IET Research Journals

Journals
The Institution of Engineering and Technology

Robotic-aided Internet of Things: automated deployment of a 6TiSCH Network using an Unmanned Ground Vehicle

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Giovanni Valecce[1], Gianfranco Micoli[1], Pietro Boccadoro[1 3], Antonio Petitti[2], Roberto Colella[2], Annalisa Milella[2], Luigi Alfredo Grieco[1]

[1] Dep. of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy, Email:name.surname@poliba.it
[2] Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing, National Research Council, Bari, Italy ,Email: name.surname@stiima.cnr.it
[3] CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Politecnico di Bari, Bari, Italy.

Abstract: The automated set up of Internet of Things (IoT) systems in industrial environments is an open challenge at the forefront of networking and robotics domains. In this work, a robotic-aided deployment system is proposed and experimentally tested with reference to the Internet Protocol version 6 (IPv6) over the Time Slotted Channel Hopping (TSCH) mode of IEEE 802.15.4 (6TiSCH) technology. To this end, a design methodology has been conceived to attain the set up of a fully connected IoT network in a target area, based on the measured Received Signal Strength Indication (RSSI) of wireless links, required spatial sensing resolution, and number of available IoT nodes. To demonstrate the effectiveness of the proposed methodology, an experimental testbed has been set up, consisting of an Unmanned Ground Vehicle (UGV) that automatically deploys an IoT network in a laboratory environment. Experimental results clearly show that the UGV is able to deploy a fully connected 6TiSCH network that matches spatial resolution requirements, highlighting how the proposed policy affects the position of IoT nodes release points.

1 Introduction

Nowadays, IoT technologies are considered as the angular stone of the Industry 4.0 digital innovation process [1]. In this context, one of the key challenges to face is the automated deployment of industrial IoT networks, especially in harsh and/or large scale environments [2]. Robots can effectively face this challenge thanks to their inherent capability to execute tedious and repetitive tasks [3], such as the placement/maintenance/replacement/configuration of multiple IoT nodes (known as motes) also in wide environments.

When wireless technologies are used, the optimal placement of a finite number of motes within a target area entails several competing requirements that should be properly traded off. From one hand, based on the monitoring application, a minimum number of motes per surface unit should be deployed to pursue the desired spatial sampling resolution of the phenomena of interest [4]. On the other hand, the placement of the different motes should consider the radio signal strength, which is very hard to predict in industrial environments, in order to result in a fully connected IoT network [5]. In addition, upon deployment, each mote should be fully configured in order to make it able to exchange data with the rest of the network. The configuration involves all layers of the protocol stack. Therefore, the robots in an automated deployment system should be able to: self locate themselves within the area of interest, sense the radio signal strength, and physically release the motes according to ad hoc decision rules.

The robotic-aided IoT deployment systems proposed so far [4]-[11] (more details about related works are discussed in Sec. 2) implicitly or explicitly assume that radio communications between motes are omni-directional and symmetric in all directions. This assumption is very far from reality since shadowing, scattering, and multipath propagation completely break the circular symmetry of any real communication process, especially in industrial environments, as very well known from basic digital communication theory and practice [12].

To bridge this gap, a robotic-aided IoT deployment system is proposed hereby, which uses an unmanned vehicle for the automated deployment of a 6TiSCH network, based on actual measurements of radio signal strength. The 6TiSCH technology has been considered as a representative example of industrial IoT stack, but the methodology proposed hereby can be adapted, with appropriate customizations, to any IoT protocol architecture. The proposed approach, in summary, can be illustrated through the following points:

- an UGV moves along a pre-defined trajectory and spans a given area of interest to automatically deploy IoT devices.
- the starting point of the path is located in close proximity with a network coordinator, which operates as a local base station for the network to be deployed, hereinafter referred to as ground network.
- the UGV carries an onboard patrolling network made of an onboard coordinator and a set of nodes to deploy. Moreover, the UGV is equipped with a probing node, connected to the ground network, that senses the surrounding area in search of radio activity.
- the UGV releases IoT nodes while it moves along its trajectory, based on the actual RSSI of wireless links, target coverage area, required spatial sensing resolution, and number of available IoT nodes.
- Once released, the IoT device is added to the ground network and becomes able to monitor environmental parameters.

The main contribution of this work is to instruct an unmanned vehicle to: (i) dynamically map the environment in terms of radio signal propagation, (ii) verify the most convenient release points for each IoT device, and (iii) deploy and configure these network entities. The envisioned solution is able to constantly monitor the environment searching for the optimal position to release the IoT device while keeping trace of the number of release events. After each release event, IoT devices are able to autonomously execute synchronization tasks, while creating and maintaining stable links. In this way, an automated deployment, based on the actual connectivity parameters, becomes possible. An experimental campaign has been carried out in a laboratory environment, focusing on the release policy of the IoT nodes affected by the RSSI value during the area patrolling. The obtained results showed that the adoption of a release policy sensibly affected the actual deployment, thus granting network connectivity.

The remainder of this work is organized as follows: Section 2 is dedicated to a detailed study of the state of the art on deployment algorithms revealing some weaknesses in the solutions proposed so far. Section 3 illustrates the motivations of this contribution and introduces the experimental setup that will be used later in the manuscript. Section 4 presents the proposed solution for the automated deployment of 6TiSCH network, and Section 5 discusses the outcomes of the experimental campaign. Finally, Section 6 summarizes the key achievements of this contribution and draws future research.

2 Related Works

Prior works related to the deployment of IoT networks [4][5] rely on the theoretical assumption of omnidirectional propagation of electromagnetic waves, modeled through two peculiar parameters: the sensing range R_s and coverage (communication) range R_c. The sensing range of a sensor borders the region where every event that takes place in this region can be detected by. With reference to a given IoT node, its communication range defines a region such that it can communicate with any other IoT node located in this region [13].

