



Towards efficient deployment in Internet of Robotic Things

Cristanel Razafimandimby, Valeria Loscrì, Anna Maria Vegni

► To cite this version:

Cristanel Razafimandimby, Valeria Loscrì, Anna Maria Vegni. Towards efficient deployment in Internet of Robotic Things . Interoperability, Integration, and Interconnection of Internet of Things Systems (Extended Papers for Springer Series on IoT), 2017. hal-01499345

HAL Id: hal-01499345

<https://hal.science/hal-01499345>

Submitted on 12 Apr 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Towards efficient deployment in Internet of Robotic Things

Cristanel Razafimandimby, Valeria Loscri and Anna Maria Vegni

Abstract

Internet of Robotic Things (IoRT) is a new concept introduced for the first time by ABI Research. Unlike the Internet of Things (IoT), IoRT provides a dynamic actuation and is considered as the new evolution of IoT. This new concept will bring new opportunities and challenges, while providing new business ideas for IoT and robotics' entrepreneurs. In this work, we will focus particularly on two issues: (i) connectivity maintenance among multiple IoRT robots, and (ii) their collective coverage. We will propose (i) IoT-based, and (ii) a neural network control scheme to efficiently maintain the global connectivity among multiple mobile robots to a desired quality-of-service (QoS) level. The proposed approaches will try to find a trade-off between collective coverage and communication quality. The IoT-based approach is based on the computation of the algebraic connectivity and the use of virtual force algorithm. The neural network controller, in turn, is completely distributed and mimics perfectly the IoT-based approach. Results show that our approaches are efficient, in terms of convergence time, connectivity, and energy consumption.

Key words: IoT-based, connectivity maintenance, IoRT, neural network

Cristanel Razafimandimby

Inria Lille - Nord Europe, France, e-mail: jean.razafimandimby_anjalalaina@inria.fr

Valeria Loscri

Inria Lille - Nord Europe, France e-mail: valeria.loscri@inria.fr

Anna Maria Vegni

University of Roma Tre, Italy e-mail: annamaria.vegni@uniroma3.it

1 Introduction

Nowadays, Internet of Things (IoT) technology begins to take an important place in economic systems and in society daily life [10]. It has got a large success in several application areas, ranging from smart city applications [4], [20] to smart grid [21]. However, most of IoT applications are based only on static actuation. Later, adding an active role for actuators will be needed, in order to optimize the systems where they are present. Robotic systems match very well to this new need, since robots can sense and interact with their environment. Therefore, integrate robots as a device in IoT is obvious.

1.1 Overview of Internet of Robotic Things

Internet of Robotic Things (IoRT) is a new concept introduced for the first time by ABI Research [1]. It was defined as an intelligent set of devices that can monitor events, fuse sensor data from a variety of sources, use local and distributed intelligence to determine a best course of action, and then act to control or manipulate objects in the physical world [1]. This new concept is expected to be the evolution of IoT and robotics.

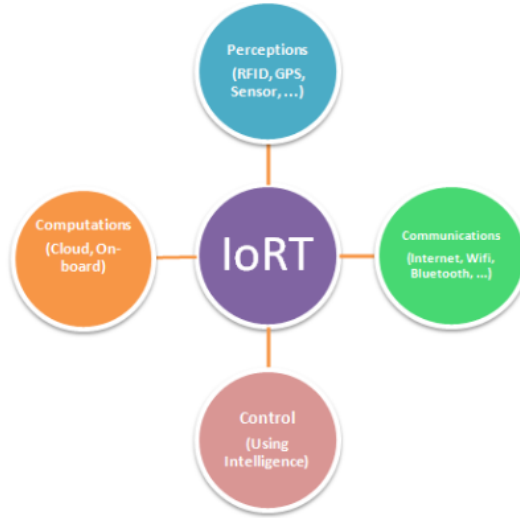


Fig. 1: Internet of Robotic Things

In most of the time, the brain of robots and control mechanisms are local ie on-board the robots themselves. However, in IoRT concept, the computation and the control can be assigned to the cloud. In this case, Internet allows the

IoRT robots to be connected to the Cloud. With advances in robot operating system framework, communication with the Internet is not complicated. It requires only a simple call of an application programming interface (API).

Assigning the processing to the Cloud increases the data-processing and interpretation capabilities. However, even if Cloud computing plays important role in the IoRT concept, local and distributed computations are still required for real-time applications or when the access to the Cloud is not feasible (e.g. after a disaster).

