UNIVERSITÉ LILLE 1 - SCIENCES ET TECHNOLOGIES

ÉCOLE DOCTORALE ED RÉGIONALE SPI 72

INRIA LILLE - NORD EUROPE

Toward Internet of Heterogeneous Things : Wireless communication maintenance and efficient data sharing among devices

Author : Anjalalaina Jean Cristanel Razafimandimby

Présentée et soutenue publiquement le 18/10/2017

pour obtenir le titre de ***Docteur***

de l'Université Lille 1 - Sciences et Technologies

specialité Informatique et applications

Numéro d'ordre : 42443

Composition du jury :

Rapporteurs :

Prof. Dr. Francesca Guerriero
University of Calabria, Italy

Prof. Dr. Nadjib AITSAADI
Engineering School ESIEE Paris / University Paris Est

Examinateurs :

Prof. Dr. Thomas Noël
University of Strasbourg

Dr. Razvan Stanica
INSA of Lyon

Directeur de thèse :

Dr. Nathalie Mitton
Inria, France

Co-encadrant de thèse :

Dr. Valeria Loscrí
Inria, France

"Tsy misy mafy tsy laitra ny zoto [Mg]
Avec du courage on vient à bout de tout [Fr]
With courage we can do everything [En]"

Malagasy proverb

Remerciements

Je tiens tout d'abord à remercier Dr. Nathalie Mitton, responsable de l'équipe FUN Inria Lille, non seulement de m'avoir accepté au sein de son équipe mais aussi pour les conseils qu'elle m'a donné tout au long de ma thèse.

Je tiens à exprimer aussi ma profonde gratitude et mes sincères remerciements à mon encadreur Dr. Valeria Loscrí, pour tout le temps qu'elle m'a consacré, pour ses directives précieuses et surtout pour la qualité de son suivi tout au long de ma thèse.

Je tiens à remercier particulièrement Dr. Tahiry Razafindralambo, pour l'attention qu'il m'a apporté durant mon stage et durant ma thèse. Merci pour les discussions intéressantes et les conseils.

Je remercie également les membres de mon jury d'avoir accepté de juger ce travail. Merci pour vos critiques constructives.

Je tiens à remercier aussi mes co-autheurs: Dr. Anna Maria Vegni et Prof. Alessandro Neri. C'été un grand plaisir et un honneur de travailler avec vous.

Mes remerciements vont aussi au personnel administratif d'Inria Lille et l'ensemble de l'équipe FUN pour leur accueil, leur bonne humeur et leur disponibilité pendant la durée de ma thèse. Un grand merci à Anne Rejl pour sa disponibilité et sa professionnalisme.

Je remercie aussi mes parents ainsi que mes frères et ma sœur, pour leur soutien incommensurable.

Enfin, j'aimerais remercier ma femme pour sa patience, ses encouragements et l'ensemble de ses conseils. «Misaotra betsaka» de m'avoir soutenu dans les périodes de doute.

Abstract

Internet of Things (IoT) technology begins to take an important place in our daily life and has already got a large success in several applications. Despite this success, most of IoT applications are based only on static actuation. However, adding an active role for actuators will be needed, in order to optimize the systems where they are present. To achieve this goal, in this thesis, we introduce a new concept called Internet of Heterogeneous Things (IoHT) which takes into account both static and dynamic actuation. The dynamic actuation is provided by a mobile robot or a mobile sensor. An IoHT device can be therefore static or equipped with motion capability. When a device has the capacity to move, we exploit the potential of controlled mobility by proposing efficient algorithms to maintain the global connectivity among IoHT devices. We show by simulation the efficiency of the proposed algorithms and their performance in terms of convergence time, connectivity, and traveled distance.

Once the connectivity among devices is guaranteed, another major challenge that should be solved is the huge amount of data they generate and transmit. To tackle this problem, we propose a Bayesian Inference Approach (BIA) which allows avoiding the transmission of high correlated data. Belief Propagation algorithm, coupled with the Markov Random Field model, is used in this case to reconstruct the missing sensing data. According to different scenarios, our approach is evaluated based on the real data collected from sensors deployed on indoor and outdoor environments. The results show that our proposed approach reduces drastically the number of transmitted data and the energy consumption, while maintaining an acceptable level of inference error and information quality.

Résumé

De nos jours, l'Internet des Objets (IdO) commence à prendre une place importante dans notre vie quotidienne. Il a obtenu un grand succès dans divers domaines d'application. Toutefois, malgré ce succès, la plupart des applications de l'IdO sont basées uniquement sur l'actionnement statique. Cependant, l'ajout d'un rôle actif pour les actionneurs sera nécessaire afin d'optimiser les systèmes où ils sont présents. Pour ce faire, dans cette thèse, nous introduisons un nouveau concept appelé Internet des Objets Hétérogènes (IdOH) qui prend en compte les actionnements statique et dynamique. L'actionnement dynamique est fourni par un robot mobile ou un capteur mobile. Les dispositifs constituant l'IdoH peuvent donc être statiques ou ont la capacité de se mouvoir. Dans ce dernier cas, nous exploitons le potentiel de la mobilité contrôlée en proposant des algorithmes efficaces pour maintenir la connectivité entre les dispositifs. Nous montrons par simulation l'efficacité des algorithmes proposés et leur performance en termes de temps de convergence, de connectivité et de distance parcourue.

Une fois que la connectivité entre les dispositifs est garantie, un autre défi majeur qui devrait être résolu est l'énorme quantité de données qu'ils génèrent et transmettent. Pour faire face à ce problème, nous proposons une approche d'inférence bayésienne qui permet d'éviter la transmission des données fortement corrélées. L'algorithme de propagation de croyance, couplé au modèle de champ aléatoire de Markov, est utilisé dans ce cas pour inférer les données manquantes. Selon différents scénarios, notre approche est évaluée sur la base des données réelles recueillies à partir des capteurs déployés sur des environnements intérieurs et extérieurs. Les résultats montrent que notre approche réduit considérablement la quantité de données transmises et la consommation d'énergie, tout en maintenant un niveau acceptable d'erreur d'inférence et de qualité de l'information.

Contents

Remerciements iv

Abstract vi

Résumé viii

1 Introduction 1
- 1.1 Towards Internet of Heterogeneous Things . . . 1
- 1.2 Issues and contributions . . . 2
- 1.3 Structure of the thesis . . . 3

2 State of the art 6
- 2.1 Connectivity maintenance and coverage problems . . . 6
  - 2.1.1 Spanning Tree . . . 7
  - 2.1.2 Laplacian matrix and algebraic connectivity . . . 7
  - 2.1.3 Coverage issues . . . 9
  - 2.1.4 Deployment for an area coverage problem . . . 9
  - 2.1.5 Virtual Force Algorithm . . . 10
- 2.2 Data prediction models . . . 11
  - 2.2.1 Time series models . . . 11
    - 2.2.1.1 Average prediction model . . . 11
    - 2.2.1.2 Constant prediction model . . . 11
    - 2.2.1.3 Linear prediction model . . . 12
    - 2.2.1.4 Exponential smoothing model . . . 12
    - 2.2.1.5 Autoregressive model . . . 12
    - 2.2.1.6 Moving average model . . . 12
    - 2.2.1.7 ARIMA model . . . 13
  - 2.2.2 Cross-sectional models . . . 13
    - 2.2.2.1 Simple regression . . . 13
    - 2.2.2.2 Multiple regression . . . 14
    - 2.2.2.3 Artificial Neural Networks . . . 14
    - 2.2.2.4 Graphical models . . . 15
    - 2.2.2.5 Inference . . . 19
  - 2.2.3 Comparative study of prediction models . . . 21
- 2.3 Summary . . . 22

3 Global Connectivity Maintenance Among IoHT Mobile Robots 24
- 3.1 Proposed solutions . . . 24
  - 3.1.1 IoT-based approach for global connectivity maintenance . . . 24
  - 3.1.2 ANN-based controller for global connectivity maintenance . . . 26
- 3.2 Simulation results . . . 26
- 3.3 Summary . . . 33

4 Neuro-Dominating Set Scheme For a Fast and Efficient Robot Deployment in IoHT **35**
4.1 Motivation and Background . . . 35
4.1.1 Dominating Sets in Graphs . . . 36
4.2 Neuro-Dominating Set Scheme . . . 36
4.2.1 NDS algorithm . . . 38
4.3 Simulation results . . . 39
4.4 Summary . . . 52

5 Bayesian Inference Approach For an Efficient Data Sharing among IoHT devices **54**
5.1 Network model . . . 54
5.2 Bayesian Inference Approach . . . 55
5.3 Experimental results . . . 56
5.3.1 Indoor environments . . . 56
5.3.2 Outdoor environments . . . 59
5.3.3 Raw data filtering in the sensing nodes . . . 61
5.3.4 Experimentation on FIT IoT-LAB platform . . . 63
5.4 Summary . . . 66

6 Conclusion And Perspective **68**
6.1 Summary of contribution . . . 68
6.1.1 Global connectivity maintenance among IoHT mobile devices . . . 68
6.1.2 Data reduction . . . 68
6.2 Perspectives . . . 69
6.2.1 Global connectivity maintenance . . . 69
6.2.2 Bayesian Inference Approach . . . 69

Bibliography **73**

List of Figures

1.1 Types of IoT devices [46] 1

2.1 Adjacency matrix element A_{ij} as a function of the distance between mobile robots i and j [23]. 8
2.2 Ideal deployment for an area coverage 10
2.3 Architecture of an Artificial Neural Network 15
2.4 An example of a Bayesian network 16
2.5 An example of a Markov random field 17
2.6 An example of MRF model with observed and hidden nodes. 18
2.7 The factor graph representation of Figure 2.5 19
2.8 Message passing algorithm used in BP [60] 20

3.1 Traveled distance and robot step according to the robots number 29
3.2 Simulation results obtained with 5 robots moving in 3x3 km area 29
3.3 Simulation results obtained with 10 robots moving in 3x3 km area 30
3.4 Simulation results obtained with 25 robots moving in 3x3 km area 30
3.5 Simulation results obtained with 50 robots moving in 3x3 km area 31
3.6 Simulation results obtained with 25 IoHTs devices using heterogeneous scenarios 32

4.1 Dominating Set 36
4.2 Traveled distance and Convergence time according the values of α 40
4.3 Variation of robot position according the values of α. Scenario with 15 robots. 40
4.4 Traveled distance according the number of the robots, for different approaches 41
4.5 Average number of neighbors for different number of robots 42
4.6 Simulation results obtained with 5 robots 44
4.7 Simulation results obtained with 10 robots 45
4.8 Simulation results obtained with 15 robots 46
4.9 Simulation results obtained with 25 robots 47
4.10 Simulation results obtained with 35 robots 48
4.11 Simulation results obtained with 50 robots 49
4.12 Simulation results obtained with 60 robots 50
4.13 Simulation results obtained with 70 robots 51

5.1 A cloud-based IoHT network model 55
5.2 Wireless sensors installed in the Intel Berkeley Research Lab [38] 56

5.3 Correlation between collected data during the first three hours of readings. 57
5.4 The PEACH network [11]. 59
5.5 Relationship between humidity and temperature data. . . . 60
5.6 Relationship between humidity and temperature data during the six days of readings. 62
5.7 Prototype of a sensor node and IoHT gateway 63
5.8 Variation of (a) the transmitted data, (b) the estimation error and (c) MSE according the value of the threshold $|e|_{Max}$. . . 65

6.1 Correlation between the nodes that are close (nodes in the red circle in (a)) using data from Intel Berkeley Lab. 70

List of Tables

2.1 Comparative study of prediction models 22

3.1 Simulation parameters 28

5.1 Results obtained during the first three hours of readings for different scenarios. 59
5.2 Results obtained during eighteen hours of readings for different scenarios. 60
5.3 Results obtained during six days of readings for different scenarios and using the created prototype. 62
5.4 Results obtained during the two days and half of readings. . 64

List of Abbreviations

ANN	Artificial Neural Network
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
BER	Bit Error Rate
BIA	Bayesian Inference Approach
BN	Bayesian Network
BP	Belief Propagation
CDS	Connected Dominating Set
DS	Dominating Set
EC	Energy Consumption
ER	Estimation error
EVFA	Extended Virtual Force-Based Approach
FG	Factor Graph
GM	Graphical Models
GPS	Global Positioning System
IoHT	Internet of Heterogeneous Things
IoRT	Internet of Robotic Things
IoT	Internet of Things
MA	Moving Average
MRF	Markov Random Fields
MRS	Multi Robot Systems
MSE	Mean Squared Error
NDS	Neuro Dominating Set
QoS	Quality of Service
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
SINR	Signal to Interference plus Noise Ratio
ST	Spanning Tree
VFA	Virtual Force Algorithm
WSN	Wireless Sensor Networks

Dedicated to all my supporters

Chapter 1

Introduction

1.1 Towards Internet of Heterogeneous Things

Nowadays, Internet of Things (IoT) technology begins to take an important place in economic systems and in society daily life [17]. It has got a large success in several application areas, ranging from smart city applications [5], [42] to smart grid [45]. With the progress of current technologies, the IoT is composed of heterogeneous devices such as sensors, RFID tags, arduino, Raspberry Pi, mobile phones, etc. IoT allows therefore a creation of new applications/services which require all these devices to communicate, to interact, to share data and processes. According Cisco's prediction, the number of internet-connected devices would exceed 50 billion by 2020 [16].

IoT devices can be classified into three types according their capacities [46]: (*i*) thin devices, (*ii*) intelligent devices, and (*iii*) actuated devices (see Figure 1.1).

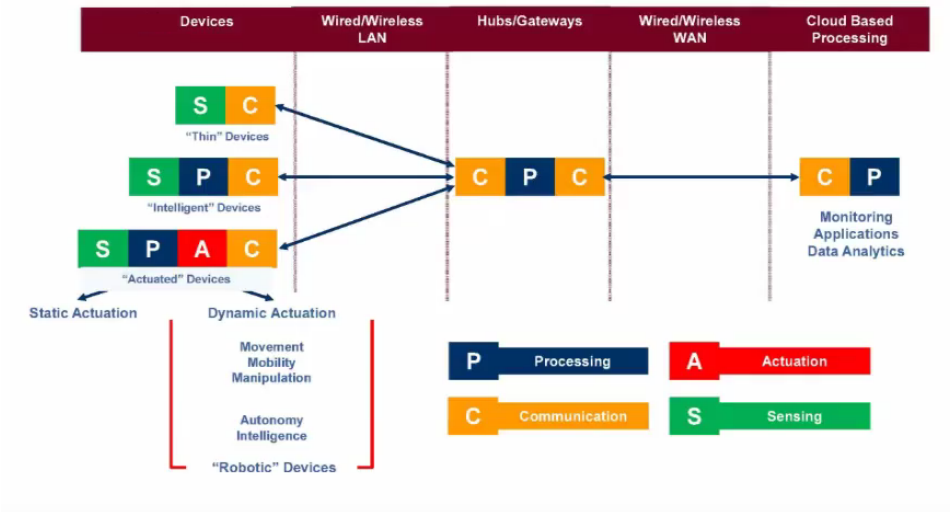


FIGURE 1.1: Types of IoT devices [46]

A- Thin devices

Thin devices are the most constraining of IoT devices. They can only sense and communicate data. All data processing is done on the remote servers. Infra-Red temperature and humidity sensors are a concrete examples of such devices.

B- Intelligent devices

Intelligent devices, in addition to their sensing and communication capabilities, provide a certain level of intelligence for data processing. They

therefore have a certain level of autonomy but most processing still occurs on remote servers. An example of intelligent device could be a smart metering device such as EMU-2 [7].

C- Actuated devices

Actuated devices are intelligent devices which can also act on the physical world. Two types of actuation can be defined: static and dynamic.

In static actuation, actuators perform a simple control according to the feedback they get. A very simple example is the intelligent thermostat which can decrease or increase the home temperature depending on whether it is too hot or too cold. However, static actuated devices do not have the capacity for manipulation, mobility, autonomy and movement.

Dynamic actuated devices, in turn, provide much more advanced abilities including movement, mobility, manipulation, and autonomy. Willow Garages PR2 robot [21] and Wifibot robot [48] are good examples of such devices.

Robots can be considered as dynamic actuators since they meet all the capacities cited above i.e. they are autonomous and have the ability to manipulate objects and to move around. Hence, their integration in the IoT paradigm is obvious and can extend the abilities of the traditional IoT infrastructure. To point out this synergistic nature between IoT and robotics, ABI Research introduced a new concept called Internet of Robotic Things (IoRT) [46]. IoRT 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. In some cases, they can physically move in the physical world [46].

In this thesis, the fusion of IoRT (hence IoT) with the data sharing scheme is called ***Internet of Heterogeneous Things*** (IoHT). An IoHT device can be therefore static (e.g. static sensors, arduino, Raspberry pi, etc) or equipped with motion capability (e.g. mobile sensors and robots).

1.2 Issues and contributions

IoHT is composed of tens to billions heterogeneous devices which interact with each other and with our environment. In general, these devices have limited resources in terms of CPU, memory, storage and power resources. Providing IoHT with energy saving approaches is thus a key issue.

Based on the proposed IoHT concept and the above-mentioned constraints, in this thesis we address the following questions:

1. **How to efficiently maintain the communication coverage among IoHT mobile sensors or robots?**
   As stated before, in this thesis, we focus on IoTH's context which consists of multiple mobile and static devices cooperating for tasks execution. Such IoHT system has several applications, e.g. surveillance, environment monitoring, disaster rescue, etc. The performance of an IoHT application relies on the efficient coordination among devices, which in turn strongly depends on reliable communication to support efficient data sharing among devices. The existence of reliable communication among devices is therefore a prerequisite for IoHT applications. Thus, maintaining the communication among IoHT devices

is really a crucial problem, especially if the devices have the motion capability (eg. mobile robot and mobile sensor). To deal with this challenge, in this dissertation, we exploit the potential of controlled mobility and address the connectivity maintenance problem in the context of deploying a group of mobile robots (or mobile sensors) into a unknown environment to form a desired communication coverage for supporting efficient data sharing. Mobility of some devices, such as mobile robots, will be therefore considered as a feature to be exploited instead of a challenge to be faced [37].

2. **How to efficiently share data from various heterogeneous sources?** One major challenge that should be addressed in the IoHT concept is the huge amount of data generated by the sensing devices, which make the control of sending useless data very important. The transmission of these huge amount of data to the network may affect the energy consumption of sensing devices, and can also cause network congestion issues. A strong reduction of massive amount of data generated by these sensing devices is therefore necessary. To do so, many approaches on data aggregation and compression have been proposed in the literature. However, these approaches do not take into account the most distinctive aspects of the IoHT i.e. the devices have limited resources and the data collected and transmitted by the sensor devices can be predicted using a simple inference algorithm. Using such an algorithm, one can exploit the power of double prediction. In this case, the sensor devices transmit their measurements only in cases where predictions fail i.e. when the prediction error exceeds a predefined threshold. Therefore, unlike data compression and aggregation, inference approaches can considerably eliminate the need to communicate the data. We go in this direction in this thesis by proposing an efficient Bayesian Inference Approach (BIA) for data sharing among IoHT devices. We take into account the heterogeneous aspect of the data and devices.

1.3 Structure of the thesis

The thesis is organized as follows:

1. Chapter 2 provides the state of art about the connectivity maintenance, coverage issues and data prediction models.

2. Chapter 3 presents the global connectivity maintenance among IoHT mobile robots. Two proposed approaches will be detailed : an hybrid approach based on the IoT concept and a distributed approach based on neural network.

3. Chapter 4 focuses on the Neuro-Dominating Set scheme for the mobile robot motion control. We adopt different roles among mobiles robots to enable them better achievement of global system performance.

4. Chapter 5 is dedicated to the data reduction solution for the IoHT concept. We present the Bayesian Inference Approach based on Belief Propagation algorithm for an efficient data sharing in indoor and outdoor scenarios.

5. Chapter 6 Concludes the thesis and draws up the perspectives in order to improve the proposed solutions.

Chapter 2

State of the art

2.1 Connectivity maintenance and coverage problems

As it is already stated, in this thesis, we address the connectivity maintenance problem in the context of deploying a team of IoHT mobile robots into a unknown environment to form a desired communication coverage for supporting an efficient data sharing.

Indeed, in various IoHT applications such as smart agriculture, smart environment monitoring, smart exploration and smart disaster rescue, the use of mobile robots' teams brings many advantages over one powerful robot. As a matter of fact, a team of robots (or a team of mobile sensors) can accomplish tasks more efficiently, faster and more reliably than a single robot [13], [39], [6]. In general, IoHT team members have limited capacity and need to communicate with each other, often via a wireless link (i.e. Wifi, Bluetooth), to carry out cooperative tasks. Maintaining connectivity among IoHT devices is therefore a crucial issue.

A lot of connectivity maintenance approaches have already been proposed in the literature [23], [49], [10], [3], [62], [54], [26], [50], [14]. These approaches can be classified into two groups i.e., (*i*) local and (*ii*) global connectivity maintenance. Local connectivity maintenance consists of preserving the initial set of edges which define the graph connectivity throughout the deployment 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.

In this thesis, we focus only on global connectivity maintenance since the preservation of each local link communication in the network is a very restrictive requirement which significantly limits the capability of the systems itself [49].

As a reminder, our mobile robot team can be represented by a graph $G(V, E)$ where V is the set of vertices representing each mobile robot and $E \subseteq V^2$ is the set of edges. E can be defined as :

$$E = \big\{(i,j) \in V^2 \mid i \neq j \ \wedge \ d(i,j) \ \leq R\big\} \tag{2.1}$$

where $d(i,j)$ is the euclidean distance between i-th and j-th mobile robots and R is the communication range.

Following the above representation, let N_i be the one-hop neighborhood of the i–th mobile robot. Thus, N_i is the set of robots which can exchange information with robot i. N_i can be defined as follow :

$$N_i = \{j \in V \mid d(i,j) \ \leq R\} \tag{2.2}$$

Definition 1 *An undirected graph $G(V, E)$ is connected if there exists a path between each pair of vertices $(i, j) \in V^2$.*

According to this definition, the most trivial way to determine the connectivity of a graph is to determine whether, for each node in the graph, there is a communication path between it and all the other nodes. Obviously, this naive solution is very costly in term of complexity. However, we have properties based on the Spanning Tree (ST) and Laplacian graph that give most effective solutions for the determination of connectivity [32]. It is therefore more advantageous to work with ST or Laplacian graph for any connectivity issue.