Starting from these premises, different strategies are proposed in literature to optimize the automated deployment of IoT nodes. In particular, in [4] a methodology is proposed to attain the full coverage of a target area with the minimal number of IoT nodes, while avoiding dead-end traps. To this end, the robot carries a set of IoT devices that will be released gradually, assuming that the geographic boundaries of the monitored region are known. The release process is driven by the communication range R_c, which is assumed to be at least $\sqrt{3}$ times higher than the sensing range R_s.

In [6] an algorithm is proposed, based on snake-like robot mobility patterns, to cover an orthogonal region that is not known a priori. In this case, given L and W, i.e., the length and width of the monitored region, the sensing range R_s is used to derive the ideal number (N_{ideal}) of IoT nodes to be deployed, based on geometrical considerations:

$$N_{\text{ideal}}(W, L) = \left\lceil \frac{W * L}{\text{Area}} \right\rceil = \lceil \frac{W * L}{3\sqrt{3}R_s^2/2} \rceil. \tag{1}$$

In [7], instead, a deployment algorithm is designed to reduce the energy consumption of the IoT network. In particular, a LAyered CIrcular Deployment (LACID) is defined with a geographical subdivision of the network field into circular concentric crowns. Choosing the proper number of nodes to place in every crown, guarantees energy balance among all crowns. This choice is accomplished by accounting for routing and communication activities.

The sensing ranges of sensors can be all equal or heterogeneous [14]. [8] takes into account the heterogeneity of IoT devices in terms of sensing ranges providing an adaptive deployment design. Given a random number of network devices already deployed, the author presents a theory based on virtual forces acting on the nodes. These forces guide the sensors to their suitable positions in order to enhance the sensing coverage. Each node represents a source of force from the others. If two devices are too close, they apply repulsive force for separating themselves.

Further studies in [11] consider the robot as a *sentient unit* suitable for data collection, and the sensors as simple *markers* useful to point out the areas already patrolled by the robot, or as *geographical tags* that suggest the direction of the next shift. Given an area of interest already covered by randomly deployed sensors, [5] suggests a robot-driven nodes relocation strategy to obtain sufficient coverage and network connectivity. In both cases, communication aspects are left unspecified without considering the radio communications variability. In this type of re-deployment strategies, each robot can release some of the on-board IoT nodes on the field and recover the redundant/faulty ones from the area of interest. In a multi-robot approach, the network redeployment and the motes number optimization are critical issues, especially if combined with robot movements and data exchange procedures.

The approaches discussed so far are heavily grounded on the hypothesis of circular symmetry of the communication range: unfortunately, as very well known from the classic literature on wireless communication systems, effects like shadowing, scattering, and multipath propagation can significantly affect the applicability of this hypothesis [12, 15], thus motivating novel approaches to the development of IoT systems, based on actual measures of the radio signal strength as opposite to estimates derived from ideal models.

In this direction, [10] suggests that, during the deployment, any robot could measure the RSSI, and release the IoT devices when a specific threshold is reached. Unfortunately, no experimental evidence on the effectiveness of this strategy is provided.

To bridge this gap, in this work a new design methodology is proposed for automated deployment of IoT systems based on actual measurements of the radio signal strength and an experimental evidence of its effectiveness is provided in a laboratory environment.

3 Problem statement and experimental setup

Without lack of generality, this paper assumes a grid-based deployment to reach the coverage of an area of interest. This assumption is very commonplace because it nicely fits the monitoring requirements of many application domains, thus including monitoring vines in a vineyard or trees in a commercial plantation or reforestation project, studying traffic or pollution levels on city streets, measuring humidity and temperature at regular intervals on library shelves, acoustic testing at each of the seats in a theater, and so on [16].

To this aim, the area has to be discretized, identifying geometric patterns, according to some homogeneous characteristics, such as: surface color, physical appearance, and crop ordering. This preliminary assessment leads to the definition of the so called *Elementary Sensing Area* A_s, which represents the elementary square cell composing the grid. The overall sensing area can be defined as $A = A_s \cdot n$, where n is the total number of cells to monitor. Since the application requirement is to deploy at least one mote per sampling unit, $\pi * R_s^2$ represents the area of the inscribed circle into elementary sensing area square, where R_s is the sensing range of an IoT node.

Considering a real environment deployment, an a-priori positioning of the IoT nodes into sensing areas center, together with the unpredictable real shape of communication range [17], can lead to a not fully connected topology. Indeed, radio propagation variability could isolate some IoT nodes or IoT network portions. As a consequence, the data gathering process could be affected too.

In order to define an automated robotic-aided deployment strategy, the following questions need to be answered:

1. Where the automated deployment strategy should start from?
2. Which trajectory should the UGV follow?
3. How to choose the best release point for each IoT node to be deployed?

With reference to our approach, the following answers can be provided.

As regards to question 1, given a pre-defined sampling grid, the central position is occupied by the first IoT device with coordination role: this choice will reduce the path length from motes to the coordinator. The shorter the path, the higher the reliability and timeliness of data communications.

With reference to question 2, the UGV trajectory has to be minimized in order to lower the energy footprint of the system. This principle leads to the definition of an expanding spiral-wise path that starts from the center of the area of interest and then reach the farthest boundaries.

Question 3 can be answered by defining a compromise decision policy between connectivity and coverage requirements. A dynamic IoT network deployment strategy can be aimed at maximizing both connectivity and reliability. To be effective, one, or eventually more, parameter(s) can be addressed. Indeed, the conducted experimental campaign has been carried out by constantly monitoring the received radio signal strength to decide whether to release the IoT device or not. More details about the proposed methodology will be provided in next section 4.