As mentioned before, local and distributed intelligence are also used to determine a best course of action. Hence, intelligence is also a vital part for the IoRT concept. The different kinds of intelligences used in this work will be detailed in Sections 3 and 4.

1.2 Issues and motivations

In various IoRT applications (e.g: smart agriculture, smart environment monitoring, smart exploration, smart disaster rescue, etc.), the use of mobile robots' teams brings many advantages over one powerful IoRT robot. As a matter of fact, a team of robots can accomplish tasks more efficiently, faster and more reliable than a single robot [7], [18], [5]. To carry out cooperative tasks, IoRT team members need to communicate with each other, often via a wireless link (i.e. Wifi, Bluetooth). Maintaining communication among multiple mobile IoRT robots is therefore a crucial issue.

Many approaches have been designed to maintain the connectivity of multi-robot and multi-agent systems. These approaches can be classified into two groups i.e. (i) local and (ii) global connectivity maintenance. With the local connectivity maintenance, the initial set of edges which define the graph connectivity must be always preserved over time. Unlike local connectivity maintenance, global connectivity maintenance allows suppression and creation of some edges, as long as the overall connectivity of the graph is conserved. Different examples and results related to these approaches can be found in [14], [23], [6], [24], [15], [8].

In Multi-Robot Systems, global connectivity maintenance is often used since the local connectivity maintenance presents some restrictions. Besides connectivity maintenance, the major problematic in global connectivity maintenance approaches is how to maximize the network connectivity. Maximizing the connectivity is important to ensure reliable communication between any pair of IoRT robots. Many works based on graph theory were proposed in the literature to face this problematic. These works are extensively used in multi-agent systems and are based on the maximization of the algebraic connectivity. In this work, we try to migrate some of these ideas in IoRT applications. Precisely, we will use the graph connectivity metric to maintain the global connectivity of IoRT robots' team, when they are in mobility.

This work addresses also the coverage issue. In general, coverage issue aims to determine how well the sensing field is monitored or tracked by sensors. In literature, Virtual Force Algorithm (VFA) was widely used to formulate this problem [27]. However, these methods have limitations since there are situations that do not allow the systems to converge in a stable state [9]. We will present a new solution to this problem later in this work.

To summarize, in this work we address two problems i.e. (i) connectivity maintenance, and (ii) collective coverage. However, it is interesting to mention that maximizing these two parameters simultaneously is difficult (if not impossible). Maximize the collective coverage may lead poor communication quality and conversely (i.e. a very good communication may lead poor coverage). It follows that our goal is to capture the trade-off between collective coverage and communication quality. Leveraging on the above motivations, in this work, we propose two motion control strategies which maintain global connectivity between IoRT robots to a desired QoS level. The first approach is an IoT-based while the second is a distributed trained neural network controller.

The main contribution of this work is the design of two approaches that have the following properties:

- The first approach is IoT-based, IoRT robots works with a central object which has high computation capability for network connectivity computing/monitoring and for the robot motion decision;
- Our both approaches converge to the desired communication quality level;
- Connectivity between any pair of IoRT robots is kept all along the deployment procedure;
- Our approaches use a distributed virtual force algorithm when the access to the central object is available and a distributed trained neural network controller otherwise. Both strategies are computed locally and based only on the local neighborhood information.

The rest of this chapter is organized as follows. Section 2 provides some backgrounds which include information on algebraic graph theory and neural networks. Section 3 describes the IoT-based approach, while Section 4 details the neural network approach. Section 5 provides the simulation results. Finally, Section 6 is dedicated to the conclusions.

2 Backgrounds

2.1 Graph representation and Eigenvalues

Multi-Robot Systems (MRS) can be represented by a graph $G(V, E)$ where V is the set of vertices representing each IoRT robot and $E \subseteq V^2$ is the set of edges. E can be defined as : $E = \{(i, j) \in V^2 \mid i \neq j \wedge d(i, j) \leq R\}$,

where $d(i, j)$ is the euclidean distance between i -th and j -th IoRT robots and R is the communication range. Following the above definition, let N_i be the one-hop neighborhood of the i -th IoRT robot. Thus, N_i is the set of IoRT robots which can exchange information with IoRT robot i . N_i can be defined as follow :

$$N_i = \{j \in V \mid d(i, j) \leq R\} \quad (1)$$

2.1.1 Definition 1

An undirected graph G is connected if there exists a path between each pair of vertices.

In this paper, the graph G may evolve over time due to the IoRT robots motion but has to be always connected.