2.1.1 Spanning Tree

A Spanning Tree (ST) of an undirected graph $G(V, E)$ is a subgraph $G^*(V, E^*)$ such as there is no cycle in the subgraph G^* and the number of edges $|E^*| = n - 1$, where n is the mobile robot number and $E^* \subseteq E$.

Definition 2 *An undirected graph $G(V, E)$ is connected if and only if there exist a ST deduced from $G(V, E)$.*

So, we can always find a Spanning Tree in a connected graph and there are many works in literature that exploit this property for connectivity maintenance.

One disadvantage of ST based approaches is that the hierarchical relationship in the graph should always be preserved during the deployment. It is therefore not suited to global connectivity maintenance which we want to focus in this thesis. Furthermore, ST do not give us more information except the connectivness of the graph. We cannot know, for example, how well connected the graph is. For these reasons, it is better to work with the Laplacian graph and one of its eigenvalues.

2.1.2 Laplacian matrix and algebraic connectivity

Given an undirected graph $G(V, E)$, its Laplacian matrix L is defined as:

$$L(G) = \Psi(G) - A(G) \tag{2.3}$$

where:

- $A(G)$ is the weighted adjacency matrix of the graph $G(V, E)$ whose entries A_{ij} is defined as in [23]:

$$A_{ij=} \begin{cases} 1 & d(i,j) < D_{th} \\ e^{\frac{-5(d(i,j)-D_{th})}{R-D_{th}}} & D_{th} \leq d(i,j) \leq R \\ 0 & d(i,j) > R \end{cases} \tag{2.4}$$

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

The entry A_{ij} represents the strength of the connection among a pair of robots (i, j), which decays exponentially with the distance. Figure 2.1 illustrates the shape of A_{ij}.

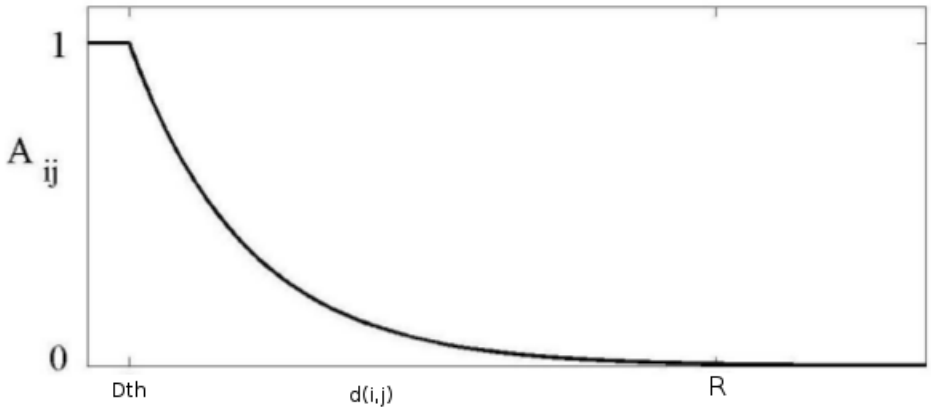


FIGURE 2.1: Adjacency matrix element A_{ij} as a function of the distance between mobile robots i and j [23].

Definition 3 *Let $L(G)$ be a $n \times n$ matrix. A scalar λ is an eigenvalue of $L(G)$ if there exists a non-zero vector w such that $L.w = \lambda.w$.*

Equation (2.5) has to be solved to find the eigenvalues of the laplacian matrix $L(G)$.

$$det_n(L) = \begin{bmatrix} l_{1,1} - \lambda & l_{1,2} & \dots & l_{1,n} \\ l_{2,1} & l_{2,2} - \lambda & \cdots & l_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ l_{n,1} & l_{n,2} & \dots & l_{n,n} - \lambda \end{bmatrix} = 0 \tag{2.5}$$

$det_n(L)$ is the determinant of the Laplacian matrix $L_{n\times n}$.
$L_{n\times n}$ has therefore n associated eigenvalues since $det_n(L)$ possesses exactly n solutions.

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

- Let **1** be the column vector of all ones. Then, L**1**= 0.
- Let λ_i, $i = 1, ..., n$ the eigenvalues of the Laplacian matrix $L(G)$.
- The eigenvalues of $L(G)$ can be ordered such that

$$0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n \tag{2.6}$$

- $\lambda_2 > 0$ if and only if the graph $G(V, E)$ **is connected**. So, any approach that maintains the value of λ_2 positive ensures also the graph connectivity. The second-smallest eigenvalue λ_2 is called also ***algebraic connectivity*** of the graph $G(V, E)$. The value of λ_2 indicates how weel connected the graph is.

2.1.3 Coverage issues

Besides the connectivity maintenance, most of deployment approaches mainly focus on robot's motion coordination to accomplish sensor coverage. Coverage issue aims to determine how well the sensing field is monitored or tracked by sensors.

Definition 4 *An area A is said to be covered by a sensor S_i if each location in A is within S_i's sensing range. A location in A is said to be k-covered if it is within at least k sensors' sensing ranges.*

Depended on the coverage objectives and applications, three types of coverage problems can be defined : (*i*) Area coverage, (*ii*) Point coverage and (*iii*) Barrier coverage [19].

The main goal in the ***area coverage*** problem is to cover the whole area. Full or partial coverage can be required up to the application requirements. We talk about full coverage if every location in the area is covered at least by one sensor. On the other hand, partial coverage is used when full coverage of a given area is not required. In this case, we just need to cover some percentage of the entire area.

In the ***point coverage*** problem, the goal is to cover a set of points or targets with known locations that need to be monitored. The target can be static (i.e. always stay at the same location) or mobile (i.e. it can change its location).

In some applications such as intrusion detection and border surveillance, sensors are deployed to detect targets as they cross a barrier. The main goal here is to minimize the probability of undetected penetration through the barrier, which is usually a long belt region. This is defined to as the ***barrier coverage problem*** [31].

2.1.4 Deployment for an area coverage problem

In this work, we focus only on area coverage issues. In the area coverage problems, the main goal is to maximize the coverage rate especially when the number of sensors is not sufficient to achieve the full coverage. The problem of maximizing the coverage rate was addressed in several works, using either Virtual Force Algorithm (VFA) [61], [64], [22], [34] or geometrical approaches [57], [33], [4] which manage the pairwise distance between any pair of nodes and locally arrange the network topology as a triangle tessellation. In this work, we focus only on the VFA, 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 shown in Figure 2.2. It is worth to mention that the ideal deployment will be achieved if the proposed approach converges to a desired distance D_{th} i.e., if after some iterations the euclidean distance between any pair of robots is equal to a desired distance D_{th}.

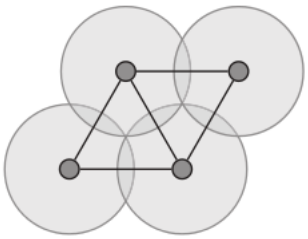


FIGURE 2.2: Ideal deployment for an area coverage

2.1.5 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 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 [64]. 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 :

$$F_{ij} = -\nabla V_{ij} \tag{2.7}$$

According to the traditional VFA, the force $\overrightarrow{F_{ij}}$ is given as :

$$\overrightarrow{F_{ij}} = \begin{cases} (w_a(d(i,j) - D_{th}), \varphi_{ij}), & if\ d(i,j) > D_{th} \\ 0, & if\ d(i,j) = D_{th} \\ (w_r(d(i,j) - D_{th}), \varphi_{ij} + \pi), & if\ d(i,j) < D_{th} \end{cases} \tag{2.8}$$

where :

- w_a is the virtual force attractive coefficient
- w_r is the virtual force repulsive coefficient
- φ_{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 [15]. Therefore, we will present a new modified version of VFA later in the next chapter.

2.2 Data prediction models

According to Cisco's prediction, the number of internet-connected devices would exceed 50 billion by 2020 [16]. If the number of data generated by these devices follows also this growth, IoHT will not be able to support all their wireless communications. Furthermore, they have limited resources in terms of CPU, memory, storage and power resources. It is therefore necessary to have an effective communication model for these constrained devices. Data reduction is one of the best approaches for achieving this efficiency. Indeed, reducing the amount of data to be transmitted over the network will reduce the energy spent on communications which is considered as the most energy consuming part. To reduce the devices' transmitted data, many approaches on data aggregation and compression have been proposed in the literature [53], [36], [40], [28]. However, these approaches do not take into account the most distinctive aspects of IoHT: (*i*) devices are constrained and (*ii*) the data collected and transmitted by the devices can be predicted using a very simple algorithm. The use of such an algorithm avoids any unnecessary communication that data compression and aggregation approaches are not able to eliminate. Therefore, we will focus only on existing prediction models in this Thesis.

There are many prediction models in the literature, ranging from the very simple to the complex one. In most cases, they have been developed in specific areas for specific purposes and can be classified into two main categories : (*i*) *time series models* and (*ii*) *cross-sectional models*.

2.2.1 Time series models

Time series predictions are used when we want to estimate a variable X that is changing over time. Each observation is often represented as x_t, where the observed value x is indexed by the time t at which it was made. The well known models in this category include **average**, **constant**, **linear**, **exponential smoothing**, **autoregressive**, **moving average** and **ARIMA models** [27].

2.2.1.1 Average prediction model

Average model is one of the simplest models. Forecasts of all future values are equal to the mean of the historical data and can be defined as :

$$\hat{x}_{t+k|T} = \bar{x} = \frac{1}{n}\sum_{i=1}^{T} x_i \tag{2.9}$$

where

- the variable X contains the time series
- $\hat{x}_{t+k|T}$ is the estimate of $\hat{x}_{t+k}$ based on the data $x_1,\ x_2, ...,\ x_T$
- n is the number of the historical data

2.2.1.2 Constant prediction model

Constant prediction model assumes that no changes will happen in the environment, and forecasts that the measurements in the future will be the

same as it was in the last observation. That is,

$$\hat{x}_{t+k} = x_t \tag{2.10}$$

where x_t is the last observed value.

2.2.1.3 Linear prediction model

Unlike the constant model, the linear prediction model assumes that the measured value may change in the future but has a linear component which does not vary in time. That is, the prediction can be defined as:

$$\hat{x}_{t+k} = (x_t - x_{t-1})k + x_t \tag{2.11}$$

2.2.1.4 Exponential smoothing model

Unlike constant and average models, in Exponential smoothing model, predictions are calculated using weighted averages where the weights decrease exponentially as we go back in time i.e. the smallest weights are associated with the oldest observations :

$$\hat{x}_{t+1} = \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + ... \tag{2.12}$$

where $\alpha \in [0; 1]$ is the smoothing constant.

It is to highlight that the forecast equation (2.12) is equal to a weighted average between the most recent observation x_t and the most recent forecast $\hat{x}_t$ i.e.

$$\hat{x}_{t+1} = \alpha x_t + (1-\alpha)\hat{x}_t \tag{2.13}$$

Further details and proof can be found in [27].

2.2.1.5 Autoregressive model

In different regression models (see subsection 2.2.2.1 and 2.2.2.2 for more details), the variable of interest is predicted using a linear combination of predictors. On the other hand, in a autoregression model, the variable of interest is predicted using a linear combination of *past values* of the variable. Hence, the model's name means that it is a regression of the variable against itself. Thus an autoregressive model of order p can be written as :

$$\hat{x}_t = c + \theta_1 x_{t-1} + \theta_2 x_{t-2} + \ldots + \theta_p x_{t-p} + w_t \tag{2.14}$$

where c is a constant and w_t is a white noise. This model is well known also as an **AR(p) model**.

2.2.1.6 Moving average model

Moving average model use the same idea than AR(p) model. However, instead of using past values of the forecast variable in a regression, it uses past forecast errors. The moving average model is called also **MA(q) model** and can be defined as:

$$\hat{x}_t = c + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \ldots + \theta_q e_{t-q} \tag{2.15}$$

2.2.1.7 ARIMA model

An ARIMA model (for Autoregressive Integrated Moving Average) is a combination of an Autoregressive and a Moving Average models. It relates the present value of a series to past values and past prediction errors. To calculate the forecasts, the autoregressive (AR) part is used to take account of the magnitude of the last observations and their trend. The Moving Average (MA) part in turn is used to consider the impact of unobserved shocks that influenced their current state. The accuracy of the AR and MA models is conditioned to the stationarity of the data. Hence, if the series are not stationary we have to transform it to a (possibly) stationary series. For that, the first differences $y_t = x_t - x_{t-1}$ will be examined and the integrated part of the mode is the reverse of the differencing. The algebraic expression of the full model is as follows :

$$y_t = c + e_t + \sum_{i=1}^{p} \alpha_i y_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} \tag{2.16}$$

This model is called also an **ARIMA(p,d,q) model**, where

- p: order of the autoregressive part
- d: degree of first differencing involved
- q: order of the moving average part

Although Time series models have been very used for data prediction in Wireless Sensor Networks (WSNs) domain [29], [51], [55], [63], [59], they are unsuitable for the IoHT context. Indeed, time series models deal only with homogeneous data and neglect the heterogeneous aspect of IoHT scenario. However, the heterogeneity of IoHT devices requires that the employed prediction model has to be applicable also to a variety of applications with heterogeneous data. A concrete example of such application is the smart agriculture, where the various agricultural inputs are collected with different sensor devices. For these reasons, in this thesis, our proposed solutions will based only on one of the cross-sectional models described below.

2.2.2 Cross-sectional models

The main goal of cross-selectional model is to infer the value of something we have not observed (called **hidden variable**), using the information on the cases that we have observed (called **evidence**). Examples of cross-sectional data include **Simple regression**, **Multiple regression**, **Artificial Neural Networks** and **Graphical models**.

2.2.2.1 Simple regression

A simple regression model consists in inferring an hidden variable y assuming that it has a linear relation with an observed variable x:

$$y = \beta_0 + \beta_1 x + \varepsilon \tag{2.17}$$

where

- β_0 and β_1 are the intercept and the slope of the line respectively
- ε is a random error

We refer the model as a simple regression since it only allows one predictor variable X.

2.2.2.2 Multiple regression

Unlike single regression, in multiple regression model, there is one variable to be forecast and several predictor variables. The algebraic expression of the multiple regression model is as follows

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \varepsilon \tag{2.18}$$

where y is the variable to be forecast, ε is a random error and $x_1..., x_k$ are the k predictor variables.

2.2.2.3 Artificial Neural Networks

An artificial neural networks (ANN) was inspired by the human brain and was designed as a computational model to solve specific problems. Its architecture is defined by (*i*) a basic processing elements called artificial neurons and (*ii*) the way in which they are interconnected. The output value of a neuron is given by:

$$output = f(\sum_i w_i x_i + b) = f(W^T X + b) \tag{2.19}$$

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 [1]. A typical artificial neuron and a basic ANN are illustrated in Figure 2.3.

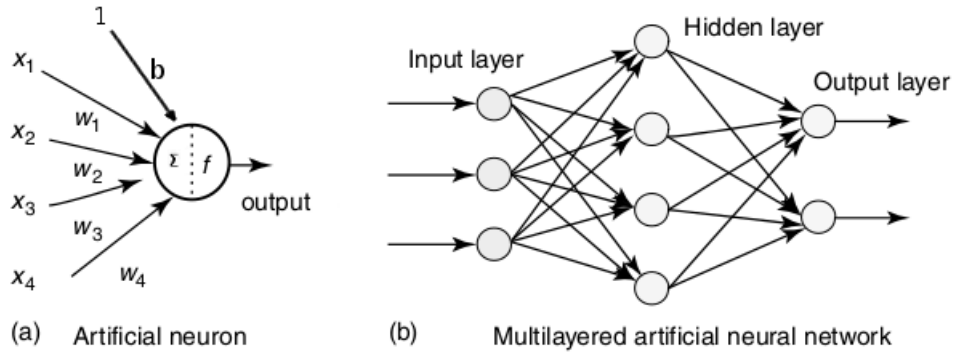


FIGURE 2.3: Architecture of an Artificial Neural Network

A key element of an artificial neural network is its ability to learn. This means 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 backpropagation algorithm [47] is one of the most used.

2.2.2.4 Graphical models

Graphical models (GM) are schematic representations of probability distributions. They consist of nodes connected by either directed or undirected edges. Each node represents a random variable, and the edges represent probabilistic relationships among variables. Models which are comprised of directed edges are known as *Bayesian networks*, whilst models that are composed of undirected edges are known as *Markov Random Fields* (MRF) [56], [9], [60], [25].

2.2.2.4.1 Bayesian networks

As said previously, Bayesian Network (BN) is a graphical model which contains only directed edges and without directed cycles. Its directed property makes it suitable when modeling the dependency relationships between random variables. Hence, the main goal of BN is to graphically represent a set of N random variables in $X = \{x_1, x_2, ..., x_N\}$, and the independencies structure between the variables in X. From the BN's independencies structure, the form of the joint probability distribution over the variables in X will be extracted. To sum up, a BN for a set of N random variables $X = \{x_1, x_2, ..., x_N\}$ consists of two things :

- a set of N nodes
- a set of directed edges between the nodes that encode the conditional independence statements associated with the variables in X

The notation x_i is used to represent both the variable and its corresponding node. Throughout this thesis, the words *'variable'* and *'node'* will be used interchangeably. Figure 2.4 shows an example of a Bayesian network with six random variables $\{x_1, x_2, x_3, x_4, x_5, x_6\}$.

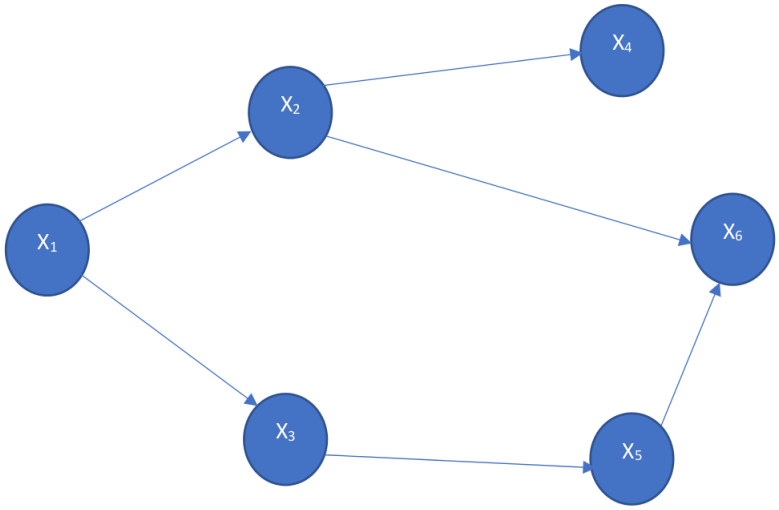


FIGURE 2.4: An example of a Bayesian network

By using this example, the joint probability $p(X) \equiv p(x_1, x_2, x_3, x_4, x_5, x_6)$ is just the product of all the probabilities of the parent nodes and all the local conditional probabilities :

$$p(X) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_1)p(x_4 \mid x_2)p(x_5 \mid x_3)p(x_6 \mid x_2, x_5) \quad (2.20)$$

In more general, the factorized equation for a Bayesian network with N random variables x_i is

$$p(x_1, x_2, ..., x_N) = \prod_{i=1}^{N} p(x_i \mid \Gamma_i) \quad (2.21)$$

where Γ_i is the set of parent nodes of the node x_i.

2.2.2.4.2 Markov Random Fields

In contrast to Bayesian Network, A Markov Random Fields (MRF) is an undirected graph i.e. contains only undirected edges. Figure 2.5 shows an example of a MRF with six random variables $\{x_1, x_2, x_3, x_4, x_5, x_6\}$.

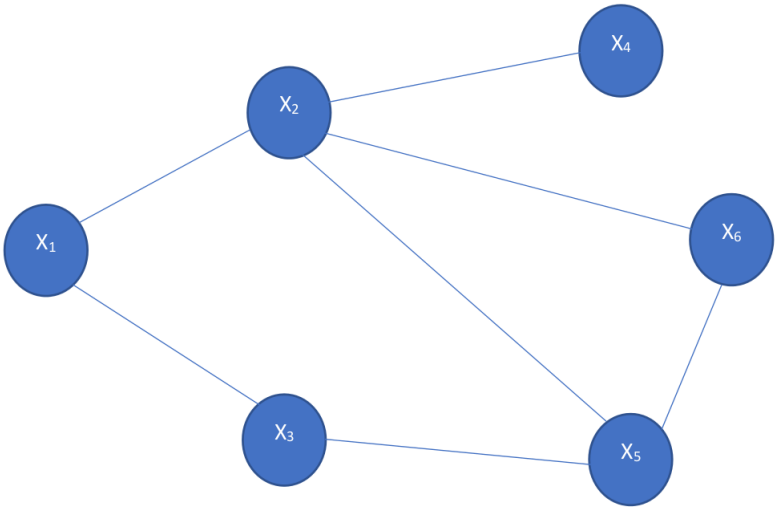


FIGURE 2.5: An example of a Markov random field

Here, the parent-child relationship no longer exists since there are no directed edges. Therefore, the local conditional probability distributions over nodes and their parents cannot be defined. Instead, functions of the variables over the *cliques* in the graph will be defined. These functions are called *potentials* and represent the probability distribution between the nodes in the *cliques*.

Definition 5 *A clique is a set of nodes in a graph that are fully connected i.e. every pair of node (i, j) in the set is connected by an edge. For example, the set (x_2, x_5, x_6) of the nodes in Figure 2.5 forms a clique. Here, this set forms also a maximal clique. A clique is maximal if no other nodes can be added without it no longer being a clique. For example, the nodes x_2 and x_5 in Figure 2.5 form a clique, but it is not maximal since node x_6 can be added to it*

If we denote $\psi(X_s)$ the potential function over a clique composed of nodes in the set X_s, based on the remarkable Hammersley-Clifford theorem, the joint probability distribution $p(X)$ of a Markov Random Fields with a set of maximal cliques C is the normalized product of the potentials over all the cliques in C:

$$p(X) = \frac{1}{Z} \prod_{c \in C} \psi_c(X_c) \tag{2.22}$$

Where X_c is the set of nodes belonging to the clique indexed by c and $Z = \sum_x \prod_{c \in C} \psi_c(X_c)$ is the normalization constant.