The experimental setup has been realized in the Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing (STIIMA)-National Research Council (CNR) of Italy Mobile Robot Laboratory, where the Advanced indoor Robotic test Environment for Networks of Autonomous vehicles (ARENA) infrastructure is installed. In particular, an high-accuracy motion capture system (VICON) monitors the environment. The main functionality of VICON is the tracking of the pose of rigid bodies for localization and position control algorithms (closing the control loop or providing the ground-truth) [18].

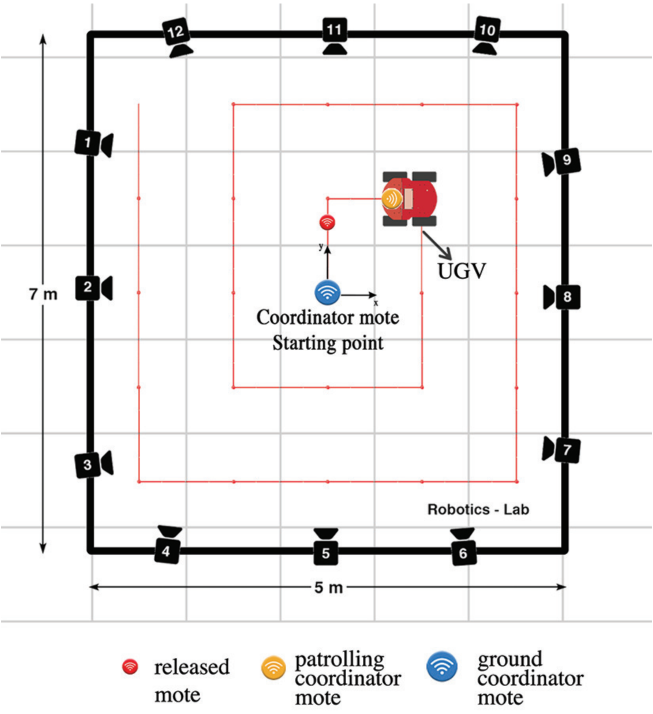


Fig. 1: Experimental scenario representation with a detail on the chosen path.

Figure 1 describes the envisioned setup and the chosen path[*]. It is mainly composed of a robotic unit and IoT devices. The former is a Pioneer[†] 3-AT, an UGV equipped with four-wheel drive. The robot is powered by three 12 V batteries, replaceable through the rear compartment, and able to grant a maximum of 4 h of autonomy. The robot is equipped with an added high-level PC, an Acer Veriton N4620G, and its own embedded low-level on-board computer. The high-level pc of the UGV runs Linux Ubuntu 16.04 Long Term Support (LTS) operating system and Robot Operating System (ROS)[‡] *'kinetic'* which executes the following tasks:

- interface towards the low-level control system of the vehicle;
- interface towards laboratory's internal localization system;
- execution of custom developed robot control software.

In particular, ROS is a group of tools and open source libraries that can be employed in the development of robotic applications. ROS enables communication between processes in a cluster composed of various computers, which can be found on many modern robots, or tiny boards such as the Raspberry Pi. Those processes are called nodes, and can perform many actions regarding the robot. The Raspberry Pi[§] 2 Model B was used to manage the ground coordinator used for the experiments. The operating system in use is "Raspbian Jessie"[¶], a light weight Debian-based Unix system.

The IoT devices involved in the experimental setup are the Telos rev B, also known by the name of TelosB[*], a well known hardware platform that has been used in both academic research activities and industrial deployments over the latest ten years [19][20][21][22]. OpenWSN [23][24] is an open source software solution for IoT devices [25]. It is an implementation of the Internet Engineering Task Force (IETF) 6TiSCH protocol stack [26][27] and relies on routines specifically written to provide two set of functionalities: (i) executing communication tasks through IoT devices and (ii) monitoring the IoT network activities while granting its connection to the whole Internet. To these aims, OpenWSN[†] is composed of (i) a firmware part, which runs on the motes, and (ii) a software part, in charge of real-time network monitoring tasks and gateway functionalities. For the sake of clarity, hardware components involved are summarized in Table 1.

Table 1 Hardware components involved.

Domain	Item	Description	Quantity
Patrolling Network	mote	TelosB	6
	coordinator	TelosB	1
Ground Network	mote	TelosB	1
	coordinator	TelosB	1
	probe	TelosB	1
Motion Capture System	Infrared Cameras	Vicon Bonita 10	12
	Workstation	Hp Z440	1

4 Proposed Solution

Starting from an equal subdivision of the area of interest in square cells, a non-deterministic and automated deployment strategy through UGV is developed hereby (Figure 1). In details, the proposed solution defines:

- a deployment algorithm that drives the release of IoT nodes in the field based on several real-time measurements;
- a network switching function that configures released IoT nodes in order to let them join the ground network;
- release policy criteria that rules the deployment algorithm.

Two entities are identified in the operational scenario (Figure 2): (1) the ground coordinator node, connected to a Raspberry Pi, and (2) an IoT node connected to the robot, hereinafter referred to as probe node, sensing the radio signal quality. In Figure 3 it can be noticed that probe node (A) is positioned in close proximity to the next IoT node to be released (B), and thus it will sense the same RSSI as the node in the holder.

4.1 Deployment algorithm

In order to patrol the area of interest and search for the best release points for IoT nodes, a robot motion algorithm has been developed,

---

[*] *Which will be discussed in details in Section* 4.