2.1.2 Definition 2

Let define a matrix $A \in R^{n \times n}$. The scalar λ is an eigenvalue of A if there exists a non-zero vector w such that

$$A.w = \lambda.w \quad (2)$$

The vector w is called eigenvector of A corresponding to λ .

2.2 Laplacian matrix and algebraic connectivity

Given a undirected graph G , its Laplacian matrix L is defined as:

$$L(G) = \Psi(G) - A(G) \quad (3)$$

where:

- $A(G)$ is the weighted adjacency matrix of graph G whose entries A_{ij} is defined as in [14]:

$$A_{ij} = \begin{cases} 1 & d(i, j) < D_{th} \\ e^{\frac{-5(d(i, j) - D_{th})}{R - D_{th}}} & D_{th} \leq d(i, j) \leq R \\ 0 & d(i, j) > R \end{cases} \quad (4)$$

- $\Psi(G)$ is a diagonal matrix such as the components $\Psi_i = \sum_{j=1}^n A_{ij}$ along the diagonal
- D_{th} is the desired distance between each pair of IoRT robots.

The Laplacian matrix $L(G)$ holds some interesting properties:

1. Let $\mathbf{1}$ be the column vector of all ones. Then, $L\mathbf{1} = 0$.
 2. Let $\lambda_i, i = 1, \dots, n$ the eigenvalues of the Laplacian matrix $L(G)$.
- The eigenvalues of $L(G)$ can be ordered such that

$$0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \dots \leq \lambda_n \quad (5)$$

- $\lambda_2 > 0$ if and only if the graph G is connected. The second-smallest eigenvalue λ_2 is called also algebraic connectivity of the graph G . The value of λ_2 indicates how well connected the graph is.

Further details on graph theory and proof can be found in [11].

2.3 Artificial Neural Networks

Artificial Neural Networks (ANN) were inspired by the human brain and was designed as a computational model to solve specific problems [2]. Its architecture is defined by (i) a basic processing element called artificial neuron, and (ii) the way in which they are interconnected. The output value of a neuron is given by:

$$output = f\left(\sum_i w_i x_i + b\right) = f(W^T X + b) \quad (6)$$

where

- x_i : the inputs
- w_i : connections' weights between x_i and the neuron
- W : weights' vector
- X : inputs' vector
- b : the bias
- f : the activation function

The basic architecture of ANN contains three neuron layers: input layer, hidden layer and output layer. In this case, the outputs of one layer become the inputs of next layer [2]. A typical artificial neuron and a basic ANN are illustrated in Fig. 2.

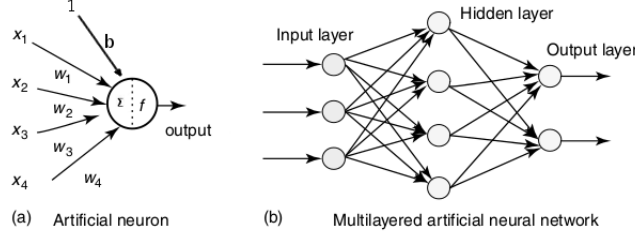


Fig. 2: Architecture of an artificial neuron and a multilayered neural network

A key element of an artificial neural network is its ability to learn. This meant that ANN has to learn from a data set in order to match the inputs to the desired output. During the learning process, weights and biases are adjusted till the desired output will be reached. There are several learning algorithm but in this work we will use the backpropagation algorithm [22].

2.4 Virtual Force Algorithm

Virtual Force Algorithm (VFA) is extensively used to solve the coverage problem on robots and sensors networks. The main idea is to model each robot or sensor as a particle in the potential field. The potential field exerts forces on the nodes nearby. The force may be either attractive or repulsive force according whether they are close or far to each other. If two nodes are placed closer than the desired distance D_{th} , repulsive forces are exerted on each other. Otherwise, attractive forces are exerted if two nodes are farther than D_{th} . The repulsive force aims to avoid a poor coverage while the attractive force ensures that a globally uniform node placement will be achieved [27]. For any pairwise of node i and j , the mutual force F_{ij} can be written as the negative gradient of the potential field. So, we can build a potential function V_{ij} such as :

$$F_{ij} = -\nabla V_{ij} \quad (7)$$

According to the traditional VFA, the force \vec{F}_{ij} is given as :

$$\vec{F}_{ij} = \begin{cases} (w_a(d(i,j) - D_{th}), \theta_{ij}), & \text{if } d(i,j) > D_{th} \\ 0, & \text{if } d(i,j) = D_{th} \\ (w_r(d(i,j) - D_{th}), \theta_{ij} + \pi), & \text{if } d(i,j) < D_{th} \end{cases} \quad (8)$$

where :

- w_a is the virtual force attractive coefficient
- w_r is the virtual force repulsive coefficient

- θ_{ij} is the orientation of the line segment from nodes
- D_{th} is the desired distance between each pair of nodes
- $d(i, j)$ is the euclidean distance between nodes i and j

This traditional VFA has limitations since there are situations that do not allow the systems to converge in a stable state [9]. We will present a new modified version of VFA in Section 3.