By using the formula (2.22) above, the joint probability distribution $p(X)$ of the MRF model presented in Figure 2.5 can be defined as :

$$p(X) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{13}(x_1, x_3) \psi_{24}(x_2, x_4) \psi_{256}(x_2, x_5, x_6) \tag{2.23}$$

It should be noted that for simplicity, in this thesis, we consider only the pairwise MRF, *i.e.* MRF with the maximum clique of two nodes.

In an inference problem, it is assumed that some values y_i of the node i are known, and one wants to infer other quantities x_i on the same node i [60]. It is also assumed that there is some statistical dependence between x_i and y_i, which written as an *evidence function* $\theta(x_i, y_i)$ (in literature, the evidence function is often written as $\theta(x_i)$ since y_i is already known and can be fixed). Hence, the joint distribution p(X) of a pairwise MRF model (knowing some evidence Y) is given as:

$$p(X) = \frac{1}{Z} \prod_i \theta_i(x_i) \prod_{i,j \in E} \psi_{ij}(x_i, x_j) \tag{2.24}$$

where Z is the normalization constant, $\theta_i(x_i)$ is the evidence function, E is the set of edges encoding the statistical dependencies between two hidden nodes i and j, and $\psi_{ij}(\cdot)$ represents the potential function. Note that the graphical model parameters (*i.e.*, θ_i and ψ_{ij}) can be estimated from the observed data by using a learning algorithm like in [24].

Figure 2.6 illustrates a graphical depiction of the model described above. The filled-in circles represent the observation nodes (*i.e.*, y_i) and the empty circles represent the hidden nodes (*i.e.*, x_i). The potential functions are associated with the links between x_i whilst the evidence functions are associated with the links between y_i and x_i.

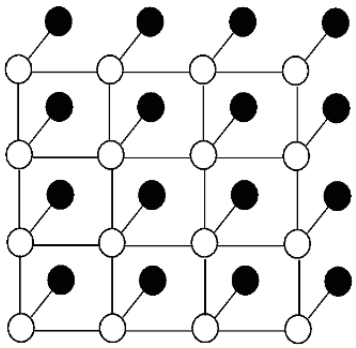


FIGURE 2.6: An example of MRF model with observed and hidden nodes.

2.2.2.4.3 Factor Graphs

Definition 6 *A Factor graph (FG) is a bipartite graph which explicitly expresses how a global function of many variables factors into a product of local functions [30].*

By using this definition, on can assume that a global function $p(x_1, x_2, x_3, x_4, x_5, x_6)$ of six variables can be written, for example, as a product of four functions ψ_{12}, ψ_{13}, ψ_{24} and ψ_{256}

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = \psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{24}(x_2, x_4)\psi_{256}(x_2, x_{5,} x_6) \tag{2.25}$$

This factorization form was obtained by using the MRF model presented in Figure 2.5. This conclude that an MRF model can be converted into equivalent factor graph. Figure 2.7 shows the equivalent factor graph of the MRF

model in Figure 2.5. The variable nodes are denoted by circles, while the functions nodes for each factor are denoted by squares.

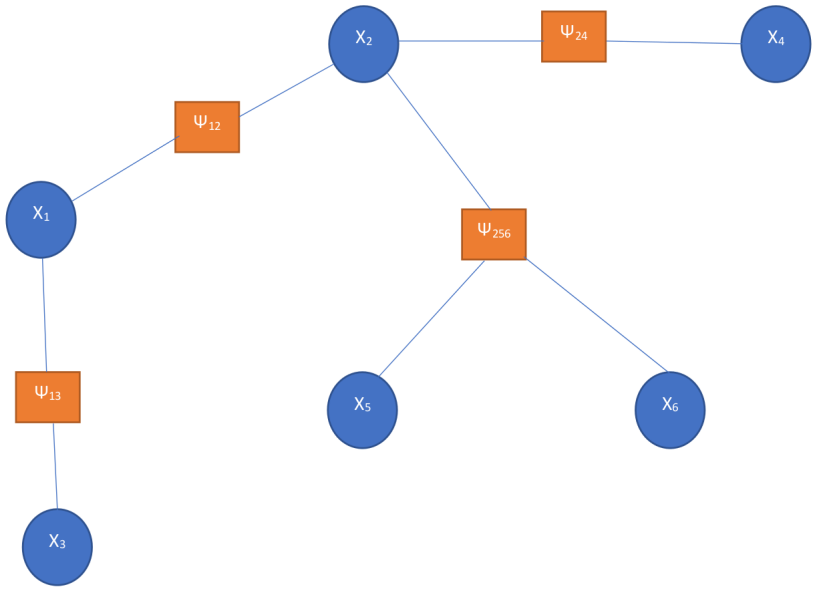


FIGURE 2.7: The factor graph representation of Figure 2.5

The joint probability of a factor graph of N variables with M functions can be therefore defined as:

$$p(X) = \frac{1}{Z} \prod_{a=1}^{M} \psi_a(X_a) \tag{2.26}$$

2.2.2.5 Inference

The main objective when working on a graphic model is to make an inference. Mathematically, this consists of computing the marginal probability which is defined as the sums on all the possible states of all the other nodes in the model. In Formula (2.27), for example, we consider a model composed of n nodes and we want to calculate the marginal probability of the last node i.e x_n. To this end, it is necessary to sum over all possible states of the $n-1$ nodes in the model.

$$p(x_n) = \sum_{x_1} \sum_{x_2} ... \sum_{x_{n-1}} p(x_1, x_2, x_3, ..., x_n) \tag{2.27}$$

Obviously, using (2.27), the complexity of a complete enumeration of all possible assignments to the whole graph is $O(|\mathcal{Y}|^{n-1})$, which is intractable for most choices of n. Therefore, we need a faster algorithm like Belief Propagation (BP) for computing the marginal probability. The BP algorithm computes the marginal probability with a time that only increases linearly with the number of nodes in the model.

2.2.2.5.1 Belief Propagation

Belief Propagation (BP) is a classic algorithm for performing inference

on graphical models that are tree structured or can be represented as tree-structured factor graphs [60]. It uses a message-passing algorithm to compute the marginal probability distribution of any query node [1]. The marginal probability at each i-th node (i.e. the belief), is computed through all the incoming messages from the neighboring nodes and the local evidence (Formula (2.28)) :

$$p(x_i) = belief(x_i) = k\theta_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i) \tag{2.28}$$

where k is a normalization constant, $N(i)$ denotes the neighbors of node i and m_{ji} represents the message from a hidden node j to hidden node i about the state of node i should be in.

The messages m_{ji} are computed recursively by respecting the message update rule defined by the Formula (2.29) bellow :

$$m_{ij}(x_j) \leftarrow \sum_{x_i} \theta_i(x_i)\psi_{ij}(x_i, x_j) \prod_{u \in N(i), u \neq j} m_{ui}(x_i) \tag{2.29}$$

Figure 2.7 shows an illustration of message passing algorithm used in BP.

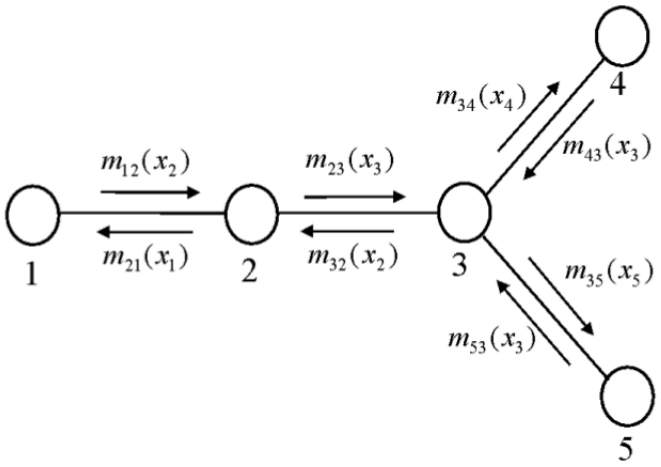


FIGURE 2.8: Message passing algorithm used in BP [60]

Now, to better understand the BP algorithm. Suppose we want to know the belief at the node 2 of the Figure 2.8. By using the Formula (2.28), we get

$$p(x_2) = belief(x_2) = k\theta_2(x_2)m_{12}(x_2)m_{32}(x_2) \tag{2.30}$$

[1] Query node refers to the node that we want to get the marginal probability distribution

Using the message update rule for m_{12} and m_{32}, we get

$$p(x_2) = k\theta_2(x_2) \sum_{x_1} \theta_1(x_1)\psi_{12}(x_1, x_2) \sum_{x_3} \theta_3(x_3)\psi_{32}(x_3, x_2)m_{43}(x_3)m_{53}(x_3) \tag{2.31}$$

Using the message update rule for m_{43} and m_{53}, we get

$$p(x_2) = k\theta_2(x_2) \sum_{x_1} \theta_1(x_1)\psi_{12}(x_1, x_2) \sum_{x_3} \theta_3(x_3)\psi_{32}(x_3, x_2) \sum_{x_4} \theta_4(x_{34})\psi_{43}(x_4, x_3) \sum_{x_5} \theta_5(x_5)\psi_{53}(x_5, x_3) \tag{2.32}$$

Finally, by organizing the sums, we can easily see that the belief at node 2 is the same as the exact marginal probability at node 2 :

$$belief(x_2) = p(x_2) = k \sum_{x_1,x_3,x_4,x_5} p(x_1, x_2, x_3, x_4, x_5) \tag{2.33}$$

2.2.3 Comparative study of prediction models

As we have seen in the previous section, the prediction models can be classified into two main categories: (i) time series and cross-selectional models. Times series models have been defined as a function of the past observations. Some known models have been presented, namely Autoregressive (AR), Moving Average (MA), Exponential Smoothing (ES) and Autoregressive Integrated Moving Average (ARIMA). These models have been widely used in Wireless Sensor Networks (WSN) field but unfortunately they are only suitable for homogeneous data. They therefore neglect the heterogeneous aspects of IoT. Indeed, with the advancement of current technologies, IoT is composed of heterogeneous devices ranging from small integrated sensor, RFID TAG and robot to powerful computers. For these reasons, it is preferable to work first with models which take into account the heterogeneity of the data. Hence the advantage of using cross-sectional models.

The main goal of cross-selectional model is to infer the value of variable we have not observed (called hidden variable), using the information on the cases that we have observed (called evidence). We have also presented some well known models such as linear regression models, Artificial Neural Network (ANN) and graphical models.

Linear regression models are very simple models but unfortunately they are ineffective when dealing with many hidden variables. By cons, ANN is very efficient especially if it has been well trained but it is too complex to be used on thin devices. Graphical models in turn are efficient and , if coupled with belief propagation (BP) algorithm, can computes the marginal probability with a time that increases only linearly with the number of nodes in the model. Table 2.1 below illustrates the Comparative study of all the data prediction models we have seen so far.

Classes	Models	Comments
Time series	AR, AM, ES, ARIMA	• There is only one data type • Neglect the heterogeneous aspect of IoT scenario
Cross-selectional	Linear regression	Ineffective when dealing with many hidden variables
	ANN	High complexity
	Graphical models	• Efficient • Low complexity if used with BP

TABLE 2.1: Comparative study of prediction models

2.3 Summary

First, we have seen the different methods for connectivity maintenance in the context of deploying a group of mobile robots. Then, we have discussed about the coverage problems and have presented existing deployment methods for an area coverage problem. Finally, we have addressed some well known data predictions models such as ARIMA, regressions, and graphical models.

Chapter 3

Global Connectivity Maintenance Among IoHT Mobile Robots

In this chapter, we provide answers to the first problematic posed in this thesis. To recall, the first question that we want to answer is : *How to efficiently maintain the communication coverage among IoHT mobile sensors or robots ?*

As already stated before, we address the connectivity maintenance problem in the context of deploying a group of mobile robots into a unknown environment to form a desired communication coverage for supporting efficient data sharing.

According to the properties of the Laplacian matrix we have seen in subsection 2.1.2, a graph $G(V, E)$ is well connected if we can coordinate robot's' motion in the network so that the algebraic connectivity λ_2 is always higher than zero. This property is a fundamental basis for many global connectivity maintenance approaches. Often, a gradient method is used to move the mobile robots in the direction maximizing the algebraic connectivity λ_2 [23]. By using this method, the maximum connectivity is ensured but unfortunately it also results in poor coverage. Indeed, maximizing the connectivity and the coverage simultaneously is difficult (if not impossible). Maximizing the collective coverage may lead to a poor communication quality and conversely (i.e. a very good communication may lead to a poor coverage). It follows that our goal is to capture the trade-off between collective coverage and communication quality. To achieve this, in this chapter, we propose two motion control strategies which maintain global connectivity between IoHT mobile robots to a desired distance and QoS level. The first approach is an IoT-based while the second is a distributed trained neural network controller.

3.1 Proposed solutions

3.1.1 IoT-based approach for global connectivity maintenance

Our first proposition exploits the nice property of the Laplacian matrix for maintaining desired wireless communication coverage among mobile robots. To this end, the proposed approach uses Cloud (or a gateway with high computation capability) to compute and monitor the connectivity of the overall multi-robot system. We assume that each mobile robot knows its own position by using GPS or other localization system. Probe packets

are used to allow mobile robots exchange their positions with their one-hop neighbors.

Each mobile robot in the system applies a distributed Virtual Force Algorithm 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 one-hop neighbor, the $i-th$ robot should move away from the 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 mobile robots. This simple control law generates a vector position $\overrightarrow{P_{ij}}$ such that the $i-th$ robot keeps the line of sight of the robot j. $\overrightarrow{P_{ij}}$ is defined as :

$$\overrightarrow{P_{ij}} = \begin{cases} (0.1 \times k \times \Delta d, \varphi_{ij}) & if\ d(i,j) > D_{th}\ and\ \Delta d > \epsilon \\ (k \times \Delta d, \varphi_{ji}) & if\ d(i,j) < D_{th}\ and\ \Delta d > \epsilon \end{cases} \tag{3.1}$$

where :

- $\Delta d =\mid d(i,j) - D_{th} \mid$
- φ_{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 original VFA, we set the attractive coefficient w_a to one tenth of repulsive coefficient k ($w_a = 0.1 \times k$).

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

$$\overrightarrow{P_i} = \sum_{j \in N_i} \overrightarrow{P_{ij}} \tag{3.2}$$

After calculating their new positions, each mobile robot sends the computed position to the Cloud (or gateway). Then, the Cloud (or gateway) computes the algebraic connectivity λ_2 of the mobile robots network according robots' new positions by using Formula (2.5). Each robot is allowed to move to their new positions if and only if the algebraic connectivity is greater than zero (i.e $\lambda_2 > 0$). This guarantees that global connectivity is always maintained throughout the deployment process.

The following algorithm summarizes our approach:

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

Phase I : Neighbor Discovery
  $MyNeighbor \leftarrow FindNeighbor(RobotId)$
Phase II : Compute the position $\overrightarrow{P_{ij}}$ between two robots
  $Compute\ \overrightarrow{P_{ij}}\ using\ Formula$ (3.1)
Phase III : Compute the new position $\overrightarrow{P_i}$
  $Compute\ \overrightarrow{P_i}\ using\ Formula$ (3.2)
Phase IV : Compute algebraic connectivity
  $Compute\ \lambda_2\ of\ the\ dynamic\ Laplacian\ matrix\ L(G)$
Phase V : Deployment
  if $\lambda_2 \geq 0$ **then**
    $move\ to\ \overrightarrow{P_i}$
  else
    $do\ not\ move$

3.1.2 ANN-based controller for global connectivity maintenance

The drawback of our previous approach is the use of an hybrid approach. In this case, an access problem to the Cloud (or to the gateway) will penalize all the deployment procedure. For example, a rescue operation may be difficult after a disaster when the access to the Cloud is not available. An approach which easily adapts to any type of situation and environment is more than necessary. Therefore, it is better to use a decentralized approach which is able of replicating the same performance as the hybrid approach. To meet this need, we provide an Artificial Neural Network (ANN) based technique which can perfectly mimic the behaviors of our IoT-based approach. The ANN-based approach is completely distributed and is trained from a set of data obtained by using the hybrid IoT-based approach. The trained ANN is constituted by 2 input units and 1 output unit. The 2 input units are $d(i,j)$ and φ_{ij}, while the output is $\overrightarrow{P_{ij}}$. Therefore, the trained ANN is executed locally for each mobile robot to control its movement according to its one-hop neighbor's distance $d(i,j)$ and angle φ_{ij}. When the position $\overrightarrow{P_{ij}}$ is estimated, the new position $\overrightarrow{P_i}$ of the robot is computed by using the formula (3.2). Then, the collective movement of all robots will allow our trained ANN to converge to the desired distance D_{th}. The global connectivity will also be kept if our ANN is well trained (i.e. if training error equals zero or near to zero). Backpropagation algorithm [47] has been used to train the neural network. The algorithm 2 below illustrates our ANN-based approach.

3.2 Simulation results

In this Section, we describe the simulation parameters and 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 mobile robot (hence to the desired communication quality matching D_{th}). We will see also how the density of mobile 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

Algorithm 2 ANN approach (runs every t units of time)

1: **Phase I : Neighbor Discovery**
2: $MyNeighbor \leftarrow FindNeighbor(RobotId)$
3: **Phase II : Estimate the position $\overrightarrow{P_{ij}}$ between two robots**
4: $\overrightarrow{P_{ij}} \leftarrow trained\_ann(d(i,j), \varphi_{ij})$
5: **Phase III : Compute the new position $\overrightarrow{P_i}$**
6: $Compute\ \overrightarrow{P_i}\ using\ Formula$ (3.2)
7: **Phase IV : Deployment**
8: $move\ to\ \overrightarrow{P_i}$

approaches will be compared to the approach described in [34] 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.

All the algorithms were implemented in version 2.29 of Network Simulator with Marco Fiore's patch [20], that reflect a realistic channel propagation and error model. This patch is used in order to provide the effect of interference and different thermal noises to compute the signal to interference plus noise ratio (SINR) and accounting for different bit error rate (BER) to SINR curves for the various codings employed [41]. Table 3.1 summarizes the parameters used in the simulations.

All of the obtained results are the average of 100 times simulations and we assume that the topology is totally connected at the beginning of each simulation. According to [14], so as to avoid the coverage hole, the only constraint is that $D_{th} \leq\sim 0.851C_R$. In this Thesis, we set D_{th} to the upper bound i.e. $D_{th} = 0.851C_R$, where C_R is the communication range of a robot.

We assess our techniques w.r.t. (*i*) the algebraic connectivity, (*ii*) the robot traveled distance, (*iii*) the average distance between any pair of robots, 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^2.

We can observe in the Figures [3.2 - 3.5](a) that the IoT-based approach always kept the global connectivity since it always take into 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. 3.1(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 always keep its value greater than zero. This condition is enough to maintain the global connectivity of the network.

Figures [3.2 - 3.5](c) illustrate the convergence of our algorithms to the desired distance throughout the simulation. We can notice that our approaches converge quickly to the aforementioned D_{th}, which is not always the case for EVFA. We can see also that the ANN-based approach mimics perfectly the behaviours of the IoT-based approach. This is due to the fact that our neural network has been well trained. It should be noted that the

Physical	Propagation	Two ray ground
	Error model	Real
	Antennas gain	$G_{Tx} = G_{Rx} = 1$
	Antennas height	$h_{Tx} = h_{Rx} = 1$ m
	Communication range	250 m
Statistics	Number of samples	100
	Simulation time	3000 s
	Confidence Interval	95%
Mobility	Computation of the new position	see Algorithms 1, 2
	Reducer coefficient α	$[0, 1]$
	D_{th}	212 m
Topology	Topology width	3 km
	Topology height	3 km
ANN	Layer number	4
	Input number	2
	Output number	1
	Neuron number in hidden layers	15
	Desired Error	0.00001
	Max epochs	10000
	Activation function	sigmoid symmetric
	Learning rate	0.2
	Training algorithm	backpropagation

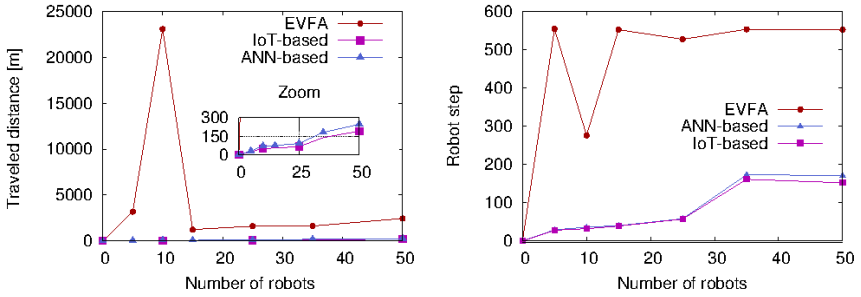
TABLE 3.1: Simulation parameters

convergence is acquired if and only if, for any pair of robots (u,v) in the network, $d(u, v) \in [D_{th} - \epsilon, D_{th} + \epsilon]$, where ϵ is the tolerance value used to avoid useless small movements.

Figures [3.2 - 3.5](d) show also that our proposed approaches can maintain the desired QoS level (RSSI) among neighboring robots. In this paper, the choice of the desired QoS level was made on the basis of the desired distance D_{th} (i.e. the RSSI measured at a distance D_{th}) but can be used, of course, independently. We can for example setup the desired QoS level based on the results in [52] which show that the packet receive rate is at least 85% for links with RSSI above 3.16 pw (ie -87dBm), using CC2420 single-chip RF transceiver, which is compliant with IEEE 802.15.4 standard.