[†] *http://www.mobilerobots.com/Libraries/Downloads/Pioneer3AT-P3AT-RevA.sflb.ashx*

[‡] *www.ros.org/*

[§] *https://www.raspberrypi.org/products/raspberry-pi-2-model-b/*

[¶] *https://www.raspberrypi.org/downloads/raspbian/*

[*] *http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf*

[†] *https://openwsn.atlassian.net/wiki/spaces/OW*

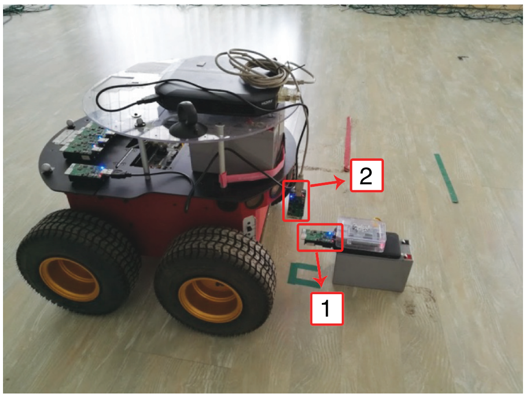


Fig. 2: Robotic Unit close to the ground coordinator (1) at the beginning of the patrolling mission, monitoring the RSSI value via the probing node (2).

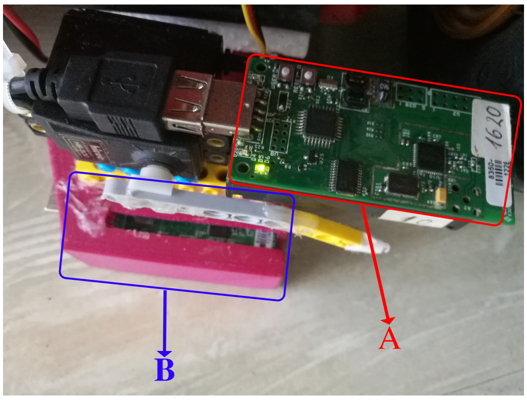


Fig. 3: (A) Probe IoT node and (B) release candidate IoT node

accounting for network connectivity issues. The motion algorithm drives the UGV to draw a connected acyclic path between the waypoints placed at the center of the elementary sensing areas. Two important requirements are identified to fully cover the target area:

•execute a progressive distancing of the UGV from the coordinator;
•minimize the path run by the robot;
•strengthen the quality of radio links in the IoT network.

The first one considers the multi-hop topology in IoT networks. With a progressive distancing from the central coordinator IoT node, the number of possible nodes directly connected to it increases. This configuration magnifies the coverage capabilities of the coordinator in a simple start topology.

The second one takes into account the need for minimizing the path that connects all waypoints (cell centers, hereafter referred to as goals). In this case, classic shortest path algorithms, such as Traveling Salesman Problem (TSP) [28] have not been considered, because, given the equal subdivision of the area into cells, the minimum path between one goal and the other will always be equal to l meters, i.e. the Euclidean distance. The chosen path is shown in Figure 1: the waypoints are represented at the center of each cell to be visited and the path between them starts from the center of the area of interest, proceeding indifferently towards one of the orthogonal directions. It would not be convenient consider moving to a $45°$ goal from the current position since the path to this would be equal to $\sqrt{2}l$ meters. Assuming that the coordinator is already positioned at the center of the area, total distance run by the robot will be n times the length of the cell side, where n is the number of waypoints.

The third requirement implies the need for a reliable connectivity between IoT nodes, which translates into a good link quality among neighboring nodes. An important indicator of the link strength is the RSSI, which represents the intensity of the received Radio Frequency (RF) signal. It has been shown that RSSI has small variations over Link Quality Indicator (LQI) for any link over time, suggesting that RSSI on a single packet is a good estimate of the average RSSI for many packets exchanged. RSSI values larger than -87 dBm indicate a good link quality [29][30].

Starting from these premises, and considering a mote A to be released, the robot will patrol the sampled area and release the mote A in the position along the path where the RSSI is just above a critical condition for network formation. To this end, a threshold value for the RSSI is defined and a dedicated algorithm conceived that forces the robot to release a mote in a point when the threshold is reached. The threshold is set to -70 dBm*.

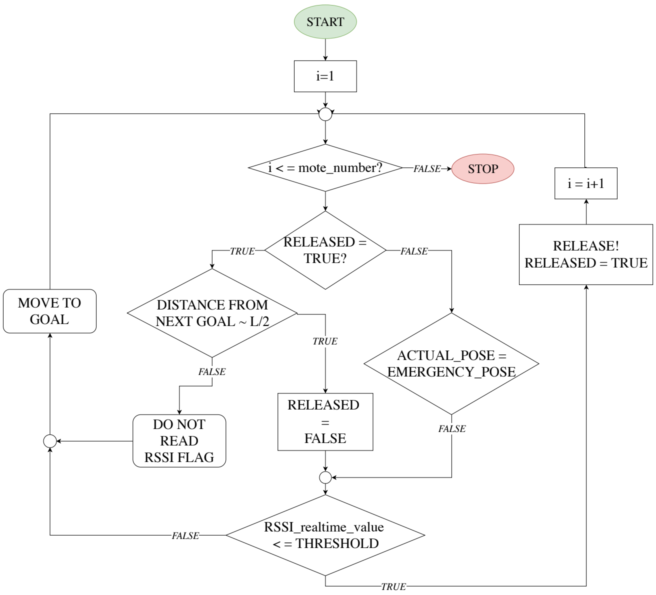


Fig. 4: Deployment algorithm

The designed algorithm is shown in Figure 4. Given n motes to be released in the area of interest, they will correspond to n elementary sensing areas. During the robot movement, the algorithm executes two checks:

•verifies that there are still IoT nodes to deploy. In case they have been already released, the algorithm ends and the robot stops its walk.
•prevents the release of multiple nodes in the same cell: if a mote has been already released in the correct cell, the robot will keep moving towards the next goal, without releasing any mote until the current cell is left. Indeed, it is critical for the robot to recognize when a new cell is approached. For this purpose, given l the length in meters of the square side of the sampling unit, when the distance to the next target is equal to $l/2$, the robot resets the flag.