2.5 Ideal deployment for full coverage

The problem of maximizing the coverage of robots and sensors network was addressed in several works, using either virtual force algorithm [26], [27], [13], [17] or geometrical approaches [25], [16], [3] which manage the pairwise distance between any pair of node and locally arrange the network topology as a triangle tessellation. In this work, we focus only on the virtual force algorithm, since geometry-based approaches computation can only be done when the global location information of all the nodes in the network is known. An ideal deployment structure is show in Fig. 3. It is worth to mention that the ideal deployment will be achieved if the proposed approach converges to the desired distance D_{th} . There is no coverage hole in an ideal deployment.

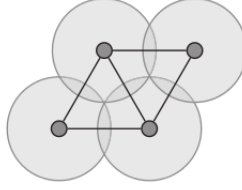


Fig. 3: Ideal deployment for full coverage

3 IoT-based approach

As we saw in the previous Section, an undirected graph is connected if and only if $\lambda_2 > 0$. Therefore, any strategy which maintains λ_2 at positive values guarantees global connectivity among multi-robot systems.

In this Section, we present an IoT-based approach which is capable of maintaining desired wireless communication coverage among neighboring robots.

The proposed approach uses a Central Object (*CO*) with high computation capability to compute and monitor the connectivity of the overall multi-robot system. We assume that each IoRT robot knows its own position by using GPS or other localisation system. Beacon messages are used to allow IoRT robots exchange their positions with their one-hop neighbors. Each IoRT robot in the system applies a modified version of VFA (as described on formula 9) to control its movement. This computation is only based on the local neighborhood information. In order to keep the desired distance and hence the desired connectivity quality with its neighbor, the i -th IoRT robot should move away from the IoRT robot $j \in N_i$ if $d(i, j) < D_{th}$ and should move close if $d(i, j) > D_{th}$. D_{th} is the desired distance between each pair of IoRT robots. This simple control law generates a vector position \vec{P}_{ij} such that the i -th IoRT robot keeps the line of sight of the IoRT robot j . \vec{P}_{ij} is defined as :

$$\vec{P}_{ij} = \begin{cases} (0.1 \times k \times \Delta d, \theta_{ij}) & \text{if } d(i, j) > D_{th} \text{ and } \Delta d > \epsilon \\ (k \times \Delta d, \theta_{ji}) & \text{if } d(i, j) < D_{th} \text{ and } \Delta d > \epsilon \end{cases} \quad (9)$$

where :

- $\Delta d = |d(i, j) - D_{th}|$
- θ_{ij} is the orientation of the line segment from robots i to j ;
- k is the damping coefficient
- ϵ is a lower bound of Δd . It will be used in order to avoid useless small movements.

In order to overcome the problem in the traditional VFA, we set the attractive coefficient w_a to one tenth of repulsive coefficient k ($w_a = 0.1 \times k$).

When the i -th IoRT robot has more than one neighbor, its new position is calculated as the summation of the position decisions with respect to all the neighbors :

$$\vec{P}_i = \sum_{j \in N_i} \vec{P}_{ij} \quad (10)$$

After calculating their new positions, each IoRT robot sends the computed position to the Central Object (*CO*). Then, *CO* computes the algebraic connectivity λ_2 of the IoRT robots network according to the formula 3. The central object *CO* allows each IoRT robot to move to their new positions if and only if $\lambda_2 > 0$. This guarantees that global connectivity is always kept all along the deployment procedure. It is important to note that IoRT robots and CO can communicate to each other through an IoT platform.

The following algorithm summarizes our approach:

Algorithm 1 IoT-based (runs every t units of time)

Phase I : Neighbor Discovery

MyNeighbor = FindNeighbor(RobotId)

Phase II : Compute the position \vec{P}_{ij} between two robots

Compute \vec{P}_{ij} using Formula 9

Phase III : Compute the new position \vec{P}_i

Compute \vec{P}_i using Formula 10

Phase IV : Compute algebraic connectivity

Compute λ_2 of the dynamic Laplacian matrix $L(G)$

Phase V : Deployment

if $\lambda_2 > 0$ **then**
 move to \vec{P}_i
else
 do not move

4 ANN-based approach

As we mentioned before, the connection to the central object is not always possible. For example, a rescue operation may be difficult after a disaster when the access to the central node is not available. An approach which easily adapts to any type of situation and environment is more than necessary.