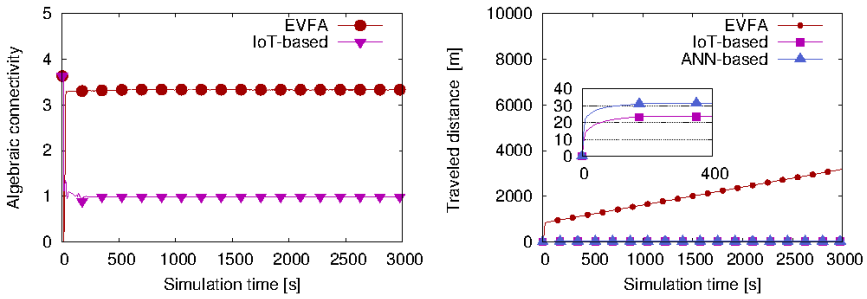
Figures [3.2-3.5](b) depict the distance traveled by a robot during the simulation time. By considering the relationship between energy and traveled distance, we can say that our approaches are energy efficient as compared to EVFA. However, it is observed that ANN consumes a bit more than IoT-based approach. This is because ANN has made a bit more step[1] to converge (see Fig. 3.1(b)).

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

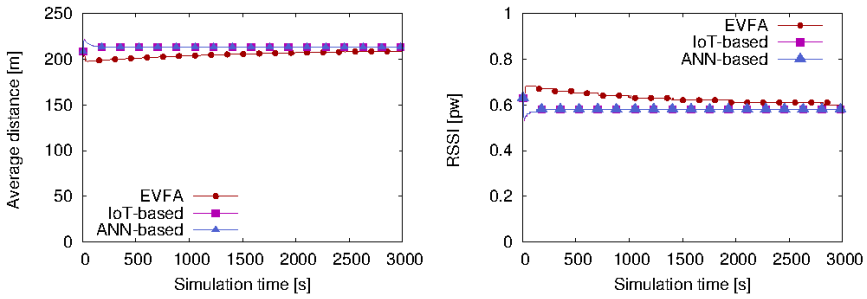


(a) Traveled distance (b) Robot step

FIGURE 3.1: Traveled distance and robot step according to the robots number

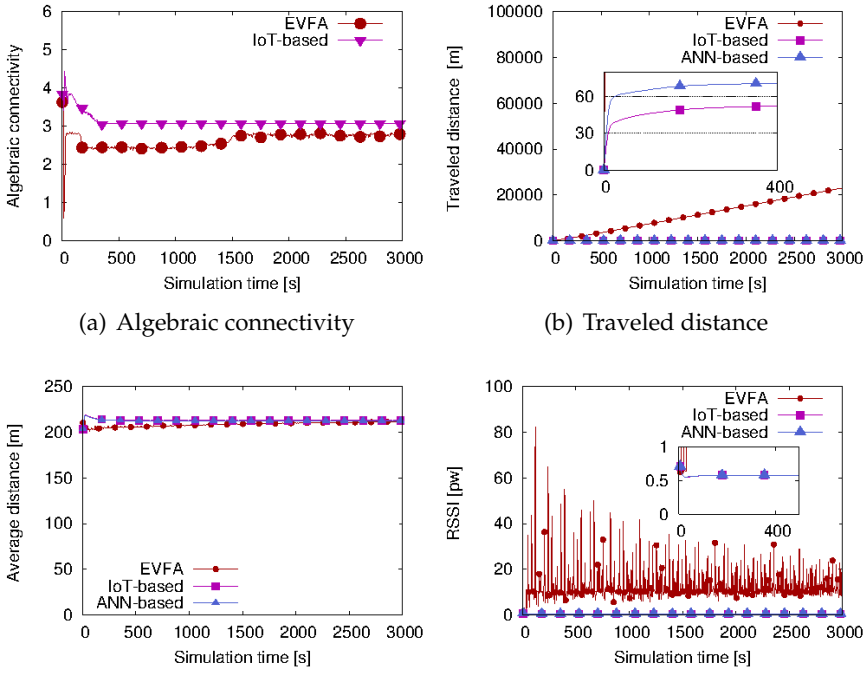


(a) Algebraic connectivity (b) Traveled distance



(c) Position (d) RSSI

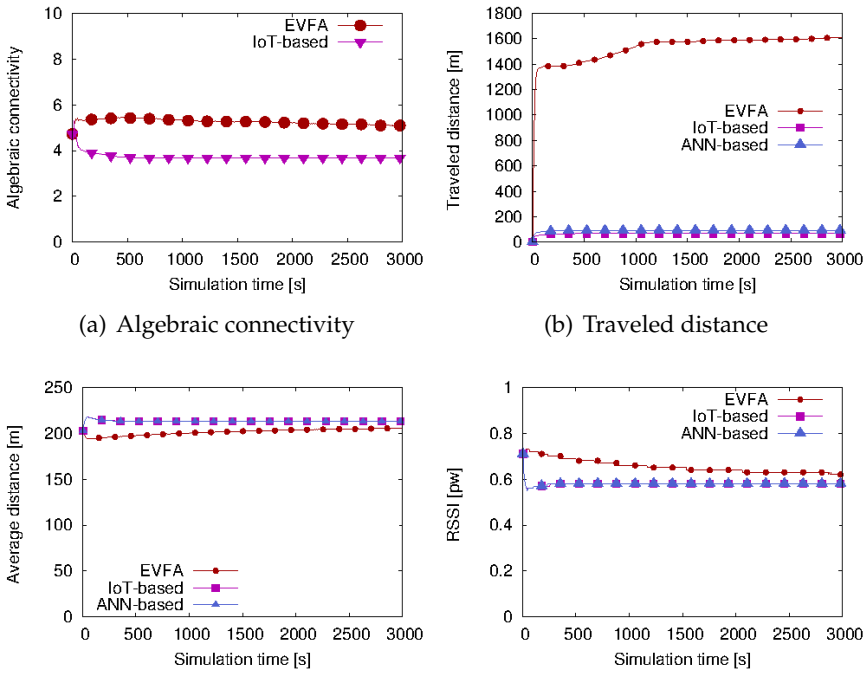
FIGURE 3.2: Simulation results obtained with 5 robots moving in 3x3 km area



(a) Algebraic connectivity (b) Traveled distance

(c) Position (d) RSSI

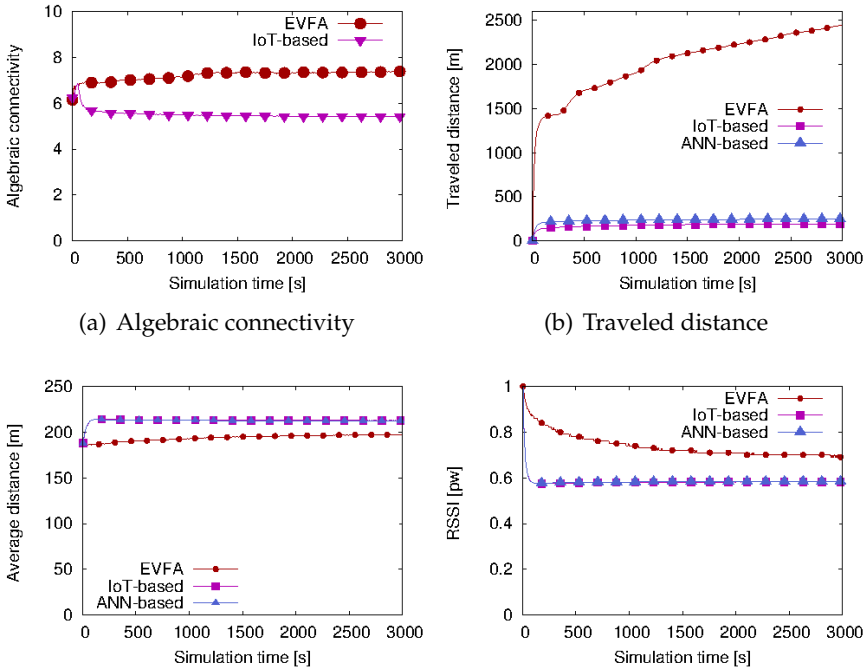
FIGURE 3.3: Simulation results obtained with 10 robots moving in 3x3 km area



(a) Algebraic connectivity (b) Traveled distance

(c) Position (d) RSSI

FIGURE 3.4: Simulation results obtained with 25 robots moving in 3x3 km area



(a) Algebraic connectivity (b) Traveled distance

(c) Position (d) RSSI

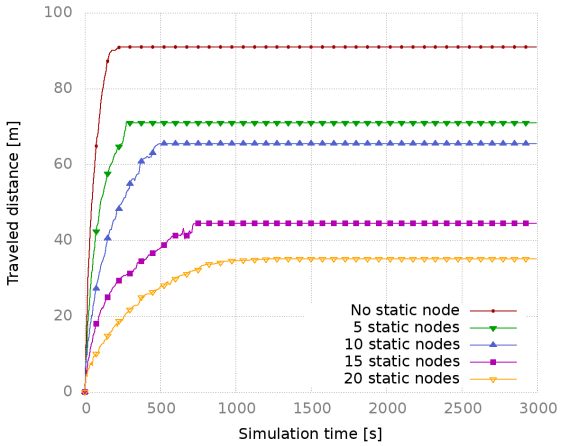
FIGURE 3.5: Simulation results obtained with 50 robots moving in 3x3 km area

All the results we have obtained so far (i.e. Figures [3.1 - 3.5]) are based on the assumption that all the IoHT devices have the ability to move. Now, let us consider that some devices can be static and some can be mobile. Then, let us examine the behaviors of the ANN-based algorithm[2] w.r.t (*i*) the robot traveled distance, (*ii*) the average distance, and (*iii*) the QoS level.

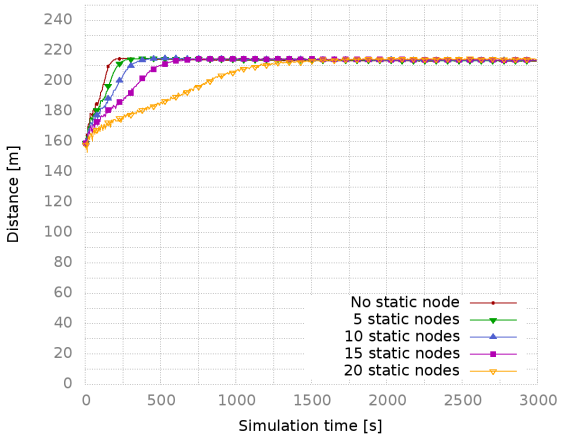
The study was carried out on a network of 25 nodes. To evaluate the behaviors of the algorithm, we have varied the number of static nodes in the network. Figures 3.6 illustrates the obtained results. It can be seen in Figure 3.6(a) that the average distance traveled by a robot decreases as a function of the number of static nodes in the network. This means that a robot moves less if most of the nodes in the network are static.

Figures 3.6(a) and 3.6(b) demonstrate that our algorithm allows the mobile robots converge to the desired distance and QoS even in the presence of static nodes. However, we can see that the convergence time depends strongly on the number of static nodes. A robot takes longer to converge when there is more static node in the network. It can be seen that the convergence time increases depending on the number of static nodes.

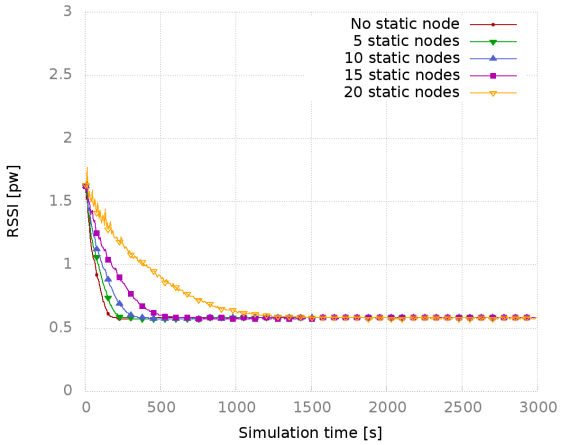
[2]We only consider the ANN-based approach since it perfectly imitates the behavior of IoT-based approach.



(a) Traveled distance



(b) Position



(c) RSSI

FIGURE 3.6: Simulation results obtained with 25 IoHTs devices using heterogeneous scenarios

3.3 Summary

In this chapter, we have implemented an IoT-based and ANN control schemes to maintain global connectivity among multiple IoHT devices. The proposed approaches have captured the trade-off between network coverage and communication quality expressed as RSSI level. Moreover, they allowed the IoHT mobile robots converge to the desired distance and communication quality even in the presence of static nodes in the network. Through extensive simulations with the NS-2 network simulator, we have showed that our approaches outperform the EVFA approach proposed in [34], in terms of traveled distance and convergence time. Furthermore, our proposed methods always maintain the global connectivity throughout the simulation.

In order to improve the ANN-based approach, in the next chapter, an Neuro-Dominating Set algorithm will be discussed. The main goal is to exploit the beneficial effect of using different behaviors among IoHT mobile devices.

Chapter 4

Neuro-Dominating Set Scheme For a Fast and Efficient Robot Deployment in IoHT

In the previous chapter, we have proposed two algorithms for maintaining global connectivity in the context of deploying a group of IoHT mobile robots into a realistic environment. The first approach was an hybrid IoT-based approach while the second was a distributed trained neural network controller (ANN-based). The second approach (i.e. the ANN-based) was proposed to avoid the single point of failure existing in the IoT-based. Like the IoT-based approach, it has also allowed the mobile robots to reach the desired distance and desired communication quality while ensuring global connectivity. However, in order to achieve these objectives, all the mobile robots in the network robots were endowed with the same algorithm and behavior.

In this chapter, we go a step further by studying the beneficial effect of using different behaviors in the IoHT concept. Therefore, we propose a Neuro-Dominating Set (NDS) approach for the global connectivity maintenance and robots' motion control. We use the term Neuro-Dominating Set to describe our approach, since it is inspired by both neural network and dominating set strategies. Each IoHT mobile device adopts different strategy according whether it is a dominating robot or an dominated robot. This heterogeneity of strategy may improve global efficiency in terms of traveled distance while keeping more or less similar convergence time w.r.t the ANN-based approach.

4.1 Motivation and Background

Biological societies show various examples of diversity allowing participants to self organize and solve global problems in a more efficient way [39], [8], [12]. The use of this diversity or heterogeneity in the IoHT context may therefore open possibilities to solve more complex tasks since different skills and behaviours can be combined. In the literature, the definition of the heterogeneity often varies according authors and used applications [18], [43], [44]. Heterogeneity can be defined in terms of variety in capacity, hardware, size, cognition, behaviour, etc.

In this chapter, heterogeneity refers to difference in behaviour since we adopt different roles among mobile robot to enable them better achieve the global system performance. It follows that our main goal here is to decrease the overall traveled distance while maintaining network connectivity and

an acceptable convergence time, still in the context of deploying a group of IoHT mobile robots in a realistic environment to form a desired communication coverage for providing efficient data sharing.

4.1.1 Dominating Sets in Graphs

Definition 7 *A set $D \subseteq V$ of vertices in a graph $G(V, E)$ is called a Dominating Set (DS) if every vertex $v \in V$ is either an element of D or has a neighbor in it [35].*

For instance, Figure 4.1 (a) represents an undominating set, since not all the vertexes $v \in V$ have a neighbor in D (*e.g.*, the node F does not have a neighbor in D). The nodes in a DS are called *dominating* (*i.e.*, the red nodes in Figure 4.1), the others are called *dominated* (*i.e.*, the blue nodes in Figure 4.1). This kind of set is not necessarily connected (see Figure 4.1 (b)). We refer to a Connected Dominating Set (CDS) when the subgraph induced by D is connected, as represented in Figure 4.1 (c). On every connected graph $G(V, E)$ it can be found at least one Dominating Set since the set of all vertices is dominating according to the definition.

In this work, we compute locally DS while trying to minimize its size. The computation of DS is also periodically executed in order to take into account the variation of the graph $G(V, E)$ over time. In this work, we compute locally DS while trying to minimize its size. The computation of DS is periodically executed in order to take into account the variation of the graph $G(V, E)$ over time.

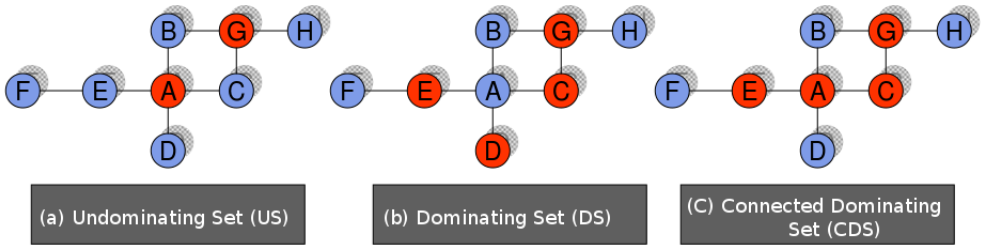


FIGURE 4.1: Dominating Set

4.2 Neuro-Dominating Set Scheme

It is easy to observe that if a subset of the robot in the network moves less than the other robots, the global traveled distance will be decreased and hence the energy consumption as well. In order to benefit from this effect, we need to exploit the good properties of Dominating Set.

Our mobile robot network can be decomposed into two subsets i.e., (i) *dominating set* and (ii) *dominated set*. Let A be the *dominating set* and $m = \|A\|$ its cardinality (respectively, B the *dominated set* and $l = \|B\|$ its cardinality). If we have n mobile robots in the network and the computed dominating set has a minimum size, the two formulas below are always true :

- $n = m + l$
- $m \leq l$

Now, let $Dist_{Tot}(t)$ be the total distance traveled by all robots in the network at time t. $Dist_{Tot}(t)$ can be defined as :

$$Dist_{Tot}(t) = d_1(t) + d_2(t) + ... + d_n(t) = \|\overrightarrow{P_1}(t)\| + \|\overrightarrow{P_2}(t)\| + ... + \|\overrightarrow{P_n}(t)\| = \sum_{i=1}^{n} \|\overrightarrow{P_i}(t)\| = \sum_{i=1}^{n} d_i(t) \tag{4.1}$$

where

- $d_i(t)$ is the traveled distance of robot i at time t.
- $\overrightarrow{P_i}(t)$ is the vector position of robot i with respect to all its neighbors.
- $d_i(t) = \|\overrightarrow{P_i}(t)\|$

Considering that some robots are dominating and some are dominated, we have:

$$Dist_{Tot}(t) = \sum_{a\,in\,A} d_a(t) + \sum_{b\,in\,B} d_b(t) \tag{4.2}$$

where

- $d_a(t)$ is the traveled distance of a dominating robot a at time t.
- $d_b(t)$ is the traveled distance of a dominated robot b at time t.

Two possible approaches can be deduced from formula (4.2) *i.e.*, (*i*) dominating robots move less distance than dominated robots so that

$$\sum_{a\,in\,A} d_a(t) < \sum_{b\,in\,B} d_b(t), \tag{4.3}$$

and (*ii*) the dominated robots move less distance than dominating robots *i.e.*,

$$\sum_{b\,in\,B} d_b(t) < \sum_{a\,in\,A} d_a(t). \tag{4.4}$$

Let us call NDS-A the first approach (i.e dominating robots move less distance than dominated robots) and NDS-B the second approach.

The total traveled distance using NDS-A approach can be defined as:

$$Dist_{Tot_A}^{NDS-A}(t) = \alpha \sum_{a\,in\,A} d_a(t) + \sum_{b\,in\,B} d_b(t), \tag{4.5}$$

whille, the total traveled distance using NDS-B approach is defined as:

$$Dist_{Tot_B}^{NDS-B}(t) = \sum_{a\,in\,A} d_a(t) + \alpha \sum_{b\,in\,B} d_b(t), \tag{4.6}$$

where $\alpha \in [0, 1]$ is a discount factor.

To know which approach is the most efficient in term of minimization of traveled distance, let us subtract (4.6) and (4.5), i.e. :

$$Dist_{Tot_B}^{NDS-B}(t) - Dist_{Tot_A}^{NDS-A}(t) = (1-\alpha)\left[\sum_{a\,in\,A} d_a(t) - \sum_{b\,in\,B} d_b(t)\right] \tag{4.7}$$

As $(1-\alpha) \geq 0$, we will focus our study on $\sum_{a\,in\,A} d_a(t) - \sum_{b\,in\,B} d_b(t)$.

We assume that all the robots have the same velocity either they are dominating or dominated. If $a \in A$ is a dominating robot, it is connected at least to a robot in B. Therefore, d_a is a magnitude of a resultant force generated by one or more robot in B. This means that :

$$\sum_{a\,in\,A} d_a(t) \leq \sum_{b\,in\,B} d_b(t) \tag{4.8}$$

Therefore, we get:

$$Dist_{Tot_B}^{NDS-B}(t) \leq Dist_{Tot_A}^{NDS-A}(t). \tag{4.9}$$

The above formula proves that better efficiency, in term of traveled distance, is obtained when NDS-B approach is used. In other word, the global traveled distance of our robot network will improve if the dominated robots move less distance than the dominating robots.

4.2.1 NDS algorithm

As mentioned before, our goal is to decrease the overall traveled distance while maintaining an acceptable convergence time. As in the previous chapter, the convergence is acquired if and only if, for any pair of robots (u,v) in the network, $d(u,v) \in [D_{th} - \epsilon,\ D_{th} - \epsilon]$, where ϵ is the tolerance value.

Our proposed NDS algorithm is a mix of the ANN-based approach and dominating set strategy. In this algorithm, NDS-B approach will be used since its effectiveness w.r.t. NDS-A has already been proved in the previous section. In this case, the dominated robots have to travel less distance than the dominating robots.

The major steps of the chosen NDS-B algorithm that runs in each robot are enlisted as follows *i.e.*, (*i*) neighbor discovery, (*ii*) computation of the dominating set, (*iii*) computation of the vector position between two neighboring robots, (*iv*) computation of the robot new position, and (*v*) movement towards the computed position.

Algorithm 3 summarizes our proposed NDS-B approach. First, the i-th robot needs to know its one-hop neighbors (*i.e.*, Phase I). Then, in Phase II it computes the minimum dominating set (MDS) through a distributed greedy algorithm that approximates the MDS. The computation of the MDS is NP-hard, therefore a lot of works in the literature proposed an approximation version of MDS. The aim here is to show that, with the help of the dominant set approach, we can reduce the distance traveled by a robot. After identifying the MDS, the i-th robot computes the force $\overrightarrow{P_{ij}}$ that it has to exert with its j-th neighbor by using a trained ANN (*i.e.*, Phase III). The computation of $\overrightarrow{P_{ij}}$ is different according whether the i-th robot is dominating or dominated. If the i-th robot has many neighbors, its new position $\overrightarrow{P_i}$ is calculated according to Formula (3.2). Finally, the i-th robot moves to the computed position (*i.e.*, Phase IV).