During the movement, the robot can deploy a new IoT node whenever the threshold value for the RSSI is met. However, if the RSSI data read by the probe node never falls below the threshold, it is mandatory to induce a forced release of one mote per cell, to comply the application requirements. This is called *emergency release*. To this end, two candidate positions are chosen as *emergency release* place: in the goal (i.e., in the middle of the cell) or just before the end of the cell (i.e.,close to the border of the cell).

The algorithm foresees a preliminary control of the number of released IoT devices. This quantity has to be lower than the total amount of available motes. Known that the release criteria foresees one IoT device released for each cell, the total amount of IoT device

---

**The reasons that will be explained in Section 5.*

represents the number of elementary sensing areas to be explored by the UGV. Once the condition is verified, the robot begins to move. While crossing an elementary sensing area, it is of great importance that it does not release multiple motes within the same area. From a coding perspective, a boolean variable, i.e. *released* flag, is used in order to keep trace of the event. Where a release event has occurred, the deployment procedure is temporarily disabled and the robot keeps moving until it crosses the border of a new cell. When moving in a cell where the release has not been completed yet, the probing IoT device onboard of the UGV keeps listening to the RSSI value until reaching two alternative conditions: the specified RSSI threshold value is met or the robot reaches the specified emergency pose. In the former case, if the RSSI threshold value is reached before the appointed release position, the IoT device is released. Otherwise the mote will be forcibly released in the aforementioned emergency pose. The procedure is handled by updating two variables: the aforementioned *released* boolean flag, thus showing the successful release event of a mote in the current cell, and the counter variable i, used to indicate the total amount of released motes. It is worth specifying that, given the one-to-one relationship between the mote to be released and the cells to be visited, the indication of the total number of released mote also indicates the total number of visited cells. The algorithm proceeds cyclically until the cells to be visited are over.

4.2 *Release and network switching primitives*

During the unmanned patrolling, the designed algorithm can execute two different actions in order to complete the deployment of the ground network:

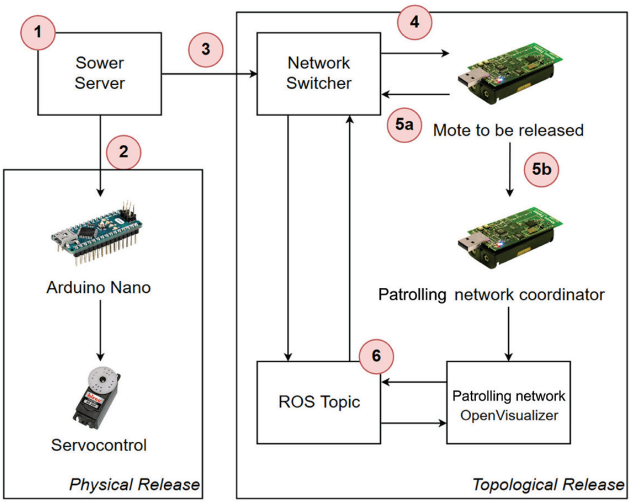
•Physical release: the mote is expelled from the *holder* in which it was initially located. This action is accomplished through a ROS node called *sower_server* used as an interface for the whole release mechanism (Figure 5). As soon as the request is received, the physical release is triggered ejecting the mote from its holder, thus initializing the topological release.
•Topological release: the mote needs to disconnect from the *patrolling* network and connect to the so called *ground* network. To this end, a network switching algorithm had to be taken into account. The proposal in [31] has been customized as to fit the requirements and work in the reference set-up environment.

Two kind of release events are considered in the proposed algorithm: on-the-move release and forced release. These policies can be triggered whenever the RSSI threshold is met or to force the deployment of at least one mote per elementary sensing area. For each of them, further details are provided in the following Sub-Sections.

4.2.1 On-The-Move release: While the robot patrols the area, the probe node senses the ground network radio activity, looking at the best perceived RSSI value from IoT nodes. When threshold is reached and the conditions set out in 4.1 are valid, the IoT node is released.

4.2.2 Forced release, goal position: In this first *emergency release* strategy, the robot has to release almost one mote before reaching the goal, i.e. the center of the elementary sensing area. If no mote has been released in the current cell, a forced release is required here.

4.2.3 Forced release, border cell position: The second strategy (which can be used as an alternative to the first one) forces the release at the end of the current cell. The current position of the robot is monitored by calculating the distance from the next target when it is within a range of $l/2 + margin$ and $l/2$ will indicate the presence of the UGV at the edge of the current cell during the movement. The spacial tolerance identified by $margin$ value (Figure 6) is practical to give the robot the necessary space to stop and release just before entering the new elementary sensing area.



① Sower Server node receives a request

② Mote is released from support using servocontrol

③ Network switching request is queued

④ Request to CSwitcher is sent for physically released mote

5a CoAP Response is sent and data published to topic

5b OBN coordinator notifies OpenVisualizer of mote release, data published to topic

⑥ Topic sends cleanup commands to OpenVisualizer and Network Switcher

Fig. 5: Manual Mote Sower architecture and single request elaboration steps.

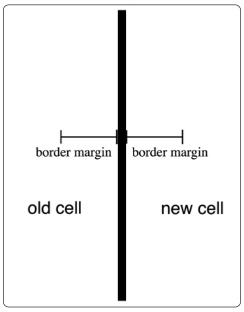


Fig. 6: Border margin and cell border

5 Experimental performance evaluation

The deployment algorithm has been functionally evaluated together with the release procedures considering different strategies.

5.1 *Experimental environment and parameters*

In order to create a test setup that can be scaled up to a real environment, the transmission power of the radio modules on the TelosB has been reduced to the minimum value*. With this setting, a preliminary test was conducted: reach the minimum value of RSSI that caused link disconnection between a nodes couple. Several RSSI values have been registered by gradually increasing the distance between

---

**2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. C) - http://www.ti.com/lit/ds/symlink/cc2420.pdf* a pair of IoT devices forming a simple network. An average value of -68.7 dBm was obtained. Thus, a -70 dBm threshold for releasing decision has been set.