To meet this need, we provide an ANN-based technique which can perfectly mimic the behaviors of IoT-based approach. The ANN-based approach is completely distributed and is trained from a set of data. The data set is obtained by using the IoT-based approach and we use backpropagation algorithm to train ANN. The trained ANN is constituted by 2 input units and 1 output unit. The 2 input units are $d(i, j)$ and θ_{ij} , while the output is \vec{P}_{ij} . Therefore, the trained ANN is executed locally for each IoRT robot to control its movement according to its neighbor's distance $d(i, j)$ and angle θ_{ij} . When the position \vec{P}_{ij} is estimated, the new position \vec{P}_i of the IoRT robot is computed by using the formula 10. Then, the collective movement of all IoRT robots will allow our trained ANN converge to the desired distance D_{th} . The global connectivity will also keep if our ANN is well trained (i.e. if training error equals zero or near to zero). The algorithm below illustrates our ANN-based approach:

Algorithm 2 ANN approach (runs every t units of time)**Phase I: Neighbor Discovery**

MyNeighbor = FindNeighbor(RobotId)

Phase II: Estimate the position \vec{P}_{ij} between two robots

$\vec{P}_{ij} = \text{trained_ann}(d(i, j), \theta_{ij})$

Phase III: Compute the new position \vec{P}_i

Compute \vec{P}_i using Formula 10

Phase IV: Deployment

move to \vec{P}_i

5 EVALUATION & DISCUSSION OF THE RESULTS

In this Section we first describe the simulation parameters and then provide the simulation results of our approaches. We are interested in studying how our approaches converge to the desired distance D_{th} between any pair of IoRT robot (hence to the desired communication quality matching D_{th}). We will see also how the density of IoRT robots influences the traveled distance of a robot. The importance of taking into account the algebraic connectivity before taking a movement decision will be also highlighted. Our approaches will be compared to the approach described in [17] called hereafter EVFA (Extended Virtual Force-Based Approach). EVFA was designed by its authors to overcome the connectivity maintenance and nodes stacking problems in the traditional Virtual Force Algorithm (VFA). Unlike our approaches, EVFA is based only on the orientation force and the judgment of distance force between node and its one-hop neighbors.

We assess our techniques w.r.t. (i) the algebraic connectivity, (ii) the robot traveled distance, (iii) the average distance, and (iv) the QoS level expressed in terms of RSSI (Received Signal Strength Indicator). Simulations have been carried out for a variable number of robots (i.e. from 5 to 50 robots) in an area of 3x3 km.

5.1 Simulation parameters

All the algorithms in this paper were implemented in version 2.29 of Network Simulator with patch from [12] that reflect a realistic channel propagation and error model. The patch is used in order to provide the effect of interference and different thermal noises to compute the signal to noise plus interference

ratio (SINR) and accounting for different bit error rate (BER) to SINR curves for the various codings employed [19]. Table 1 summarizes the parameters used in the simulations.

Physical	Propagation model	Two ray ground
	Error model	Real
	Antennas gain	Gt = Gr = 1
	Antennas height	ht = hr = 1 m
	Communication range	250 m
Statistics	Number of samples	100
	Simulation time	3000s
	Confidence Interval	95%
Mobility	Computation of the new position	see formula (8)
	Damping coefficient k	0.5
	D_{th}	212 m
ANN	Layer number	4
	Input number	2
	Output number	1
	Neuron's number in hidden layers	15
	Desired Error	0.00001
	Max epochs	10000
	Activation function	sigmoid symmetric
	Learning rate	0.2
	Training algorithm	backpropagation
Topology	Topology width	3000 m
	Topology height	3000 m

Table 1: Simulation parameters

5.2 Simulation results

All of the following results are the average of 100 times simulations and we assume that the topology is totally connected at the beginning of the simulation.

We can observe in the Figures [5-8](a) that IoT-based approach always kept the global connectivity since it always take account the algebraic connectivity constraint. Unlike IoT-based, EVFA has a connectivity problem when the robots density is low and this can explain why EVFA traveled a lot when the number of robot is less than 15 (see Fig. 4(a)). The goodness of connectivity is observed in EVFA when the robots density is higher (i.e. greater than 15 robots). However, as we mentioned before our goal is not to maximize the algebraic connectivity but just to keep it always greater than zero. This condition is enough to keep the global connectivity.