It is worth to mention that each robot knows its own position by using GPS (in case of outdoor environments) or other localization systems (for indoor environments). Beacon messages are used to allow robots exchange their positions with their one-hop neighbors.

Algorithm 3 Neuro-Dominating Set approach (runs every t units of time)

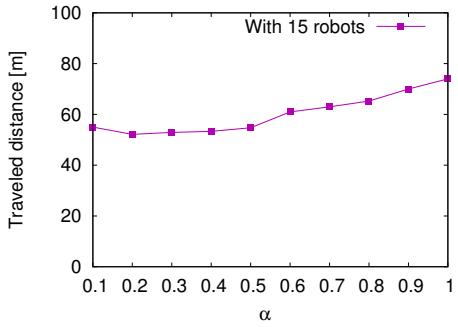
Phase I : Neighbor Discovery
  $MyNeighbor \leftarrow FindNeighbor(RobotId)$
Phase II : Compute the Dominating Set
  $DS \leftarrow GetDS()$
Phase III : Compute the position $\overrightarrow{P_{ij}}$ between two robots
  if $RobotId \in DS$ **then**
    $\overrightarrow{P_{ij}} \leftarrow trained\_ann(d(i,j), \varphi_{ij})$
  else
    $\overrightarrow{P_{ij}} = \alpha * [trained\_ann(d(i,j), \varphi_{ij})]$
Phase IV : Compute the new position $\overrightarrow{P_i}$
  $Compute \ \overrightarrow{P_i} \ using \ Formula$ (3.2)
Phase V : Deployment
  $move \ to \ \overrightarrow{P_i}$

4.3 Simulation results

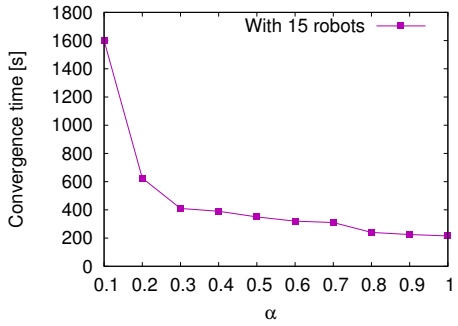
In this section, we provide the simulation results of our algorithm. We are interested in studying how our approach converges to the desired distance D_{th} between any pair of robots. The influence of the parameter α in our proposed approach will be also highlighted. Our NDS approach will be compared to the ANN-Based approach described in the section 3.1.2 of the previous chapter. We use exactly the same simulation parameters as presented in section 3.2.

We assess our algorithm w.r.t. (*i*) the minimized robot traveled distance, (*ii*) the average distance between any pair of robots, and (*iii*) the QoS level expressed in terms of RSSI (Received Signal Strength Indicator).

The discount factor α is one of the most important parameters of our proposed algorithm. Indeed, its value plays an important role for the efficiency improvement. If α is small, the traveled distance decreases but the convergence time increases accordingly since the system need more time to converge in this case. Figure 4.2 and Figure 4.3 illustrate these results for different values of $\alpha \in [0.1; 1]$. We can say that for $\alpha \rightarrow 1$, the traveled distance increases, while the convergence time reduces to the minimum value (i.e 200s). Notice that the choice of evaluating α on a network of 15 robots was arbitrary. We can actually evaluate it on different number of robots but the conclusion will remain the same, *i.e.*, a small value of α decreases the traveled distance but increases the convergence time. As a result, the value of α has to capture the trade-off between the traveled distance and the convergence time.



(a) Traveled distance



(b) Convergence time

FIGURE 4.2: Traveled distance and Convergence time according the values of α

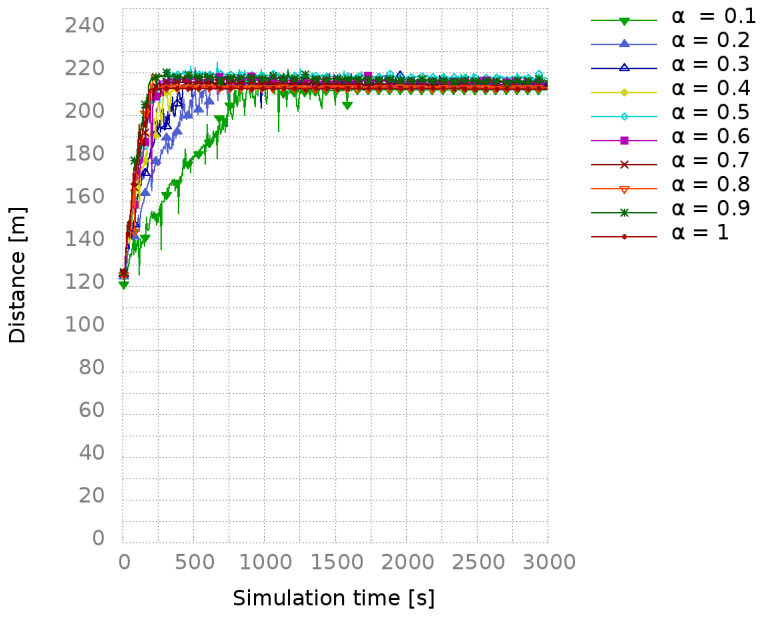


FIGURE 4.3: Variation of robot position according the values of α. Scenario with 15 robots.

For the purposes of our evaluation, α is set to 0.7 since after the sensitivity analysis we found that this value capture well the trade-off between

the traveled distance and the convergence time.

Figure 4.4 shows that whatever the used approach and when the number of robot is less than 60, the robot moving distance is proportional to the number of robots in the networks. Therefore, the average traveled distance of a robot is sensitive to robot numbers in all these algorithms. This can be explained by the fact that a robot is expected to have more neighbors when the number of robots increases. This property is no longer valid when the density of the robot is quite high (*i.e.*, specifically for a number of robots higher than 50). In this case, there will be a strong collision and the knowledge of the neighborhood will not be exact. From Figure 4.5 we can notice a false information about the neighborhood when the number of the robots is more than 50. The average number of neighbors for 60 and 70 robots is less than 2, which is completely false.

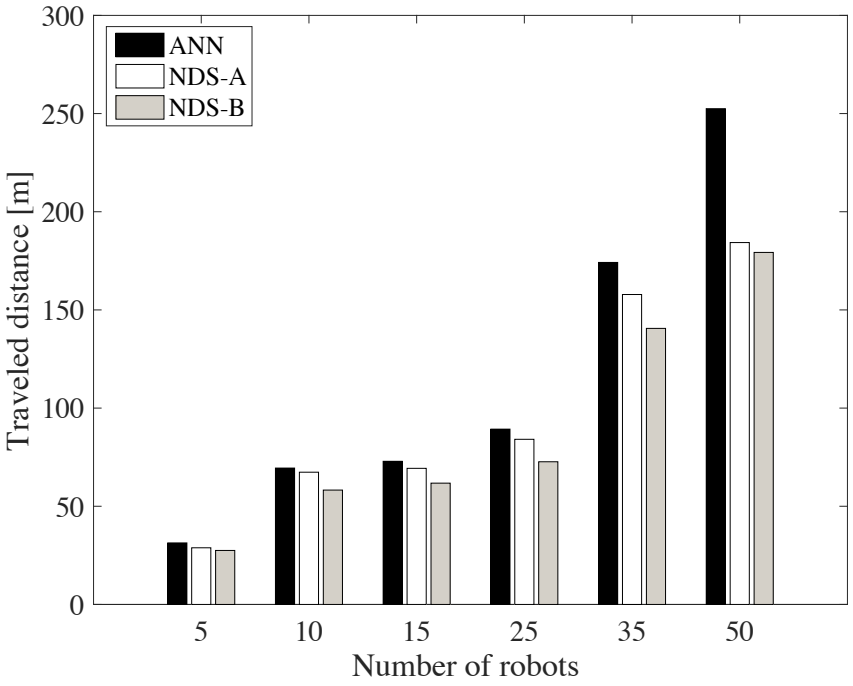


FIGURE 4.4: Traveled distance according the number of the robots, for different approaches

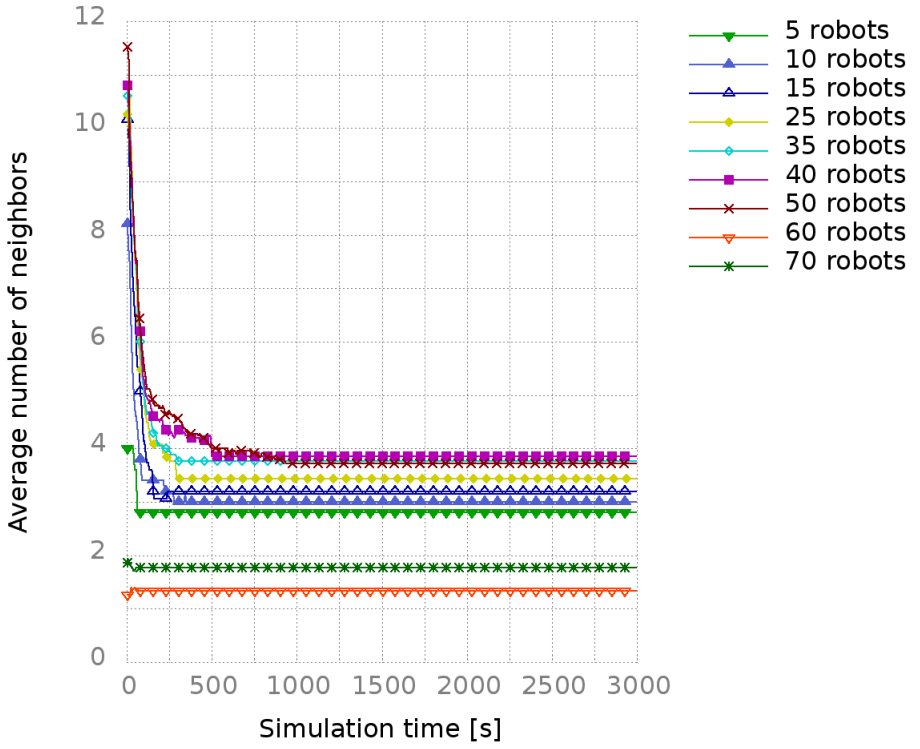


FIGURE 4.5: Average number of neighbors for different number of robots

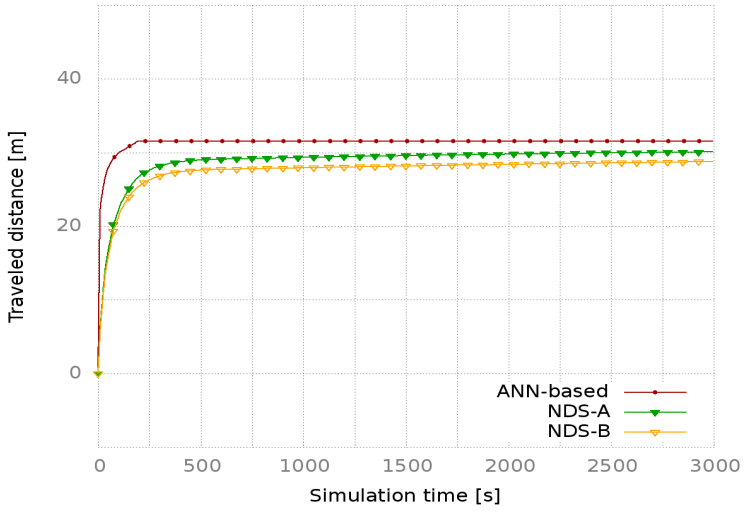
In the following, we present simulation results for different robot networks (*i.e.*, ranging from 5 to 70 robots). We aim to assess the effectiveness of our proposed NDS approaches w.r.t. ANN. Results are expressed in terms of (*i*) average traveled distance by a robot, (*ii*) convergence time to a desired distance D_{Th}, and (*iii*) power level, meaning the maintained QoS level.

From Figures [4.6 - 4.13](a) we notice that our algorithms decrease considerably the average distance traveled by a robot when we have a good number of robot (*i.e.*, less than 50). Our algorithms start to be inefficient when the number of robots is more than 50. As early mentioned, this is due to the expected high collisions in the network. Furthermore, we lose the linearity property we have noticed when the robot number is less than 50 (*i.e.*, the average traveled distance increases with the robot number). When we meet the good number of robot, NDS-B algorithm approach remains the most effective in terms of traveled distance, which validates our theoretical model described in Section 4.2. Finally, by considering the relationship between energy and traveled distance, we can say that our NDS approaches are energy efficient since it decreases the average distance traveled by a robot (hence, the energy consumed by a robot decreases as well).

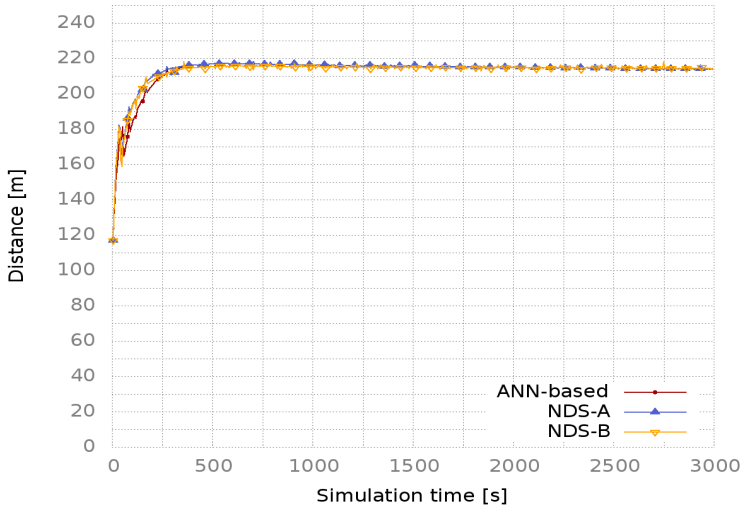
Figures [4.6 - 4.13](b) illustrate the convergence of our algorithms (*i.e.*, ANN, NDS-A and NDS-B) to the desired distance set to 212 m. We can see that the convergence time of our NDS approaches is more or less equal to the convergence time of the ANN-based when we have a good density of robot (i.e. less than 50 robots). Indeed, neighbor discovery is a key component in our approach since the robots use the knowledge of the robots

neighborhood to compute its new position. Therefore, a poor knowledge of the neighborhood will make our algorithm ineffective. This is due to the strong collision present on the network when robot density is quite high. This explains therefore the bad results that we obtained with 60 and 70 robots (see Figure 4.12 and Figure 4.13).

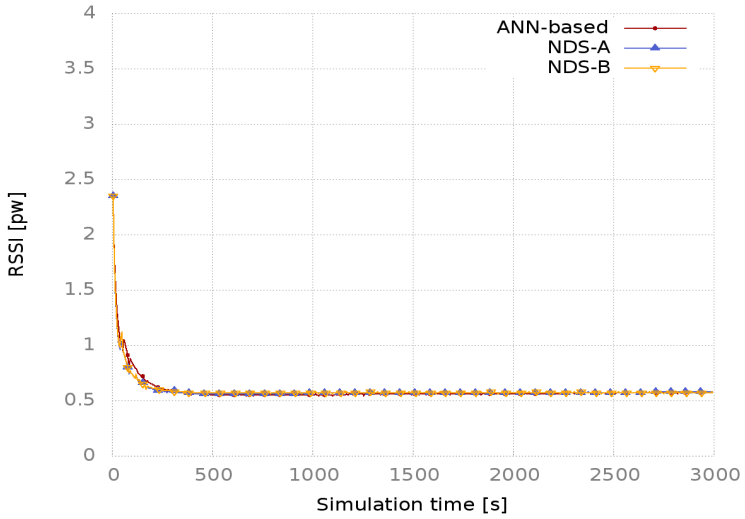
Finally, Figures [4.6 - 4.13](c) show also that our proposed approach can maintain the desired QoS level(RSSI) among neighboring robots. As in the previous chapter, the choice of the desired QoS level was made on the basis of the desired distance D_{th} (i.e. the RSSI measured at a distance D_{th}).



(a) Traveled distance

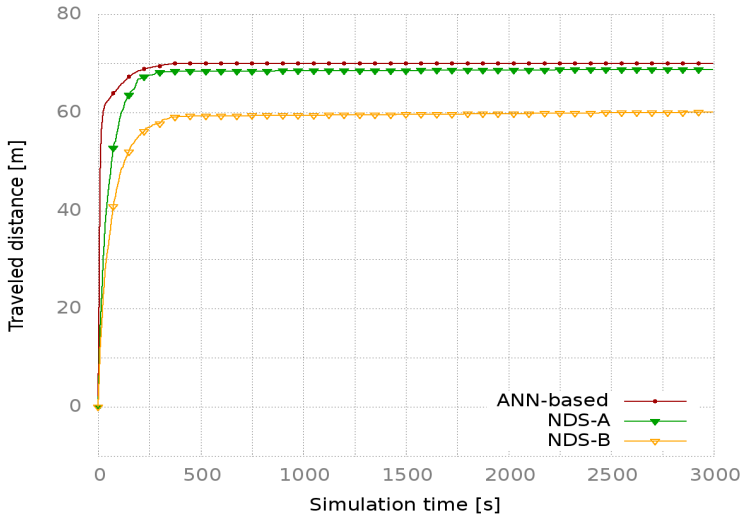


(b) Position

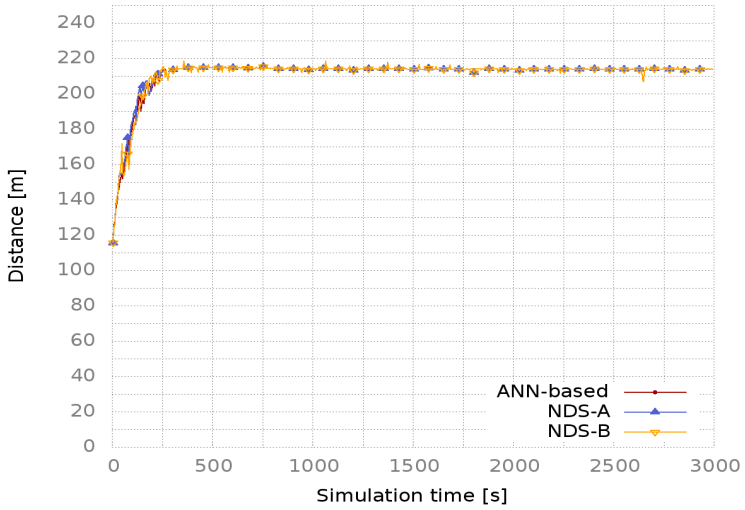


(c) RSSI

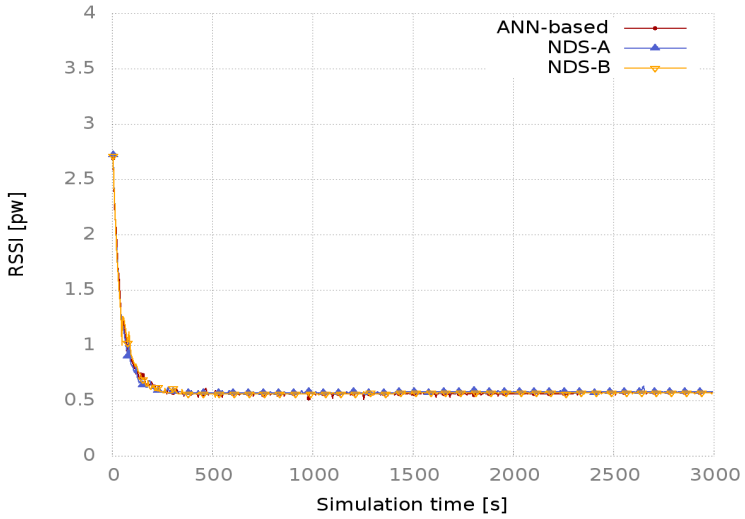
FIGURE 4.6: Simulation results obtained with 5 robots



(a) Traveled distance

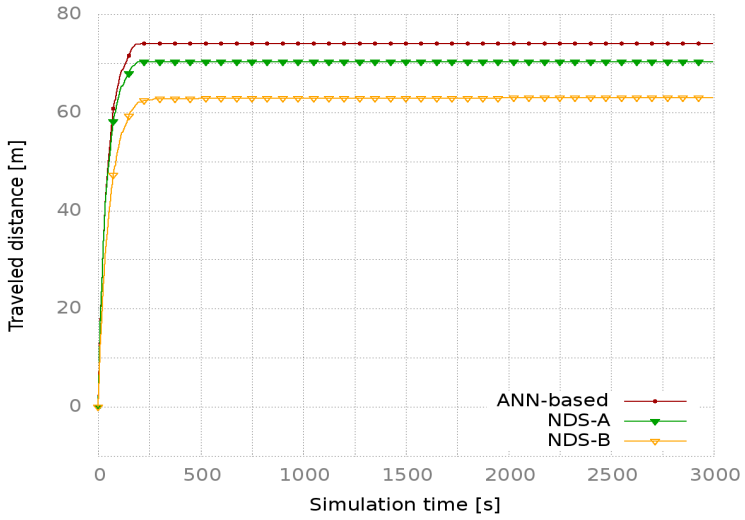


(b) Position

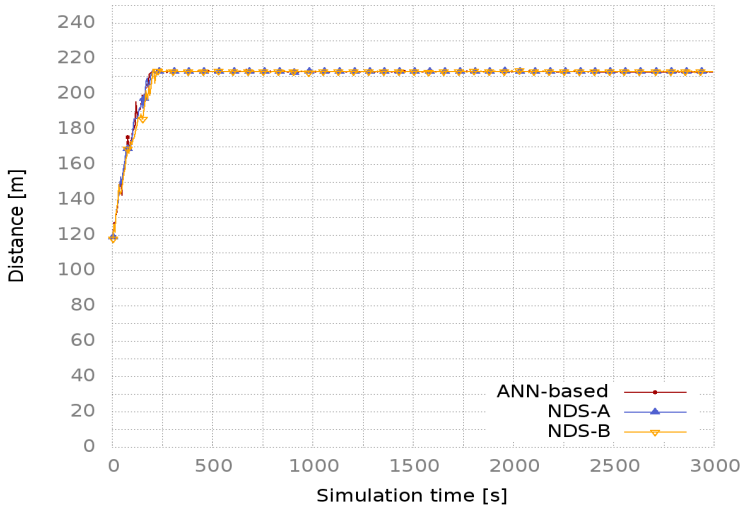


(c) RSSI

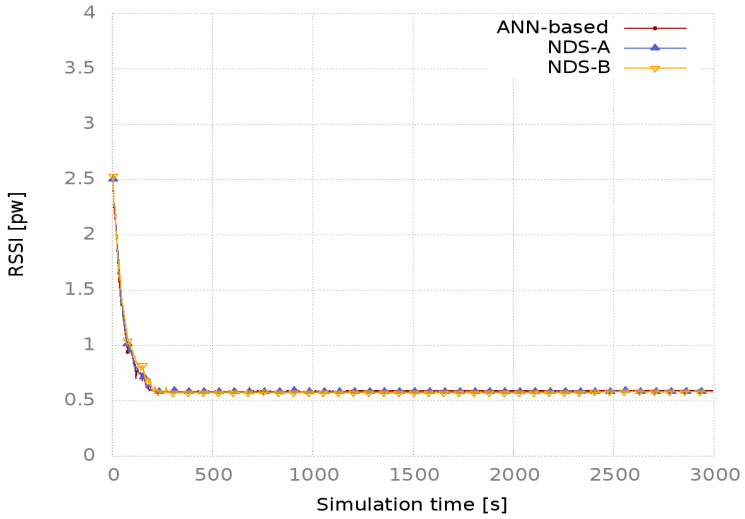
FIGURE 4.7: Simulation results obtained with 10 robots