Two set of experiments were made: the first with *emergency release* position at the goal, and the second in proximity of the border of each cell. The deployment campaign provided 10 repetitions for both strategies with 7 released motes and, consequently, 7 explored cells, always starting the patrolling from the central one, in which the IoT *ground* network coordinator has been preliminarily positioned. The experimental setup foresaw:

- the ground network coordinator connected to Raspberry Pi;
- probe mote connected to the robot on board PC;
- patrolling network coordinator mote;
- 7 mote to be released.

A 3D printed PolyLactic Acid (PLA) structure is used to stack the motes before the release (Figure 7). They are pushed out by a servo-controlled mechanical arm. There is a dedicated Wi-Fi network for Raspberry Pi, the robot on-board PC and a notebook for remote monitoring.

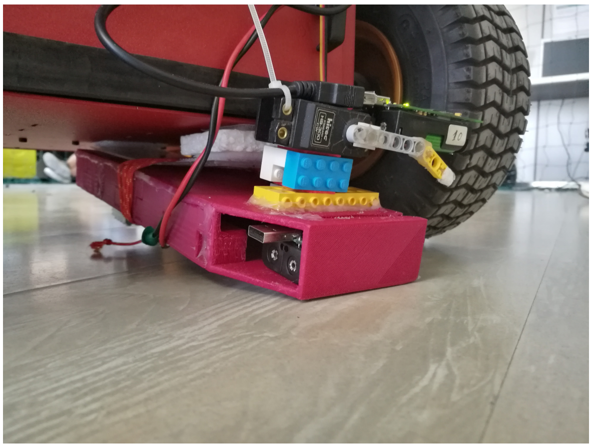


Fig. 7: PLA stack

The following data have been retrieved during each node release:

- Best RSSI perceived by the *probe mote*;
- Distance from goal;
- Release position.

5.2 *Experimental results*

The study focuses on the characterization of a dynamic deployment algorithm expecting marked differences about the release position of IoT nodes, compared to those suggested by theoretical considerations (i.e. the center of the elementary sensing areas). To this end the obtained experimental data are: (i) average positions of the released motes, (ii) mean distance from goal, and (iii) mean perceived RSSI value during the release process. The following subsections present the results collected, with respect to the different deployment strategies.

5.2.1 Emergency release: Goal: Table 2 reports the obtained results setting the goal as last resort position for releasing one mote in the current elementary sensing area. $\tilde{x}$ and $\tilde{y}$ represent the average release coordinates and x_i and y_i the goals coordinates. Assuming that $\tilde{d}$ and $\tilde{R_m}$ are the average distance from the goal and the best perceived RSSI average value, Table 3 shows the results obtained.

5.2.2 Emergency release: cell border: In a similar fashion to what previously described, the cell border release condition has been verified. The results are reported in Figure 11 and Table 4, as for release positions, Figures 12 -13 and Table 5 as for average distance and average distance from ideal point for the released motes.

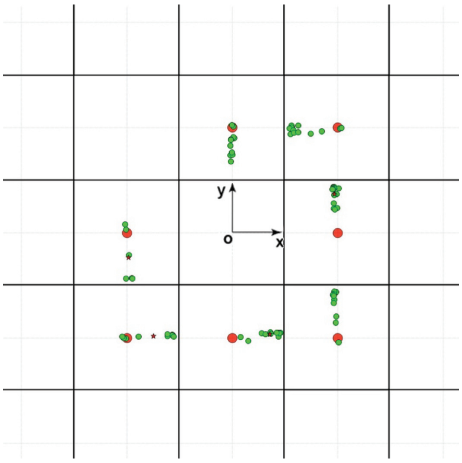


Fig. 8: Release positions (green) - Goals (red). - Goal as emergency release point

Table 2 Average release positions, goal coordinates and deviation per axis - goal as emergency release point.

GOAL	$\tilde{x}$[m]	$\tilde{y}$[m]	x_i[m]	y_i[m]	$\lvert x - x_i \rvert$[m]	$\lvert y - y_i \rvert$[m]
A	-0.004	0.853	0	1	0.004	0.147
B	0.672	0.979	1	1	0.328	0.021
C	0.968	0.369	1	0	0.032	0.369
D	0.974	-0.66	1	-1	0.026	0.34
E	0.354	-0.964	0	-1	0.354	0.036
F	-0.745	-0.982	-1	-1	0.255	0.018
G	-0.986	-0.233	-1	0	0.014	0.233

Table 3 Average release distance and perceived RSSI - goal as emergency release point.

GOAL	$\tilde{d}$(m)	$\tilde{R_m}$(dBm)
A	0.2	-72
B	0.379	-73.92
C	0.395	-79.42
D	0.363	-73.67
E	0.357	-76
F	0.274	-73.83
G	0.335	-73.3

Table 4 Average release positions, goal coordinates and deviation per axis - border as emergency release point.

GOAL	$\tilde{x}$[m]	$\tilde{y}$[m]	x_i[m]	y_i[m]	$\lvert x - x_i \rvert$[m]	$\lvert y - y_i \rvert$[m]
A	0.075	0.88	0	1	0.075	0.12
B	0.744	1.803	1	1	0.256	0.803
C	0.966	0.323	1	0	0.034	0.323
D	0.966	-0.641	1	-1	0.034	0.359
E	0.342	-0.955	0	-1	0.342	0.045
F	-0.741	-0.897	-1	-1	0.259	0.103
G	-0.966	0.079	-1	0	0.034	0.079

5.2.3 Insights: Figure 8 and 11 show the detected positions for all the release events. For both strategies, the outcome of the experiment seems to confirm the main hypothesis of the work. Indeed, in most cases, releasing the node in the center of the cell would not have coincided with the best conditions of RSSI data. This can have a remarkable consequence on radio links quality and on the whole network reliability.