Figures [5-8](c) and [5-8](d) illustrate the convergence of our algorithms to the desired distance and the desired communication quality (RSSI) through-

out the simulation. We can notice that our approaches converge quickly to the aforementioned parameters which is not always the case for EVFA. We can see also that ANN mimics perfectly the behaviours of the IoT-based approach. This is due to the fact that our neural network has been well trained.

Figures [5-8](b) depict the distance traveled by a robot during the simulation time. By considering the relationship between energy and traveled distance, we can say that our approaches are energy efficient as compared to EVFA. However, it is observed that ANN consumes a bit more than IoT-based approach. This is because ANN has made a bit more step¹ to converge (see Fig. 4).

As a conclusion, it is worth to say that the global connectivity is reached with ANN approach since it was well trained and inherited the characteristics of IoT-based approach.

It should be noted that the coverage rate strongly depends on the initial network topology. If initially the robots are very close to each other, the convergence to the desired distance enlarges the collective coverage. Else, if the distances between robots are initially bigger than the desired distance, the coverage rate will decrease but the communication quality will improved. This has been proven but we omit it in this paper for lack of space.

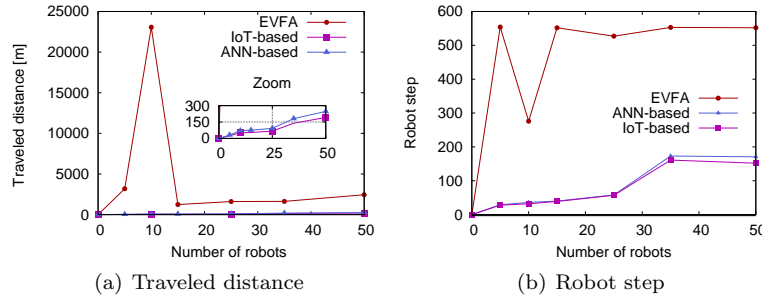


Fig. 4: Traveled distance and robot step according to the robots number

¹ Each robot increments its counter when it decides to move. In this paper, robot step is defined as the average of the maximum counter value achieved by one robot during the simulation.