(a) Traveled distance

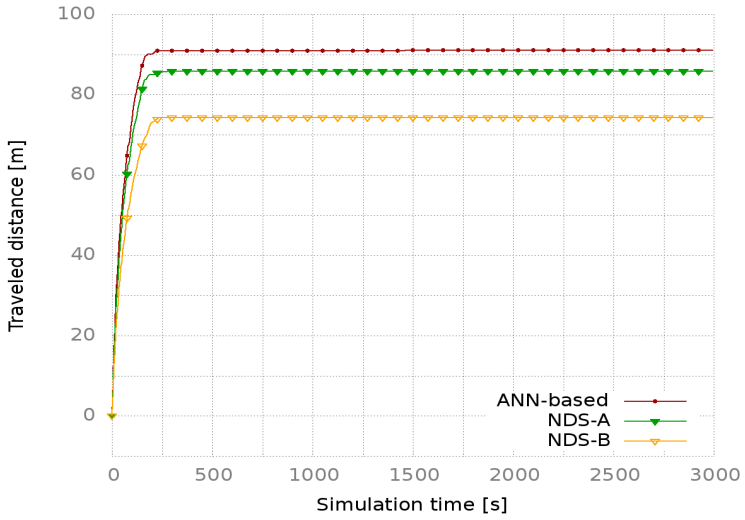


(b) Position

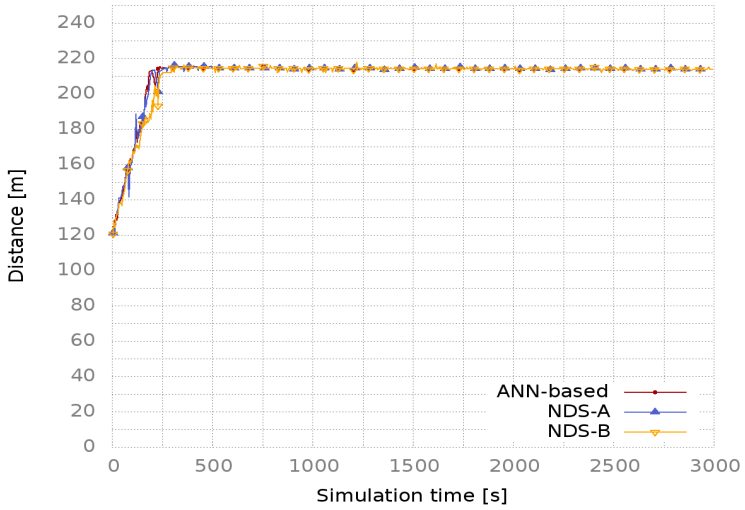


(c) RSSI

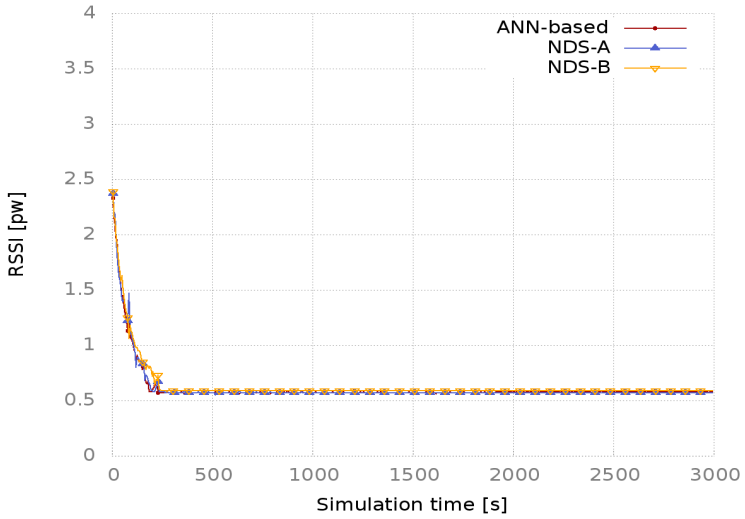
FIGURE 4.8: Simulation results obtained with 15 robots



(a) Traveled distance

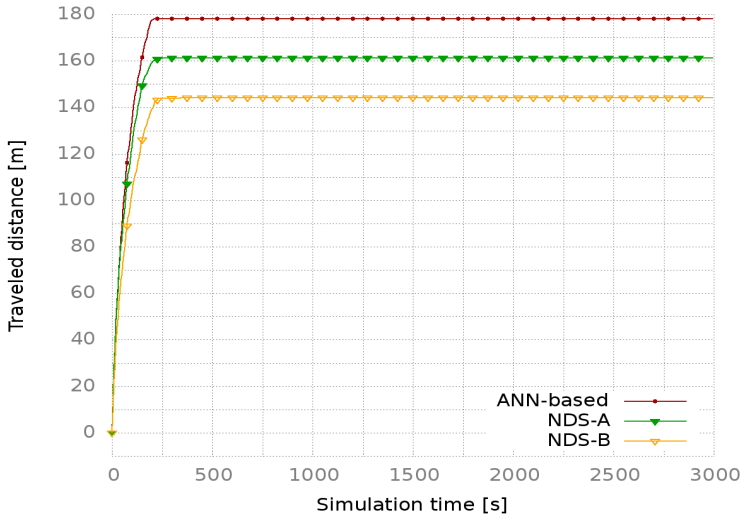


(b) Position

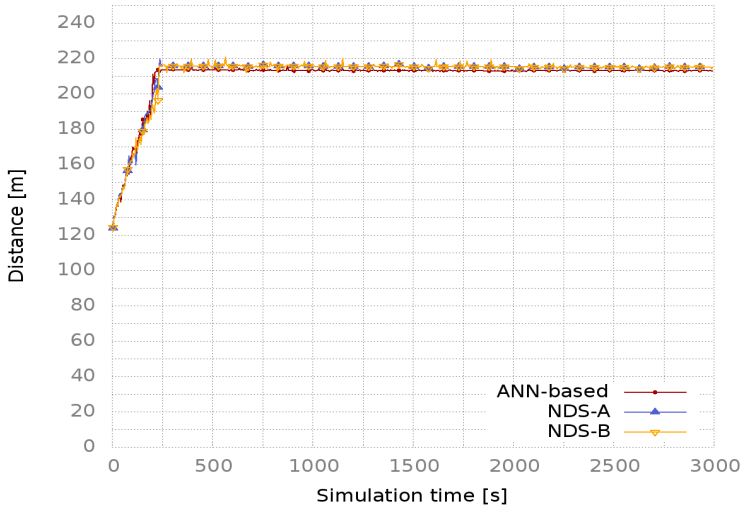


(c) RSSI

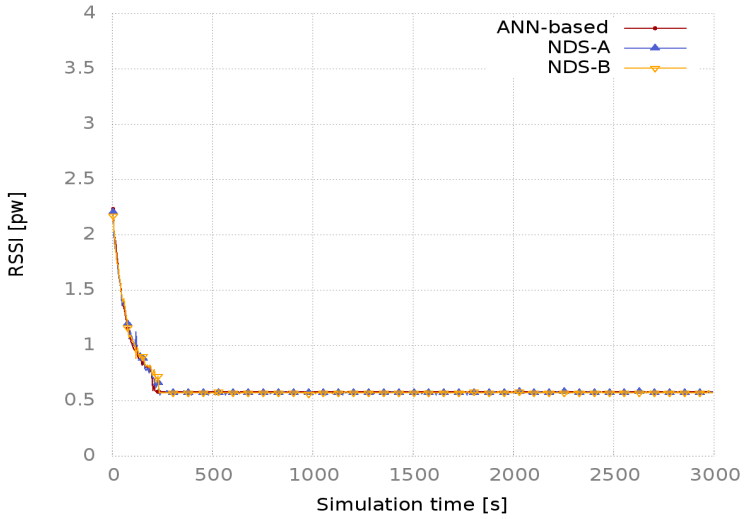
FIGURE 4.9: Simulation results obtained with 25 robots



(a) Traveled distance

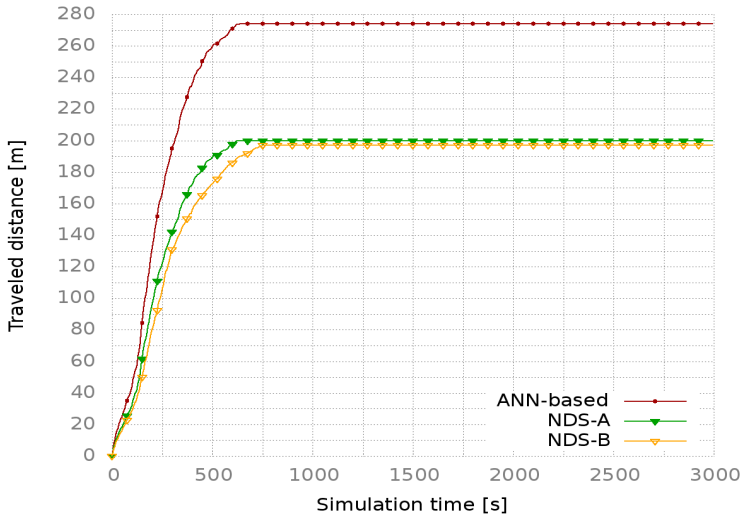


(b) Position

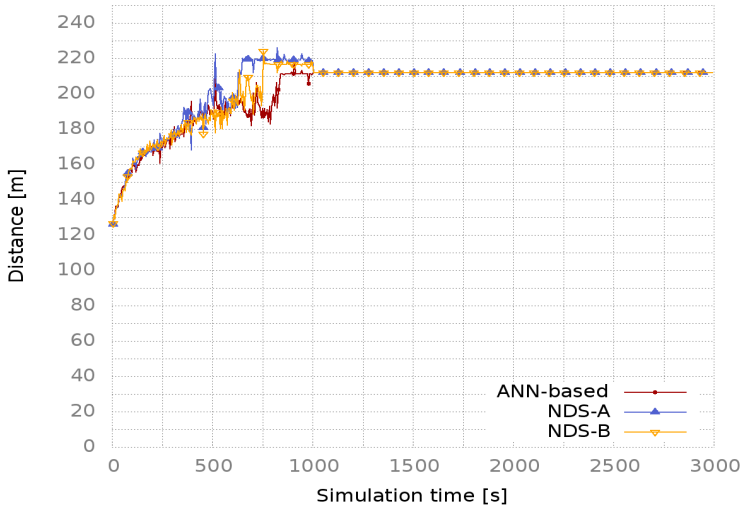


(c) RSSI

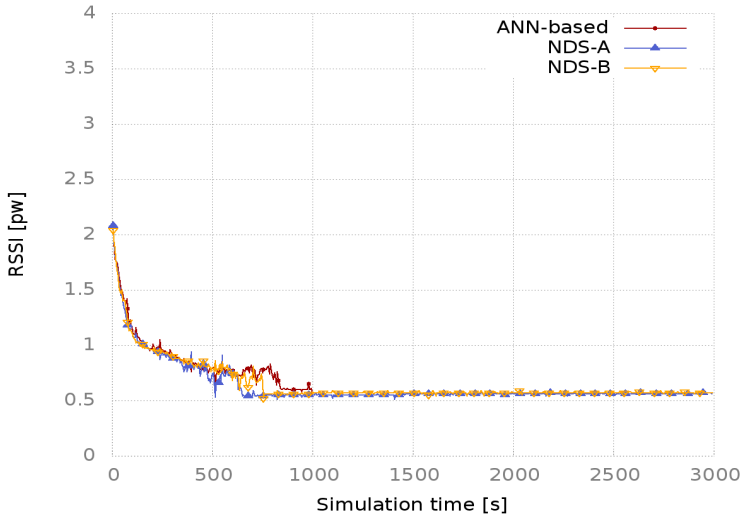
FIGURE 4.10: Simulation results obtained with 35 robots



(a) Traveled distance

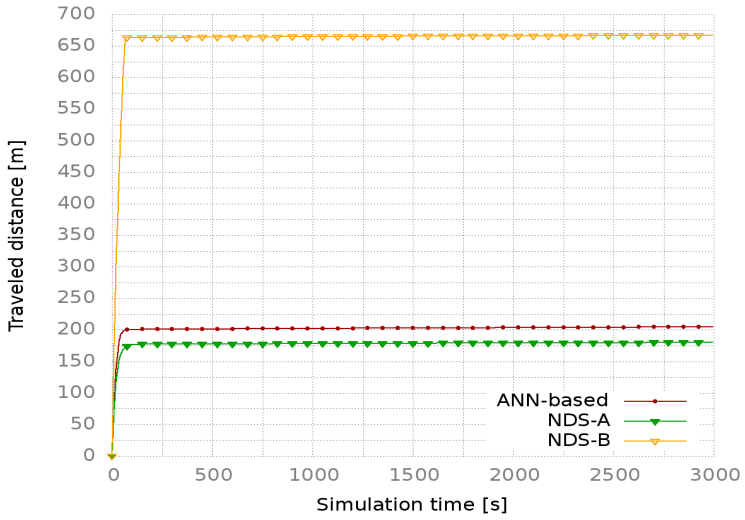


(b) Position

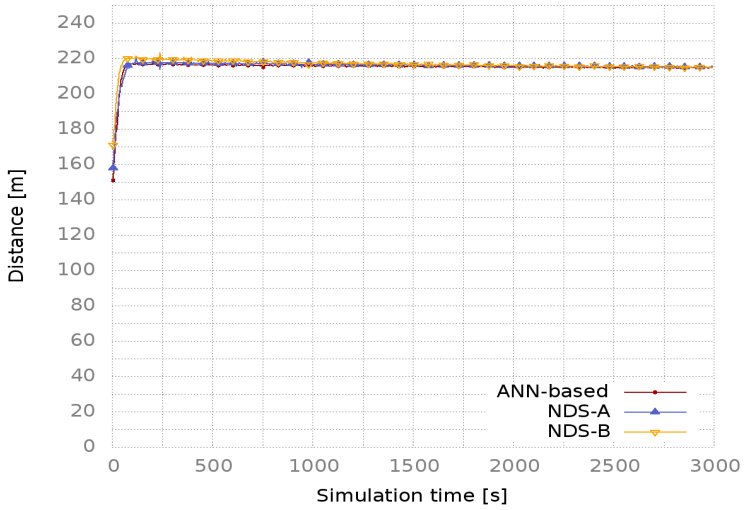


(c) RSSI

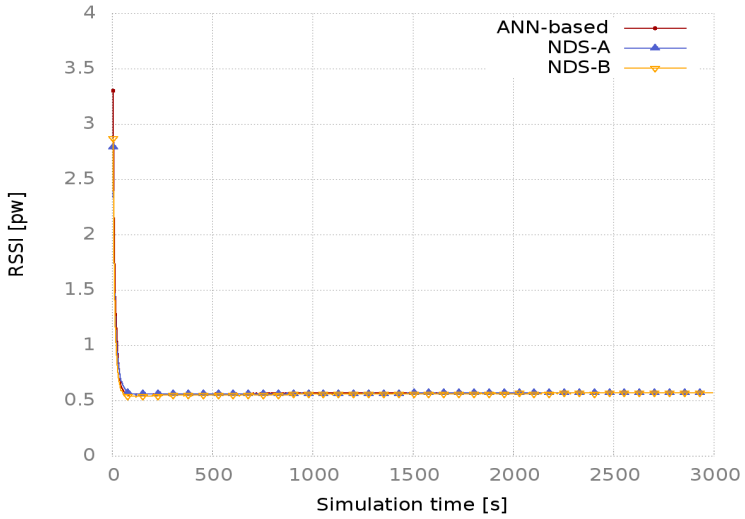
FIGURE 4.11: Simulation results obtained with 50 robots



(a) Traveled distance

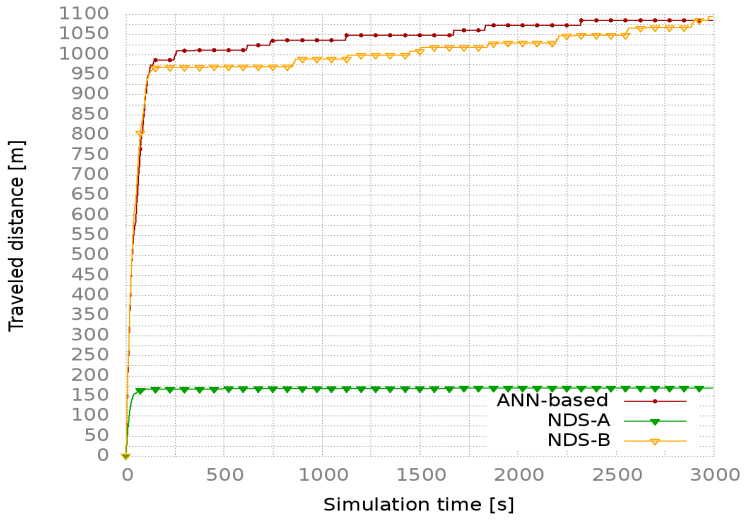


(b) Position

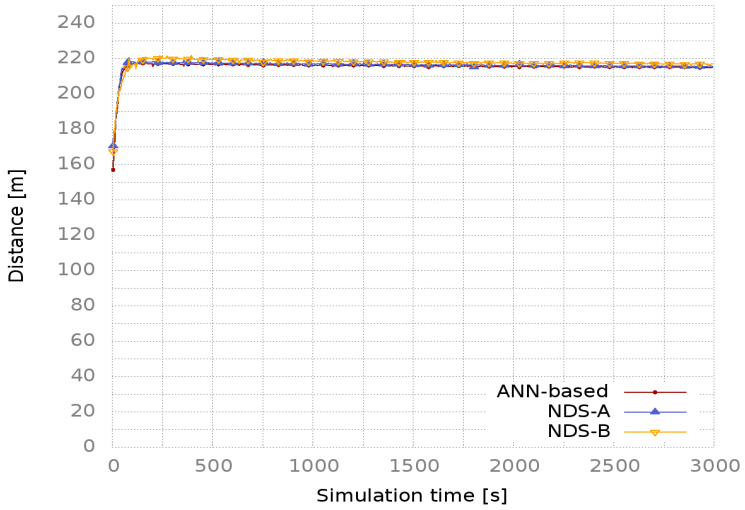


(c) RSSI

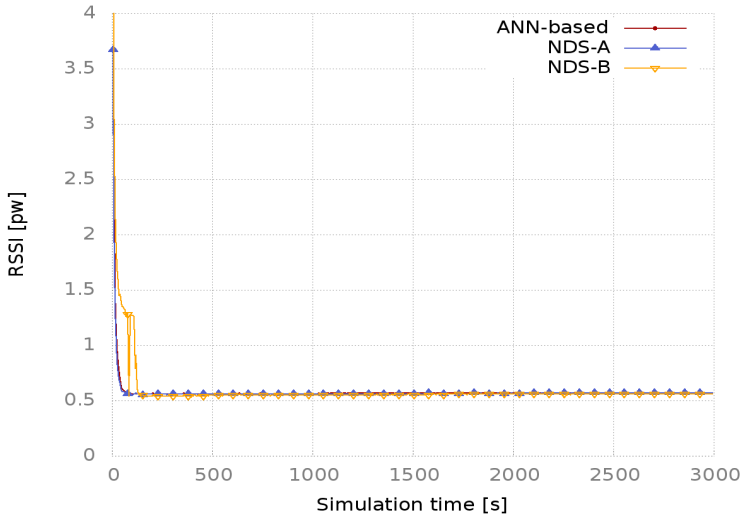
FIGURE 4.12: Simulation results obtained with 60 robots



(a) Traveled distance



(b) Position



(c) RSSI

FIGURE 4.13: Simulation results obtained with 70 robots

4.4 Summary

In this chapter, we have presented a NDS algorithm with the aim of efficiently maintain the global connectivity among multiple mobile robots to a desired quality-of-service level and distance. The proposed approach is based on neural network and dominating set strategies. Our contribution was to demonstrate that heterogeneous behaviour of robot (or mobile sensor) in IoHT concept can decrease the global traveled distance while achieving a minimum convergence time. Through a theoretical analysis and extensive simulations, we showed that the NDS approach outperforms the ANN-Based approach proposed in the previous chapter in term of traveled distance while maintaining similar convergence time.