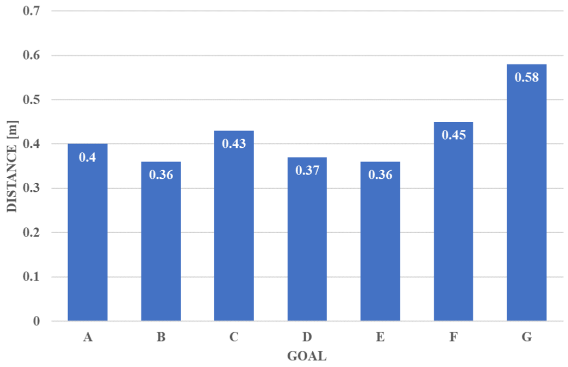


Fig. 9: Distance from goal during the release process - goal as emergency release point.

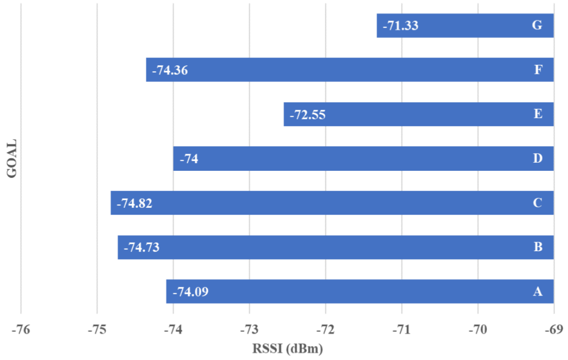


Fig. 10: Average RSSI perceived during the release process - goal as emergency release point.

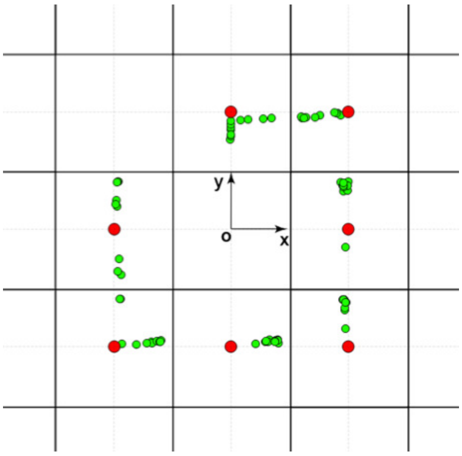


Fig. 11: Release positions (green) - Goals (red). - border as emergency release point

6 Conclusions and future research

This work presented a robotic-aided approach for automated IoT network deployment in environmental monitoring applications. In particular, the proposed system features an UGV moving all around a certain area of interest, while releasing IoT devices in conveniently detected positions. Experimental tests were presented, demonstrating that the envisioned policy allows for effective network deployment and monitoring of environmental parameters.

As for future research, several directions can be followed. First of all, an extended multi-robot approach to the deployment problem is highly recommended. In such conditions, it could be possible to effectively test scalability, reliability and providing re-deployment capabilities to the envisioned solution. Moreover, the presence of obstacles could not only lead to a trajectory change for the UGV but also impact radio propagation. An outdoor experimental campaign is necessary to test the algorithm and its possible revisions in real environments. Choosing a different positioning technology (e.g. Global Position System (GPS)) will be required and also radio signal propagation could be affected by environmental variables.

Table 5 Average release distance and perceived RSSI - border as emergency release point.

GOAL	$\tilde{d}$(m)	$\tilde{R_m}$ (dBm)
A	0.4	-74.09
B	0.36	-74.73
C	0.43	-74.82
D	0.37	-74
E	0.36	-72.55
F	0.45	-74.36
G	0.58	-71.33

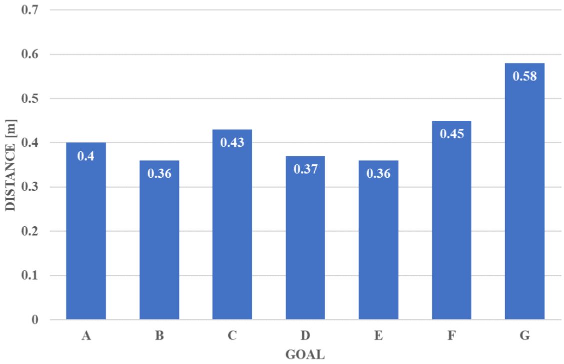


Fig. 12: Distance from goal during the release process - border as emergency release point.

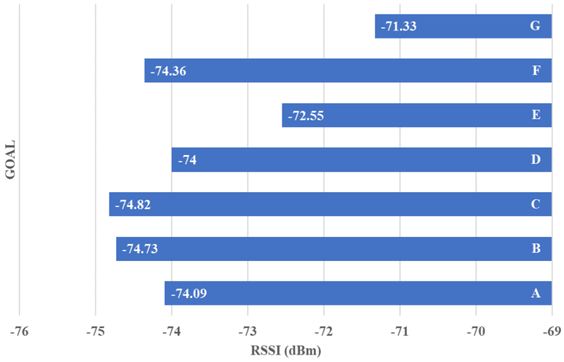


Fig. 13: Average RSSI perceived during the release process - border as emergency release point.

Acknowledgment

This work was partially founded by Italian MIUR PON projects Pico&Pro (ARS01 01061) AGREED (ARS01 00254), FURTHER (ARS01 01283), RAFAEL (ARS01 00305) and by Apulia Region (Italy) Research Projects E-SHELF (OSW3NO1) and INTENTO (36A49H6).

7 References

1 M. Wollschlaeger, T. Sauter, J. Jasperneite, The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0, IEEE Industrial Electronics Magazine 11 (1) (2017) 17–27.

2 O. Vermesan, P. Friess, Internet of things-from research and innovation to market deployment, Vol. 29, River Publishers Aalborg, 2014.