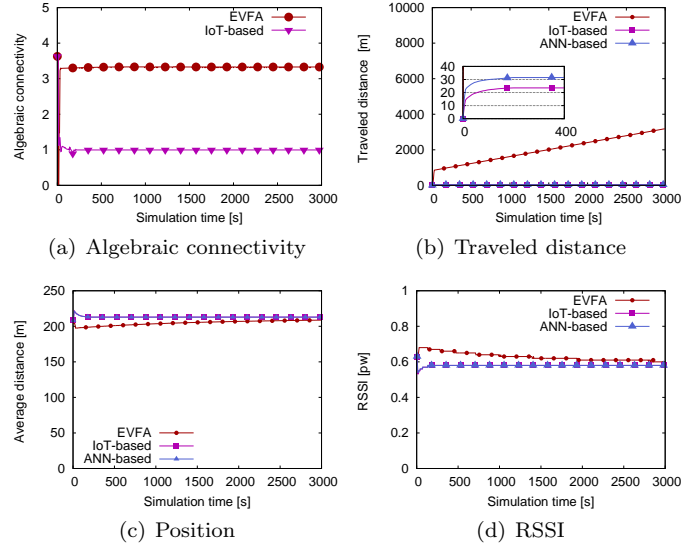


Fig. 5: Simulation results obtained with 5 robots moving in 3x3 km area

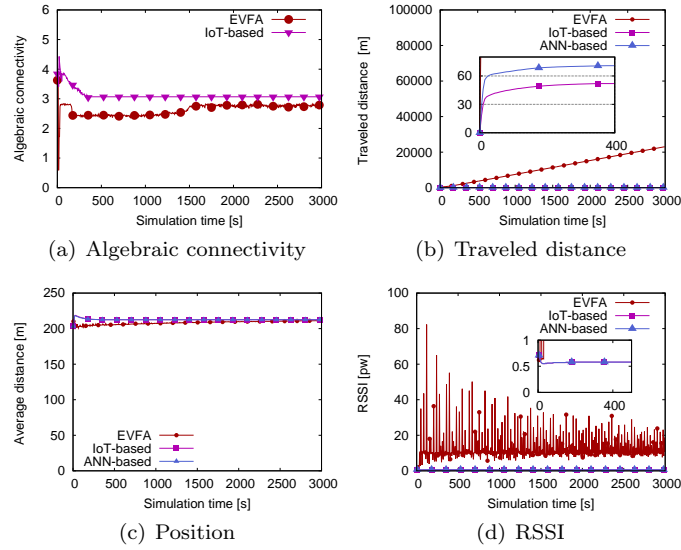


Fig. 6: Simulation results obtained with 10 robots moving in 3x3 km area

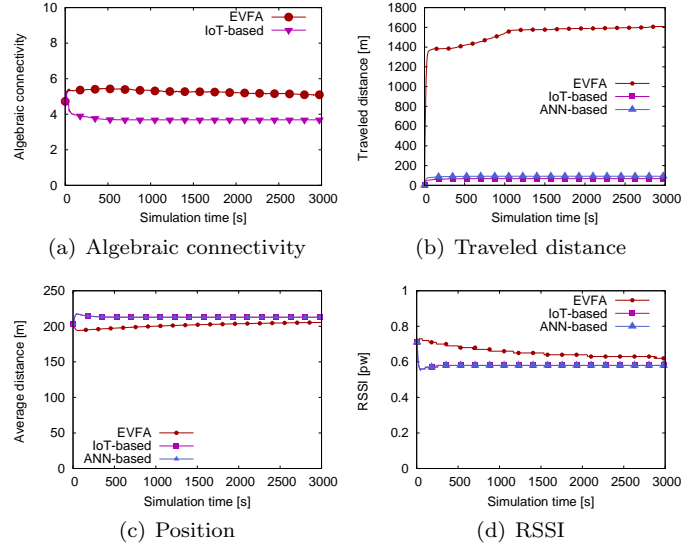


Fig. 7: Simulation results obtained with 25 robots moving in 3x3 km area

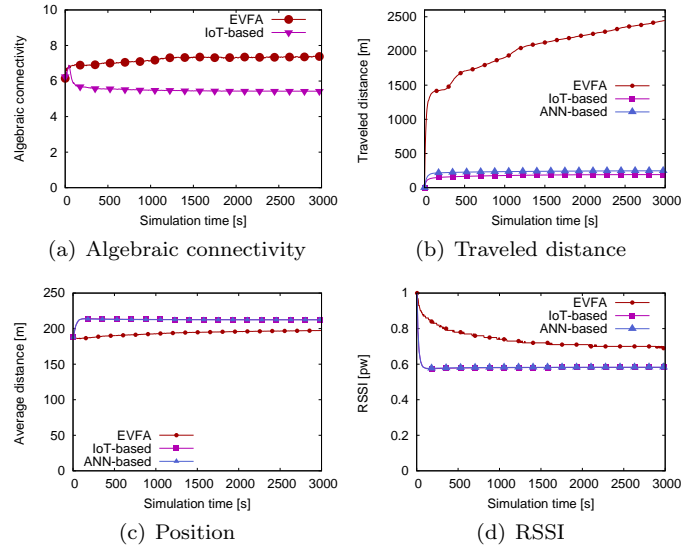


Fig. 8: Simulation results obtained with 50 robots moving in 3x3 km area

6 Conclusions

In this paper, we implemented a IoT-based and an ANN control scheme to maintain global connectivity among multiple IoRT robots. The proposed approaches tried to capture the trade-off between network coverage and communication quality expressed as RSSI level. The proposed algorithms allow the whole IoRT robot network converges to the desired distance, and hence the desired communication quality. Through extensive simulation we showed that our approaches outperform the EVFA approach proposed in [17], in terms of traveled distance and convergence time. Moreover, our proposed methods always maintain the global connectivity throughout the simulation.

Acknowledgment

This work was partially supported by a grant from CPER Nord-Pas-de-Calais/FEDER Campus Intelligence Ambiante.