Chapter 5

Bayesian Inference Approach For an Efficient Data Sharing among IoHT devices

In the two previous chapters, we have guaranteed the connectivity between IoHT mobile devices. Once the connectivity is guaranteed, another major challenge that should be solved is the huge amount of data generated and transmitted by the IoHT devices. Storing this huge amount of data locally will not be possible any more. Therefore, harnessing cloud computing capacity is needed but unfortunately this is not enough. However, it was observed that, with the increase of sensor density, data generated by IoHT devices tend to be highly redundant and correlated. Thus, uploading raw data to the cloud can become extremely inefficient due to the waste of memory and network overloading. To address this issue, either the IoHT devices should avoid the generation of useless data or the gateway device should be able to stop uploading of redundant and correlated data from some devices, to reduce consumption of network and cloud resources.

From the above considerations, in this chapter, we propose a Bayesian Inference Approach (BIA) for an efficient data sharing among IoHT devices. BIA is based on the Markov Random Fields model and Belief Propagation (BP) algorithm described respectively in the subsections 2.2.2.4.2 and 2.2.2.5.1 of Chapter 2. In the IoHT context, the belief of a device (e.g., a sensor node) is related to the physical quantity measured by the sensor device. BP allows to infer the measurements of other neighboring devices, especially in cases where the data are missing. In BP-based approaches, each node determines its belief by merging its local measurement with the beliefs of its neighboring nodes, and its beliefs obtained in the past run. A good correlation between data is important in such inference problems since it dictates the accuracy of data inference, and hence reduces the estimation error of the global information.

5.1 Network model

As depicted in Figure 5.1, in order to achieve our goal, we adopt a cloud-based architecture consisting of sensing nodes, smart gateways and data centers. Each entity in our architecture plays a different role w.r.t the functionalities, the computational and communication capabilities. Our IoHT network model may include multiple sub-networks associated with different applications. Each sub-networks is composed of IoHT devices connected to each others for data sharing, and a smart gateway that relays the

data flows to the cloud. The cloud in turn is responsible of inference, storage and all the cloud-based services.

In a given IoHT application, the sensor nodes periodically collect environmental data, such as temperature, humidity and illuminance, and forward them to the gateways using a multi-hop routing protocol. Then, the gateways collect the data and decide what has to be sent to the cloud. This decision depends on the fact that the gateway knows or not the priori probability of inference error of the used approach.

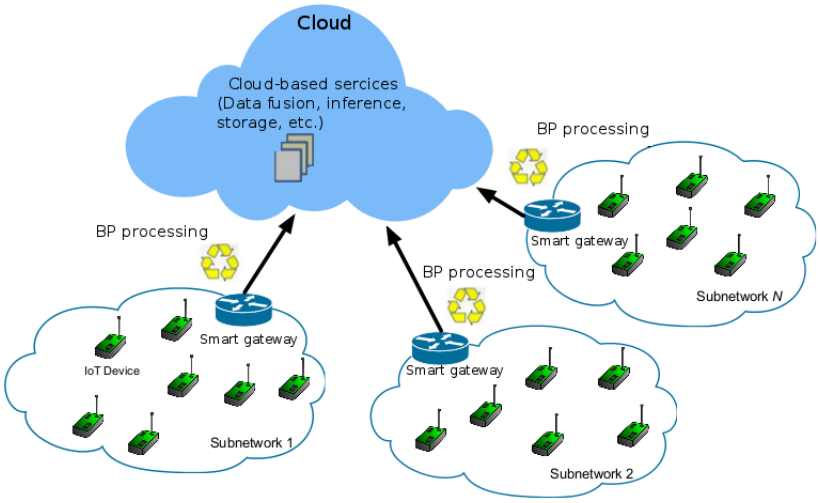


FIGURE 5.1: A cloud-based IoHT network model

5.2 Bayesian Inference Approach

As mentioned before, our main goal is to avoid sending useless data, while keeping an acceptable level of data content accuracy. As a starting point before any inference procedure, the design of a prediction model should be provided. We use a *pairwise Markov Random Fields* (MFR), modeled by means of Factor Graphs, for our BIA approach since its efficiency was already proven (see subsection 2.2.2.5 of Chapter 2 for the proof). It follows that our goal is to estimate the state X of the sensed environment starting from the sets of data collected by each IoHT device. By using the pairwise MRF model, according the Formula (2.24), the joint distribution $p(X)$ is given by the product of all the potential functions *i.e.*,

$$p(X) = \frac{1}{Z} \prod_{i} \theta_i(x_i) \prod_{i,j \in E} \psi_{ij}(x_i, x_j)$$

where Z is the normalization constant, $\theta_i(x_i)$ is the evidence function, E is the set of edges encoding the statistical dependencies between two hidden nodes i and j, and $\psi_{ij}(\cdot)$ represents the potential function.

As it is already stated, one of the main goals when dealing with graphical models is the computation of the marginal distribution (i.e the inference). Due to its efficiency, we use the BP algorithm to infer missing data. To recall, BP algorithm computes the marginal probability with a time which increases only linearly with the number of nodes in the MRF model.

Let $p(y_i)$ represents the marginal distribution (i.e the belief) of i-th node, and BP allows the computation of $p(y_i)$ at each node i by means of a message passing algorithm. Using the Formula (2.29), the message from the i-th to the j-th node related to the local information y_i is defined as:

$$m_{ij}(x_j) \leftarrow \sum_{x_i} \theta_i(x_i)\psi_{ij}(x_i, x_j) \prod_{u \in N(i), u \neq j} m_{ui}(x_i)$$

where $N(i)$ denotes the neighbors of node i and the incoming messages from previous iteration are represented by m_{ui}. It should be noted that the Formula (2.29) will be performed between all nodes in the model until the convergence or if a maximum number of iterations I_{max} will be reached. Thus, using the Formula (2.28), the belief at the i-th node is computed through all the incoming messages from the neighboring nodes and the local belief i.e

$$p(y_i) = belief(y_i) = k\theta_i(y_i) \prod_{j \in N(i)} m_{ji}(y_i)$$

5.3 Experimental results

To validate our approach, we conducted experiments on indoor and outdoor environments. We will first discuss the experimental setup and results obtained on indoor environments and we will then discuss the outdoor environments.

5.3.1 Indoor environments

Here, our approach was evaluated on the basis of real data collected from 54 sensors deployed in the Intel Berkeley Research laboratory [38]. Mica2Dot sensors with weather boards were used to collect temperature, humidity, light intensity and battery voltage, as well as the network connectivity information which makes possible to reconstruct the network topology. Figure 5.2 illustrates the map of the Intel Berkeley Research Lab, with the hexagon-shaped nodes indicating the Ids of the sensor nodes.

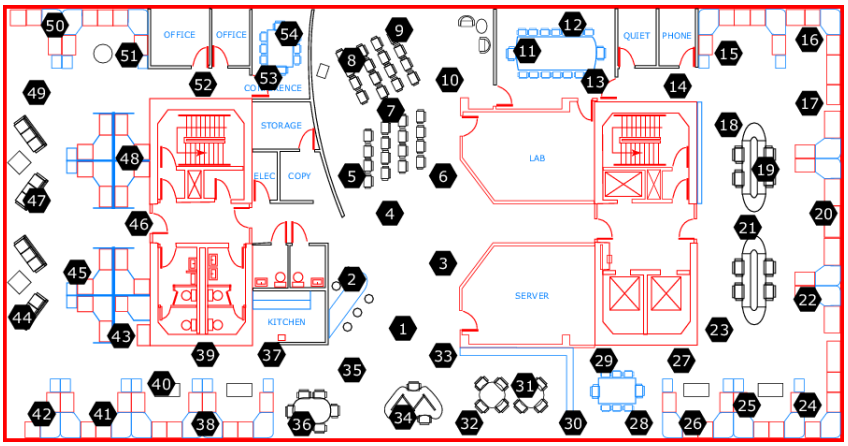


FIGURE 5.2: Wireless sensors installed in the Intel Berkeley Research Lab [38]

Sensors data was collected every 30 seconds and the collected data set consists of 38 days of readings. However, we will focus only on the first three hours of readings to validate our approach. Figure 5.3 illustrates the scatterplot matrix of our data for the first three hours of reading. This scatterplot matrix helps us visualize the relationship between data. For each panel in the figure, the variable on the vertical axis is given by the variable name in that row, and the variable on the horizontal axis is given by the variable name in that column. For example, the graph of temperature against humidity is shown on the top row, second from the left. It can be seen here that there is a strong correlation between temperature and humidity data. The temperature values can therefore be easily inferred from the humidity values and vice versa. For this reason, we decided to concentrate our studies only on the temperature and humidity data. The temperature is in degrees Celsius, whilst the humidity is a value ranging from 0-100%.

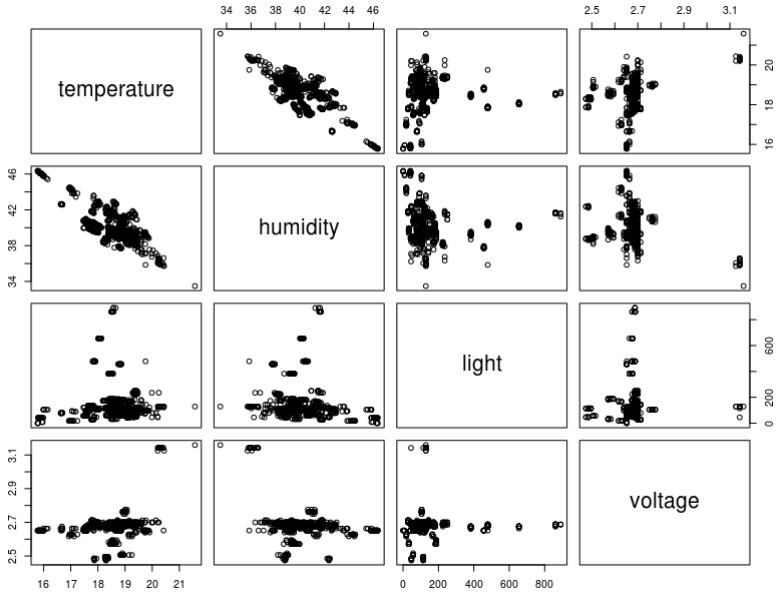


FIGURE 5.3: Correlation between collected data during the first three hours of readings.

The assessment of our BIA approach will be carried out on the basis of (*i*) the number of transmitted data, (*ii*) the average value of the inference error (*i.e.*, ER), and (*iii*) the average value of the distortion level (*i.e.*, MSE).

The number of transmitted data represents the total number of data transmission performed by all the sensors during the readings. The inference error is an important metric for any inference procedure. The goal is to have an errorless inference approach, *i.e.* an approach that is able to estimate the true value of data during all the inference procedures. However, this is almost never the case but we want that this error is as low as possible. In addition to the inference error, computing the distortion level is also important. This allows to determine the difference between the real and the estimated value, and can be expressed using the Mean Squared Error

(MSE) metric, defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y_i} - y_i)^2, \tag{5.1}$$

where $\hat{y_i}$ and y_i are respectively the predicted and true value during the n-th reading.

All of our assessments are based on three different scenarios (*i.e.*, s_1, s_2, and s_3) :

- ***scenario*** s_1 : The gateway sends to the cloud all the temperature and humidity data it receives. This means that the cloud does not perform any inference.
- ***scenario*** s_2 : The gateway sends only the temperature data to the cloud, and the cloud in turn infers the corresponding humidity data by using the BP algorithm.
- ***scenario*** s_3: Here, we consider that the gateways are "smart" devices, meaning that before sending their data to the cloud, they first compute the probability $\Pr(e|T,h)$ of making an inference error e on the cloud given the temperature data T, and the humidity data h. If there is a strong probability that the error magnitude *i.e.*, $|e|$, exceeds a predefined threshold *i.e.*, $|e|_{Max}$, the gateway sends both humidity and temperature data to the cloud, else the gateway sends only the temperature data, and the humidity value will be inferred in the cloud using the BP algorithm. This can be expressed mathematically as the inference error probability higher than a maximum allowed value $|e|_{Max}$, and conditioned to the temperature and humidity measurements *i.e.*, T and h, is lower or at least equal to a given threshold P_e^{Max}, that is:
$$Pr\{\mid e \mid > \mid e \mid_{Max} \mid T, h\} \leq P_e^{Max} \tag{5.2}$$
where the computation of $\Pr(e|T,h)$ is done by means of the BP algorithm. It should be noted that this computation requires the knowledge of the a priori probability of inference error *i.e.*, $\Pr(e)$. Also, the value of the threshold $|e|_{Max}$ strictly depends on the application context. In our case, we set this value equal to 1. A similar consideration can be applied to the probability threshold P_e^{Max}, which has been set to 0.5.

Our approach has been implemented in C++, and the assessments have been performed with respect to the ground truth collected from the 54 sensors deployed in the Intel Berkeley Research laboratory. Table 5.1 illustrates the obtained results during eighteen hours of readings, for the different simulated scenarios. We can notice that our Bayesian inference approach drastically reduces the number of transmitted data, while maintaining an acceptable level of prediction accuracy and information quality.

Not surprisingly, the amount of transmitted data is huge when using scenario s_1 since we do not use an inference approach here. This amount of data is reduced by 50% with the second scenario (i.e. s_2) but unfortunately with 26% of inference error. However, despite this inference error, we can see that there is still a good quality of information since we have a fairly low

value of MSE. This means that there was not a large difference between the real and the predicted values.

We decrease considerably the estimation error by using the scenario s_3. Indeed, the gateways are smarter in this case. By computing the a posteriori probability of inference error, they will be able to estimate the right moment and the data type to send to the cloud. However, this increases the number of transmitted data, as compared to scenario s_2. This is due to the fact that in s_2 the gateways send only the temperature data without worrying of the risk of inference error in the cloud.

Scenario	*#Transmitted data*	MSE	ER
s_1	20346	-	-
s_2	10173	0.04	0.26
s_3	12496	0.02	0.037

TABLE 5.1: Results obtained during the first three hours of readings for different scenarios.

5.3.2 Outdoor environments

Here, our approach was evaluated on the basis of real data collected from the PEACH project [58], whose aim is to dramatically increase the predictability of frost events [1] in peach orchards localized in the Mendoza region (Argentina) by means of dense monitoring using low-power wireless mesh networking technology. To predict the frost events, twenty one nodes have been used (1 manager, 3 relays and 17 motes). All the nodes are equipped with an internal temperature sensor and report value every 30 seconds. Fives motes are equipped with four external SHT31 sensors. Figure 5.4 illustrates the wireless sensor network deployed on the peach orchards in the Mendoza region (Argentina).

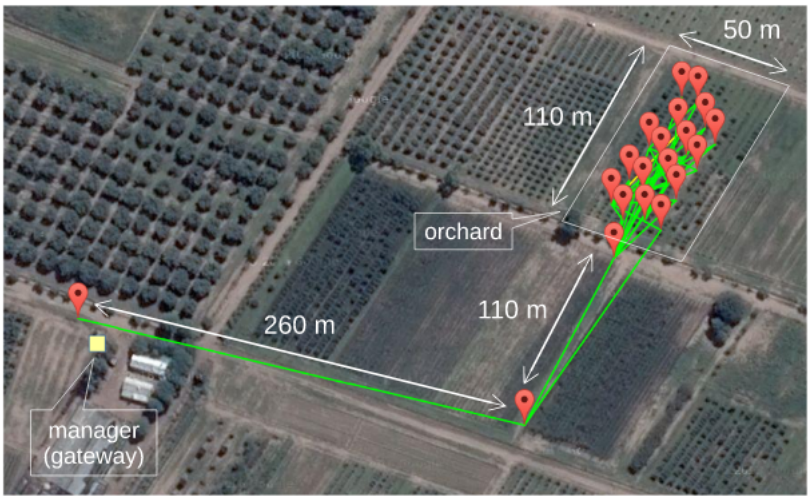


FIGURE 5.4: The PEACH network [11].

To validate our approach, we focus only on the data collected between 16 and 17 October 2016. The collection started at 10pm on October 16th and

[1] A frost event occurs when ice forms inside the plant tissue and injures the plant cells plant.

ended at 4pm on October 17th (i.e. eighteen hours). Figure 5.5 illustrates the relationship between data during the sensors reading. We can notice that there is a good correlation between temperature and humidity data, so that we can easily infer the humidity from temperature data, and vice versa. Here, we infer humidity from temperature. The temperature is in degrees Celsius, whilst the humidity is a value ranging from 0 to 100%.

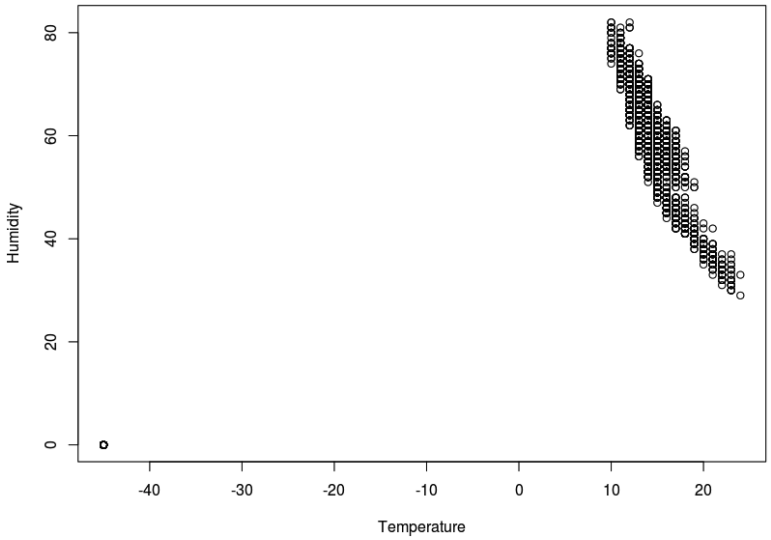


FIGURE 5.5: Relationship between humidity and temperature data.

As in the indoor environments, we assess our approach w.r.t. (*i*) the number of transmitted data, (*ii*) the average value of the distortion level (*i.e.*, MSE), and (*iii*) the average value of the estimation error (*i.e.*, ER). All of the assessments are based on the three different scenarios (*i.e.*, s_1, s_2, and s_3) already defined in the indoor environments.

Table 5.2 illustrates the obtained results during eighteen hours of readings, for the different simulated scenarios. As in the indoor environnments, the proposed Bayesian inference approach drastically reduces the number of transmitted data, while maintaining an acceptable level of prediction accuracy and information quality. The obtained results show us the same observation as with the indoor environments. We get a huge amount of with scenario s_1. This amount is reduced by 50% with scenario s_2 but with an inference error of 30% . Using Scenario s_3 (i.e using a smart gateway), the inference error is reduced to 0.6%. The information quality is acceptable in both scenario s_2 and s_3 since we get an acceptable value of MSE.

Scenario	*#Transmitted data*	MSE	ER
s_1	8408	-	-
s_2	4204	0.62	0.295
s_3	4260	0.022	0.0066

TABLE 5.2: Results obtained during eighteen hours of readings for different scenarios.

5.3.3 Raw data filtering in the sensing nodes

Although the used Intel Berkeley and the PEACH data-sets allowed to simulate the efficiency of our BIA approach, the lack of access to the deployed sensors did not allow us to implement the model directly on the sensors. Therefore, it is the gateway which does all the job so far. This means that, the proposed approach has not yet a positive impact on the devices since that is the gateway which does all the data filter. So, we want to study the feasibility of making the data filtering directly on the sensing nodes. To this end, we have created a prototype of a sensor node and IoHT gateway. We have used *Arduino* as sensor node and *Raspberry Pi* as gateway. We have used also an inexpensive *DHT22* sensor which captures the ambient humidity and temperature and sends (or not according the used scenario) it to the gateway. To sum up, we use the following for the prototyping:
Sensor Node

- Arduino Uno
- DHT22 humidity/temperature sensor
- XBee series 1

Gateway

- Raspberry Pi
- XBee series 1

The *XBee* modules have been used to create a wireless link between sensor and gateway. Figure 5.7 shows our created prototype.

Before implementing the MRF model in the sensor node and the gateway, we first carried out measurements in different place in our labs during 6 days. We did about six hours and fifteen minutes of data collection per day. Figure 5.6 illustrates the relationship between temperature and humidity data during the six days of readings. The correlation between humidity and temperature data is not so bad here (i.e cor = -0.7650688). We can therefore infer the humidity from temperature data, and vice versa. Here, we infer temperature from humidity.

We assess our approach w.r.t. (*i*) the number of transmitted data, (*ii*) the energy consumption (*i.e.*, EC), (*iii*) the average value of the distortion level (*i.e.*, MSE), and (*iv*) the average value of the estimation error (*i.e.*, ER). In our energy consumption evaluations, we assume that the energy cost for sending each temperature and humidity value is 3mW, while the sensor is powered by 3V. This energy cost has been obtained on the DHT22 used mote. Furthermore, all of our assessments are based on three different scenarios below :

- ***scenario*** s_1 : The temperature and humidity data are sent to the gateway. This means that the gateway does not perform any inference.
- ***scenario*** s_2 : Only humidity values are sent to the gateway. The corresponding temperature values will be inferred in the gateway using the BP algorithm.

- ***scenario*** s_3: Here, we consider that the sensing nodes are "smart". So, they send both temperature and humidity data only if the probability of inference error exceeding the max allowed error is greater than a given threshold (i.e, $Pr\{\mid e \mid > \mid e \mid_{Max} \mid T, h\} > P_e^{Max}$). Else, they send only the humidity values to the gateway and the corresponding temperature values will be inferred in the gateway using the BP algorithm. The thresholds $|e|_{Max}$ and P_e^{Max} have been set to 1 and 0.5 respectively.

Table 5.3 illustrates the obtained results during the six days of readings, for the three defined scenarios. Again, we can notice that our Bayesian inference approach drastically reduces the number of transmitted data and the energy consumption, while maintaining an acceptable level of prediction accuracy and information quality.

We have got an big amount of transmitted data and high energy consumption with the scenario s_1, which is quite logical since there is no filter here. With the scenario s_2, the amount of transmitted data as well as the energy consumption are reduced by 50% but unfortunately with an inference error of 40% and an information quality which is not very bad. We have reduced the inference error with the scenario s_3 while maintaining a good quality of information. However, the devices consume and transmit more compared to scenario s_2 but we think that it is a good trade-off since the results obtained with s_3 are much better than with s_1.

