEEE T Robot 21 (2005), 93–101.
[24] J. C. Giacomin and F. H. Vasconcelos, Measurement Quality of Signal Strength in the Wireless Sensor Network's Communications: a Physical Layer Approach, INFOCOMP J Comput Sci (2006).
[25] M. J. Segura, F. A. Auat Cheein, J. M. Toibero, V. Mut, and R. Carelli, Ultra wide-band localization and SLAM: a comparative study for mobile robot navigation, Sensors Vol. 11, 2035–2055, 2011.
[26] Y. Lin, P. Vernaza, J. Ham, and D. D. Lee, Cooperative relative robot localization with audible acoustic sensing. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005) pp 3764–3769 2005.
[27] Curve Fitting Toolbox: Fit curves and surfaces to data using regression, interpolation, and smoothing, Available Online: http://www.mathworks.com/products/cu rvefitting/.
[28] A. Martinelli, N. Tomatis, and R. Siegwart, Simultaneous localization and odometry self calibration for mobile robot, Auton Robots 22, 75–85, 2006.
[29] A. Fernandes, Comunicação Ad Hoc em Equipas de Robôs Móveis Utilizando a Tecnologia, ZigBee. M. Sc. and Thesis, Faculty of Science and Technology, University of Coimbra, September (2012).
[30] A. W. Fitzgibbon, M. Pilu, and R. B. Fisher, Direct Least-squares fitting of ellipses. IEEE Trans Pattern Anal Mach Intell 21 (1999), 476–480.
[31] Y. Zhang, Z. Fang, R. Li, and W. Hu, The Design and Implementation of a RSSI-Based Localization System, 5th International Conference

on Wireless Communications, Networking and Mobile Computing, (2009), pp 1–4.

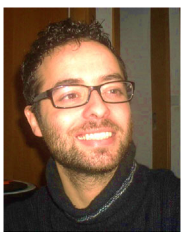
[32] M. S. Couceiro, R. P. Rocha, and N. M. F. Ferreira, Ensuring Ad Hoc Connectivity in Distributed Search with Robotic Darwinian Swarms, in proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR2011, Kyoto, japan (2011), pp 284–289.

[33] P. M. Oliveira, D. Vrančić, J. B. Cunha, and E. J. S. Pires, Teaching particle swarm optimization through and open-loop system identification project, Comput Appl Eng Educ 22 (2014), 227–237.

[34] C. E. Perkins and E. M. Royer, Ad-hoc on-demand distance vector routing, In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99), (1999).

BIOGRAPHIES

Amadeu Fernandes finished his MSc degree in Electrical and Computer Engineering at the University of Coimbra in 2012. In his dissertation, which focused on empowering teams of Mobile Robots with wireless communication skills using the ZigBee technology, he was an active researcher at the Institute of Systems and Robotics (ISR), University of Coimbra, Portugal. Currently, Amadeu works as an Entrepeneur at LFonTech, Ltd. based in Brockton, Massachusetts (USA).



Micael S. Couceiro obtained the BSc, Teaching Licensure and Master degrees in Electrical Engineering (Automation and Communications), at the Coimbra School of Engineering (ISEC), Coimbra Polytechnic Institute (IPC). He obtained the PhD degree on Electrical and Computer Engineering (Automation and Robotics) from the University of Coimbra (UC), under a PhD Grant from the Portuguese Foundation for Science and Technology. Over the past 7 years, he has been conducting scientific research on several areas, namely Robotics Computer Vision, Sports Engineering, Economics, Sociology, Digital Media and others, at the Institute of Systems and Robotics (ISR-UC) and RoboCorp (IPC). This resulted in more than 25 scientific articles in international impact factor journals and more than 50 scientific articles at international conferences. Besides being currently a researcher at ISR, he has been invited for lecturing, tutoring and organization of events (e.g., professional courses, national and international conferences, among others), both in the public and private domains. He is also the co-founder and CEO of Ingeniarius, Lda. in Portugal.

Dr. David Portugal completed his graduate studies at the University of Coimbra in Portugal with a MSc degree in Electrical Engineering and Computing in 2009. He then pursued a PhD degree from the same University, in the area of Robotics and Multi-Agent Systems, which he obtained in March, 2014. His main areas of expertise are cooperative robotics, multi-agent systems, simultaneous localization and mapping, search and rescue robotics, field robotics, meta-heuristics, graph theory and optimization. Currently, he is working as an Experienced Researcher in Citard Services Ltd. (Cyprus) and his research is mostly focused on human-robot interaction, cognitive modelling, sensor fusion, and the ageing well domain. He has been involved in several local and EU-funded research projects in Robotics and Ambient Assisted Living, such as CHOPIN, TIRAMISU, SocialRobot, CogniWin and GrowMeUp, and he has co-authored over 45 research articles included in international journals, conferences and scientific books.

João Machado Santos received the MSc degree in Electrical and Computer Engineering from the University of Coimbra on Sep. 2013, with a MSc dissertation entitled "SmokeNav - Simultaneous Localization and Mapping in Reduced Visibility Scenarios". Additionally, he held a research fellowship in the CHOPIN (Cooperation between Human and rObotic teams in catastroPhic INcidents) R&D project, at the Institute of Systems and Robotics (ISR), University of Coimbra, Portugal. Currently, he is pursuing a PhD degree at the School of Computer Science at the University of Lincoln (United Kingdom) and is involved with the STRANDS FP7 EU project. His main research interests are Mobile Robotics and SLAM, more specifically spatio-temporal environment representation, indoor robot exploration, and adaptive planning.

Rui P. Rocha completed his PhD degree on May 2006, at the Faculty of Engineering of the University of Porto. Between February 2000 and May 2006, he was a Teaching Assistant at the Department of Electrical and Computer Engineering, in the Faculty of Sciences and Technology of the University of Coimbra. Currently, he is an assistant professor at the Department of Electrical and Computer Engineering and a researcher at the Institute of Systems and Robotics, in the Faculty of Sciences and Technology University of Coimbra. His main research topics are cooperative Multi-Robot Systems, 3-D Map Building, Distributed Architectures and Intelligent Transportation Systems. He has been involved on several FP6 and FP7 European funded projects developed in consortium for the past few years and is currently the PI of a nationally funded research project about cooperation between human teams and robotics teams.