References

- [1] ABI Research. Internet of robotic things. <https://www.abiresearch.com/market-research/product/1019712-the-internet-of-robotic-things>. Accessed November 2, 2015.
- [2] Abraham Ajith. Artificial neural networks. *handbook of measuring system design*, 2005.
- [3] Akkaya Kemal, Guneydas Ismail, and Bicak Ali. Autonomous actor positioning in wireless sensor and actor networks using stable-matching. *International Journal of Parallel, Emergent and Distributed Systems*, 25(6):439–464, 2010.
- [4] Alois Gianluca, Bedogni Luca, Di Felice Marco, Loscri Valeria, Molinaro Antonella, Natalizio Enrico, Pace Pasquale, Ruggeri Giuseppe, Trotta Angelo, and Zema Nicola Roberto. Stemnet: an evolutionary network architecture for smart and sustainable cities. *Transactions on Emerging Telecommunications Technologies*, 25(1):21–40, 2012.
- [5] Arai Tamio, Pagello Enrico, and Parker Lynne E. Editorial: Advances in multi-robot systems. *IEEE Transactions on robotics and automation*, 18(5):655–661, 2002.
- [6] Bouraqadi Noury, Stinckwich Serge, Moraru Victor, Doniec Arnaud, and others. Making networked robots connectivity-aware. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3502–3507. IEEE, 2009.

- [7] Cao Y Uny, Fukunaga Alex S, and Kahng Andrew. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.
- [8] Casteigts Arnaud, Albert Jérémie, Chaumette Serge, Nayak Amiya, and Stojmenovic Ivan. Biconnecting a network of mobile robots using virtual angular forces. *Computer Communications*, 35(9):1038–1046, 2012.
- [9] Chen Jiming, Li Shijian, and Sun Youxian. Novel deployment schemes for mobile sensor networks. *Sensors*, 7(11):2907–2919, 2007.
- [10] Council . Internet of things council. <http://www.theinternetofthings.eu>. Accessed November 18, 2015.
- [11] Fiedler Miroslav. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.
- [12] Fiore Marco. Ns-2.29 wireless update patch. <http://perso.citi.insa-lyon.fr/mfiore/research.html>. Accessed November 19, 2015.
- [13] Garetto Michele, Gribaudo Marco, Chiasserini Carla-Fabiana, and Leonardi Emilio. A distributed sensor relocation scheme for environmental control. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–10. IEEE, 2007.
- [14] Gennaro Maria Carmela De and Jadbabaie Ali. Decentralized control of connectivity for multi-agent systems. In *Decision and Control, 2006 45th IEEE Conference on*, pages 3628–3633. Citeseer, 2006.
- [15] Hsieh M Ani, Cowley Anthony, Kumar R Vijay, and Taylor Camillo J. Maintaining network connectivity and performance in robot teams. 2008.
- [16] Lee Geunho and Chong Nak Young. A geometric approach to deploying robot swarms. *Annals of Mathematics and Artificial Intelligence*, 52(2-4):257–280, 2008.
- [17] Li Jun, Zhang Baihai, Cui Lingguo, and Chai Senchun. An extended virtual force-based approach to distributed self-deployment in mobile sensor networks. *International Journal of Distributed Sensor Networks*, 2012, 2012.
- [18] Parker Lynne E. Multiple mobile robot systems. In *Springer Handbook of Robotics*, pages 921–941. Springer, 2008.
- [19] Pavon Javier Del Prado and Choi Sunghyun. Link adaptation strategy for ieee 802.11 wlan via received signal strength measurement. 2:1108–1113, 2003.
- [20] Petrolo Riccardo, Loscri Valeria, and Mitton Nathalie. Towards a smart city based on cloud of ththings. In *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities, WiMobCity*, pages 61–66. ACM, 2014.
- [21] Qinghai Ou, Yan Zhen, Xianghen Li, Yiying Zhang, and Linggang Zeng. Application of internet of things in smart grid power transmission. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1293–1303. IEEE, 2003.

- [22] Riedmiller Martin. Advanced supervised learning in multi-layer perceptrons from backpropagation to adaptive learning algorithms. *Computer Standards & Interfaces*, 16(3):265–278, 1994.
- [23] Sabattini Lorenzo, Secchi Cristian, Chopra Nikhil, and Gasparri Andrea. Distributed control of multirobot systems with global connectivity maintenance. *Robotics, IEEE Transactions on*, 29(5):1326–1332, 2013.
- [24] Stump Ethan, Jadbabaie Ali, and Kumar Vijay. Connectivity management in mobile robot teams. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1525–1530. IEEE, 2008.
- [25] Wang Guiling, Cao Guohong, and La Porta Thomas F. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.
- [26] Yoon Seokhoon, Soysal Onur, Demirbas Murat, and Qiao Chunming. Coordinated locomotion and monitoring using autonomous mobile sensor nodes. *IEEE Transactions on Parallel and Distributed Systems*, 22(10):1742–1756, 2011.
- [27] Zou Yao and Chakrabarty Krishnendu. Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1293–1303. IEEE, 2003.