3 R. Murphy, Introduction to AI Robotics, MIT Press, Cambridge MA, 2000.

4 C. Chang, Y. Chin, C. Chen, C. Chang, Impasse-aware node placement mechanism for wireless sensor networks, IEEE Transactions on Systems, Man, and Cybernetics: Systems 48 (8) (2018) 1225–1237.

5 C. Houaidia, H. Idoudi, L. A. Saidane, Improving connectivity and coverage of wireless sensor networks using mobile robots, in: IEEE Symposium on Computers & Informatics (ISCI), IEEE, 2011, pp. 454–459.

6 E. Mesa-Barrameda, N. Santoro, W. Shi, N. Taleb, Sensor deployment by a robot in an unknown orthogonal region: Achieving full coverage, in: 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS), IEEE, 2014, pp. 951–960.

7 H. Idoudi, J. Bennaceur, Fault tolerant placement strategy for wsn, in: IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2016, pp. 1–5.

8 T.-Y. Lin, H. A. Santoso, K.-R. Wu, Global sensor deployment and local coverage-aware recovery schemes for smart environments, IEEE Transactions on Mobile Computing 14 (7) (2015) 1382–1396.

9 Z. Cheng, M. Perillo, W. B. Heinzelman, General network lifetime and cost models for evaluating sensor network deployment strategies, IEEE Transactions on mobile computing 7 (4) (2008) 484–497.

10 G. Tuna, V. C. Gungor, K. Gulez, An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters, Ad Hoc Networks 13 (2014) 54–68.

11 M. A. Batalin, G. S. Sukhatme, Coverage, exploration and deployment by a mobile robot and communication network, Telecommunication Systems 26 (2) (2004) 181–196.

12 P. M. Shankar, Fading and Shadowing in Wireless Systems, Springer New York, 2012.

13 H. M. Ammari, Connected k-coverage protocols for densely deployed wireless sensor networks, in: Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice, IGI Global, 2010, pp. 209–234.

14 C.-F. Huang, Y.-C. Tseng, The coverage problem in a wireless sensor network, Mobile networks and Applications 10 (4) (2005) 519–528.

15 M. M. Holland, R. G. Aures, W. B. Heinzelman, Experimental investigation of radio performance in wireless sensor networks, in: IEEE 2nd Workshop on Wireless Mesh Networks (WiMesh), IEEE, 2006, pp. 140–150.

16 S. R. Blackburn, T. Etzion, K. M. Martin, M. B. Paterson, Efficient key predistribution for grid-based wireless sensor networks, in: International conference on information theoretic security, Springer, 2008, pp. 54–69.

17 G. Zhou, T. He, S. Krishnamurthy, J. A. Stankovic, Impact of radio irregularity on wireless sensor networks, in: Proceedings of the 2nd international conference on Mobile systems, applications, and services, ACM, 2004, pp. 125–138.

18 STIIMA, Stiima-cnr: Stiima laboratory - bari, via amendola, `http://www.stiima.cnr.it/it/index.php?sez=9&ssez=6`, (Accessed on 2019-04-05.

19 V. Scilimati, A. Petitti, P. Boccadoro, R. Colella, D. Di Paola, A. Milella, L. Grieco, Industrial internet of things at work: a case study analysis in robotic-aided environmental monitoring, IET Wireless Sensor SystemsHttp://digital-library.theiet.org/content/journals/10.1049/iet-wss.2017.0032.

20 M. Tellez, S. El-Tawab, M. H. Heydari, Iot security attacks using reverse engineering methods on wsn applications, in: IEEE 3rd World Forum on Internet of Things (WF-IoT), 2016, pp. 182–187.

21 P. Boccadoro, G. Piro, D. Striccoli, L. A. Grieco, Experimental comparison of industrial internet of things protocol stacks in time slotted channel hopping scenarios, in: Proceedings of IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 2018.

22 H. Harb, A. Makhoul, Energy-efficient sensor data collection approach for industrial process monitoring, IEEE Transactions on Industrial Informatics 14 (2) (2018) 661–672.

23 Regents of the University of California, OpenWSN (April 2016).
URL `https://openwsn.atlassian.net/wiki/pages/viewpage.action?pageId=688187`

24 T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, K. Pister, Openwsn: A standards-based low-power wireless development environment, Wiley Transactions on Emerging Telecommunications Technologies 23 (5) (2012) 480–493.

25 T. B. Chandra, P. Verma, A. Dwivedi, Operating systems for internet of things: A comparative study (2016).
URL `https://arxiv.org/pdf/1504.02517`

26 D. Vasiljević, G. Gardašević, Performance evaluation of openwsn operating system on open mote platform for industrial iot applications, in: International Symposium on Industrial Electronics (INDEL), 2016, pp. 1–6.

27 T. Chang, P. Tuset-Peiro, X. Vilajosana, T. Watteyne, Openwsn amp; openmote: Demo'ing a complete ecosystem for the industrial internet of things, in: 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 2016, pp. 1–3.

28 D. Applegate, R. Bixby, V. Chvátal, W. Cook, The traveling salesman problem, ISBN: 0-691-12993-2.

29 K. Srinivasan, P. Levis, Rssi is under appreciated, in: Proceedings of the Third Workshop on Embedded Networked Sensors, Cambridge, MA, USA, Vol. 3031, 2006, p. 239242.

30 Z. Peng, X. Gui-bin, X. Ping-ping, Study on the communication link quality in wireless sensor network (2008) 66–69doi:10.1049/cp:20080938.

31 G. Micoli, P. Boccadoro, G. Valecce, A. Petitti, R. Colella, M. Annalisa, L. A. Grieco, ASAP: a decentralized slot reservation policy for dynamic 6TiSCH Networks in Industrial IoT, in: IEEE ICC 4th Convergent Internet of Things (C-IoT) Workshop, Shangai, Cina, 2019.