Scenario	*#Transmitted data*	EC [kJ]	MSE	ER
s_1	4490	1816.834	-	-
s_2	2245	908.417	3.59	0.39
s_3	2258	913.677	0.1403	0.1603

TABLE 5.3: Results obtained during six days of readings for different scenarios and using the created prototype.

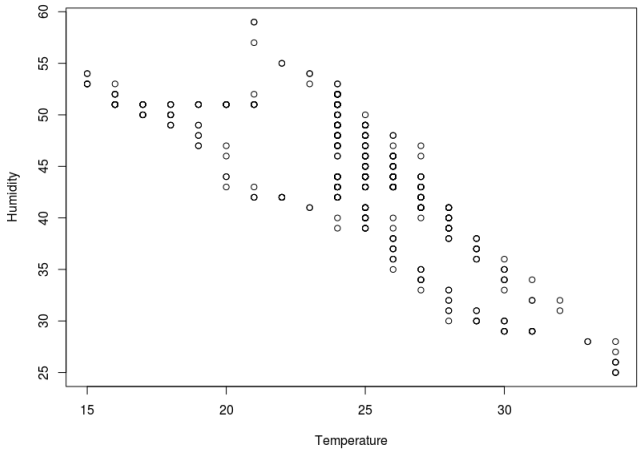


FIGURE 5.6: Relationship between humidity and temperature data during the six days of readings.

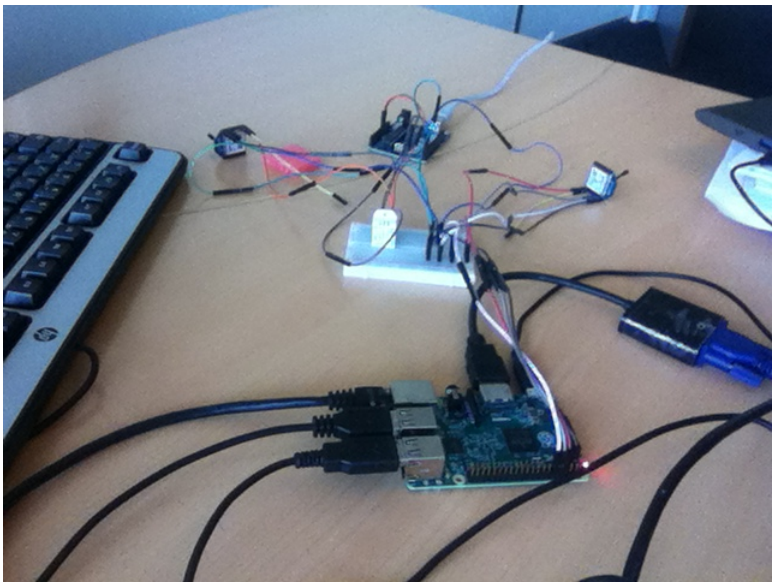


FIGURE 5.7: Prototype of a sensor node and IoHT gateway

5.3.4 Experimentation on FIT IoT-LAB platform

In order to validate the scalability of our BIA approach and filter the raw data directly in the sensing nodes, we ran experimentation on FIT IoT-LAB platform [2] which is a very large scale infrastructure facility suitable for testing small wireless sensor devices and heterogeneous communicating objects over large scale. Ten nodes from Lille site and ten nodes from Grenoble site were used for the data collection. Nodes were of the M3 type, which are equipped with an 32-bit ARM Cortex-M3 MCU, 64 kB of RAM, 256 kB of ROM, an IEEE 802.15.4 2.4 GHz radio transceiver and four different sensors (light, accelerometer, gyroscope, pressure & temperature). Data collected from all the M3 nodes has been used to build the BIA model. Each data collection has been performed every 15 minutes and the collected data consists of 2.5 days of readings.

During the 2.5 days of reading, we noticed that there is a good correlation between pressure and temperature data (it is about -0.7720841). Hence, we can infer the temperature data from pressure data and vice versa. Here, we decided to infer temperature from pressure. The temperature is in degrees Celsius, whilst the pressure is in mbar.

The assessments are always based on (*i*) the number of transmitted data, (*ii*) average value of the estimation error (ER), (*iii*) average value of the distortion level (MSE), and (*iv*) the energy consumption (EC). In the energy consumption evaluations, we assume that the energy cost for sending each temperature and pressure value is 14 mW.

We performed experimentation using scenarios s_1, s_2, and s_3. In scenario s_1, the M3 node sends to the gateway all the temperature and pressure data it receives. This means that the gateway does not perform any inference (*i.e.*, no inference). In the second scenario s_2, the M3 nodes sends only the pressure data to the gateway, and the gateway in turn infers the corresponding temperature data by using the BP algorithm. Finally, in the scenario s_3, we consider that the M3 nodes are "smart" devices, meaning

that before sending their data to the gateway, they first compute the probability $\Pr(e|T,P)$ of making an inference error e on the gateway given the temperature data T, and the pressure data P. If there is a strong probability that the error magnitude *i.e.*, $|e|$, exceeds a predefined threshold *i.e.*, $|e|_{Max}$, the M3 node sends both pressure and temperature data to the gateway, else the M3 node sends only the pressure data, and the temperature value will be inferred in the gateway using the BP algorithm. This can be expressed mathematically as the inference error probability higher than a maximum allowed value $|e|_{Max}$, and conditioned to the temperature and pressure measurements *i.e.*, T and h, is lower or at least equal to a given threshold P_e^{Max}, that is:

$$\Pr\{|e| > |e|_{Max}|\, T,P\} \leq P_e^{Max}, \tag{5.3}$$

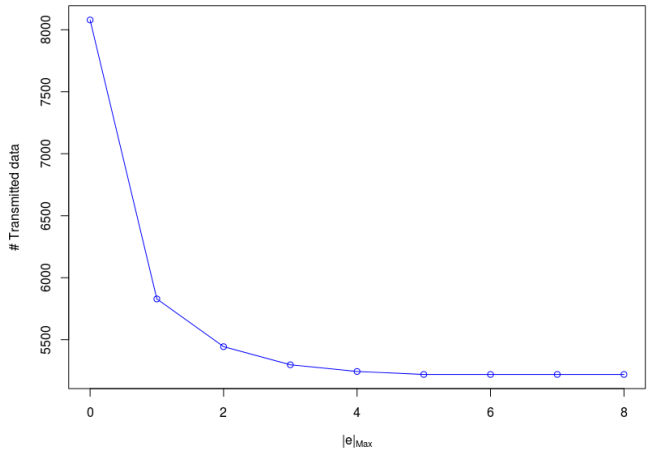
where the computation of $\Pr(e|T,P)$ is done by means of the BP algorithm. As we already said before, the value of the threshold $|e|_{Max}$ strictly depends on the application context. Here, we set this value equal to 1, but later we will provide additional results and comments that we did not provide before on how the choice of this threshold value may influence our results.

Table 5.4 illustrates the obtained results during 2.5 days of readings, for different experimented scenarios. Again, one can notice that our Bayesian inference approach drastically reduces the number of transmitted data and the energy consumption, while maintaining an acceptable level of prediction accuracy and information quality. One can notice also that we decrease considerably the estimation error by using the scenario s_3. Indeed, the M3 nodes are smarter in this case *i.e.*, by computing the a posteriori probability of the inference error, the M3 nodes will be able to estimate the right moment and the data type to send in the gateway. However, this increases the number of transmitted data and hence the energy consumption, as compared to scenario s_2. This is due to the fact that in s_2, the M3 node send only the pressure data without worrying of the risk of inference error in the gateway. It is important to say that we have a good quality of information in the scenario s_3 despite the fact that we have an inference error of 43%. This is due to the fact that we allow only a maximum error of one unit (i.e $|e|_{Max} = 1$)

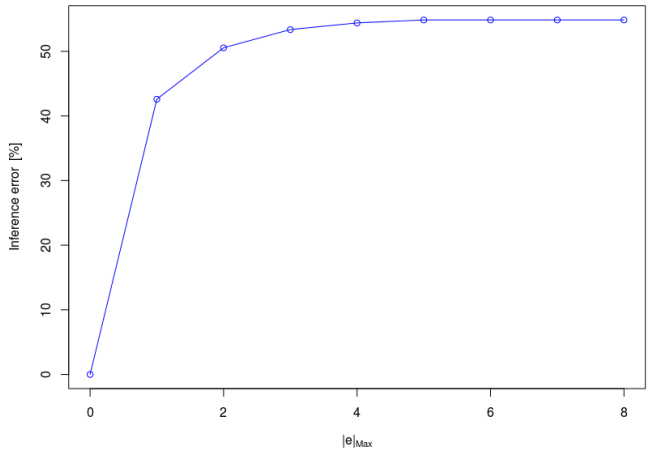
Scenario	*#Transmitted data*	EC (kJ)	MSE	ER
s1	10440	1716.64	-	-
s2	5220	858.32	1.43	0.55
s3	5829	958.46	0.43	0.43

TABLE 5.4: Results obtained during the two days and half of readings.

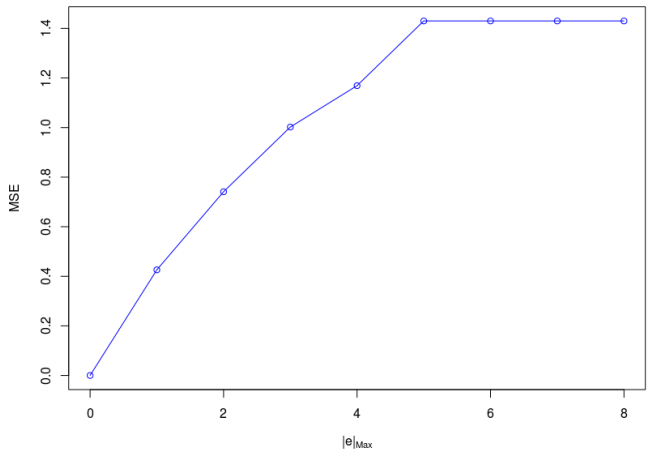
As we stated before, the value of the threshold $|e|_{Max}$ strictly depends on the application context. Its choice has a non-negligible impact on the final results. From Figure 5.8, for example, we can say that the more we use a higher threshold, the less we send data but also the more we get an inference error and the more we lose in information quality. In our experimentation the maximum value of the error that we can have is 5. This means that we always get the same results for $|e|_{Max} \geq 5$.



(a)



(b)



(c)

FIGURE 5.8: Variation of (a) the transmitted data, (b) the estimation error and (c) MSE according the value of the threshold $|e|_{Max}$.

5.4 Summary

In this chapter, we presented a Bayesian Inference Approach (BIA) with the aim of avoiding useless data transmission. The strong correlation between data was taken into account for this study. The Belief Propagation algorithm coupled with Markov Random Field was used to infer the missing data. The proposed approach was applied to the indoor and outdoor environments. It was also validated on simple prototype and FIT IoT-LAB in order to filter the raw data directly in the sensing device. Through extensive simulations and experimentation, we have showed that our BIA approach reduces considerably the number of transmitted data and the energy consumption, while keeping an acceptable level of estimation error and information quality. We have also shown that the use of smart gateway and sensor decreases significantly the inference error.

Chapter 6

Conclusion And Perspective

6.1 Summary of contribution

In this thesis, we have introduced the concept of the Internet of Heterogeneous Things (IoHT), which is a combination of both Internet of Things (IoT) and Internet of Robotic Things (IoRT) technologies. Then, we have focused particularly on two major issues i.e., (*i*) global connectivity maintenance among IoHT mobile devices, and (*ii*) reduction of the huge amount of data generated and transmitted by IoHT devices.

6.1.1 Global connectivity maintenance among IoHT mobile devices

As it is already stated, the performance of an IoHT application relies on the efficient coordination among devices, which in turn strongly depends on reliable communication to support efficient data sharing among devices. Hence, in order to maintain the global connectivity in the context of deploying a group of mobile robots, we exploited the potential of controlled mobility and proposed three adaptive and efficient schemes : (*i*) IoT-based, (*ii*) trained neural network, and (*iii*) Neuro-Dominating Set.

We have assessed the performance of our proposed approaches in terms of algebraic connectivity, traveled distance, average distance between any pair of mobile devices, QoS level expressed in terms of RSSI, and convergence time. Through extensive simulations with the NS-2 network simulator, we have showed that all the proposed approaches converge to the desired distance and QoS level while maintaining the global connectivity. We have also showed that our IoT-based and neural network based approaches outperform the EVFA approach proposed in [34], in terms of traveled distance and convergence time.

With regard to the NDS approach, it has been proposed to demonstrate that using an heterogeneous behaviour among IoHT mobile device can decrease the global traveled distance while achieving a minimum convergence time. Through a theoretical analysis and extensive simulations, we showed that the proposed NDS approach outperforms the neural network based approach in term of traveled distance while maintaining similar convergence time.

6.1.2 Data reduction

As it is already said before, the IoHT is experiencing an exponential growth in the number of inter-connected devices. If the amount of data generated by these devices follows also this growth, then the IoHT alone will have a

difficulty for supporting all the wireless communications of the devices. To deal with this problem, in this thesis, we focused on the potential benefits of adopting prediction-based strategies to reduce the huge amount of data generated and transmitted by sensing devices. Precisely, we proposed a Bayesian Inference Approach (BIA) based on the Belief Propagation algorithm and Markov Random Field model for inferring the missing data. BIA was applied not only to the indoor and outdoor environments but also on a created prototype and on FIT IoT-LAB platform which allow to filter the raw data directly in the sensing device. The strong correlation between data was taken into account for the study. The obtained results based on the real data collected from sensors devices showed that our BIA approach reduces significantly the number of transmitted data and the energy consumption, while keeping an acceptable level of estimation error and information quality.

6.2 Perspectives

The problems we have dealt with in this thesis open us some interesting perspectives. In the following, we separate these perspectives into two categories: (*i*) perspectives for the global connectivity maintenance and (*ii*) perspectives for the Bayesian Inference Approach.

6.2.1 Global connectivity maintenance

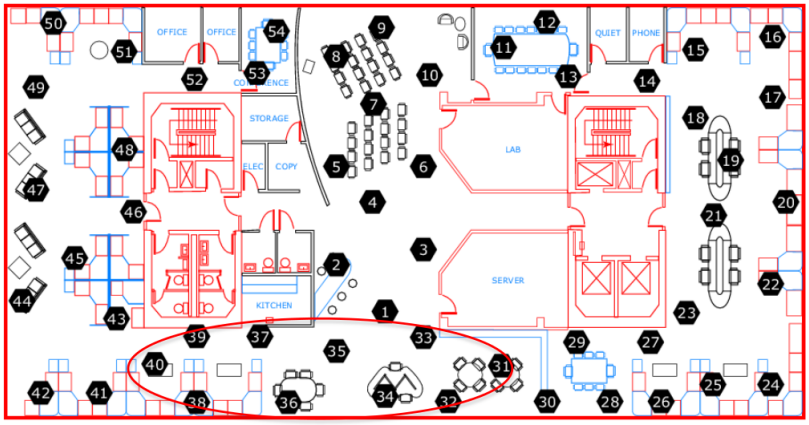
All the algorithms proposed in this thesis have shown good performances in simulation. However, even if realistic simulation parameters (MAC and PHY layers, propagation and error model, etc.) have been used, it would be interesting to implement the proposed algorithms on real robots in order to validate them in a real environment and condition.

6.2.2 Bayesian Inference Approach

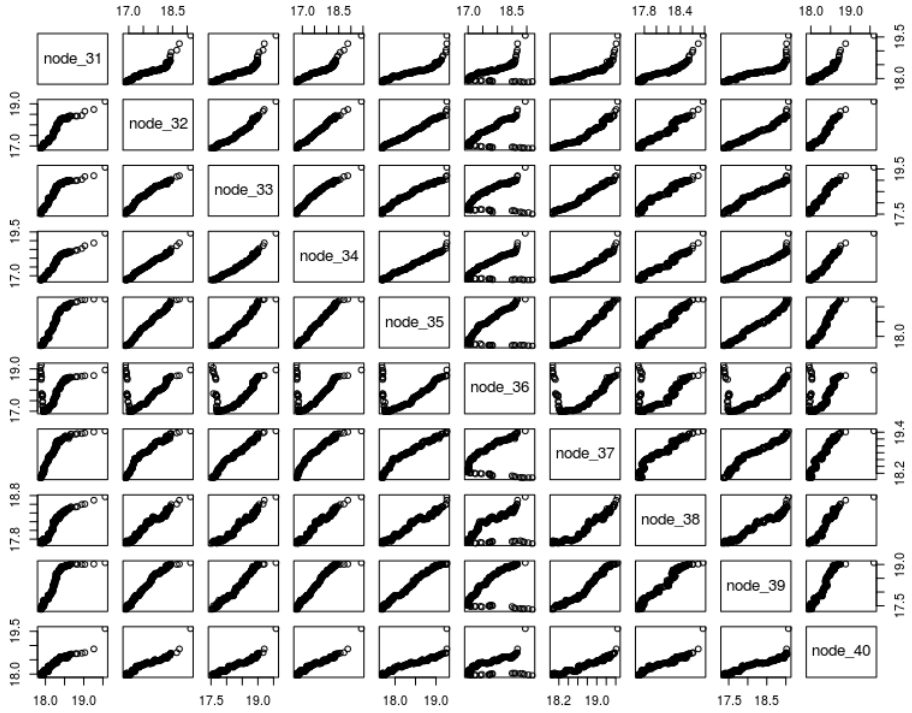
Our proposed BIA approach is based on a static MRF model so far since the used model is not updated once it is built. Therefore, our approach is only suitable for systems that do not vary much. When the environmental dynamics are high the used MRF model must be revised or re-constructed. In future work, one can go further by proposing an adaptive scheme which allows to update the used model for highly dynamic environments.

It may also be interesting to combine our BIA approach with a time series approach such as ARIMA. In this case, BIA will be applied to heterogeneous data while ARIMA to homogeneous data.

As you have noticed, in this thesis, we have focused particularly on the cross-modal correlation but we can also exploit the spatial correlation between nodes in future works. Using data from Intel Berkeley Research Lab [38], for example, I found that there is a strong correlation between the nodes that are close (see Figure 6.1). In this case, It is possible to allow certain nodes go to sleep mode without making any measurement. Then, our BIA approach can be used to infer the data values from the inactive nodes.



(a)



(b)

FIGURE 6.1: Correlation between the nodes that are close (nodes in the red circle in (a)) using data from Intel Berkeley Lab.

Publications

Conference papers

1. *Cristanel Razafimandimby*, Valeria Loscri, Anna Maria Vegni, Alessandro Neri. **A Bayesian approach for an efficient data reduction in IoT .** 3rd EAI International Conference on Interoperability in IoT, Nov 2017, Valencia, Spain.

2. *Cristanel Razafimandimby*, Valeria Loscri, Anna Maria Vegni, Alessandro Neri. **Efficient Bayesian Communication Approach For Smart Agriculture Applications.** IEEE 86th Vehicular Technology Conference, Sep 2017, Toronto, Canada.

3. *Cristanel Razafimandimby*, Valeria Loscri, Anna Maria Vegni, Alessandro Neri. **Une communication bayésienne et intelligente pour l'Internet des Objets.** Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, May 2017, Quiberon, France.

4. *Cristanel Razafimandimby*, Valeria Loscri, Anna Maria Vegni, Alessandro Neri. **A Bayesian and Smart Gateway Based Communication For Noisy IoT Scenario.** International Conference on Computing, Networking and Communications, Jan 2017, Silicon Valley, United States.

5. *Cristanel Razafimandimby*, Valeria Loscri, Anna Maria Vegni. **A neural network and IoT-based scheme for performance assessment in Internet of Robotic Things.** 1st International Workshop on Interoperability, Integration, and Interconnection of Internet of Things Systems, Apr 2016, Berlin, Germany.

Book section

1. *Cristanel Razafimandimby*, Valeria Loscri, 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

Bibliography

[1] Ajith Abraham. "Artificial neural networks". In: *handbook of measuring system design* (2005).

[2] Cedric Adjih et al. "FIT IoT-LAB: A large scale open experimental IoT testbed". In: *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE. 2015, pp. 459–464.

[3] Mazda Ahmadi and Peter Stone. "Keeping in touch: Maintaining biconnected structure by homogeneous robots". In: *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*. Vol. 21. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. 2006, p. 580.

[4] Kemal Akkaya, Ismail Guneydas, and Ali Bicak. "Autonomous actor positioning in wireless sensor and actor networks using stable-matching". In: *International Journal of Parallel, Emergent and Distributed Systems* 25.6 (2010), pp. 439–464.

[5] Gianluca Aloi et al. "STEMNet: an evolutionary network architecture for smart and sustainable cities". In: *Transactions on Emerging Telecommunications Technologies* 25 ().

[6] Tamio Arai, Enrico Pagello, and Lynne E Parker. "Editorial: Advances in multi-robot systems". In: *IEEE Transactions on robotics and automation* 18.5 (2002), pp. 655–661.

[7] Rainforest Automation. *EMU-2*. http://rainforestautomation.com. Accessed March 22, 2017.

[8] R Beckers, OE Holland, and Jean-Louis Deneubourg. "From local actions to global tasks: Stigmergy and collective robotics". In: *Artificial life IV*. Vol. 181. 1994, p. 189.

[9] Christopher M Bishop. "Pattern recognition and Machine Learning". In: *Machine Learning* 128 (2006).

[10] Noury Bouraqadi et al. "Making networked robots connectivity-aware". In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE. 2009, pp. 3502–3507.

[11] Keoma Brun-Laguna et al. "(Not so) Intuitive Results from a Smart Agriculture Low-Power Wireless Mesh Deployment". In: *CHANTS'16*. New York City, United States, Sept. 2016. DOI: 10.1145/2979683.2979696. URL: https://hal.inria.fr/hal-01361333.

[12] Adam Campbell, Cortney Riggs, and Annie S Wu. "On the impact of variation on self-organizing systems". In: *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*. IEEE. 2011, pp. 119–128.

[13] Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. "Cooperative mobile robotics: Antecedents and directions". In: *Autonomous robots* 4.1 (1997), pp. 7–27.

[14] Arnaud Casteigts et al. "Biconnecting a network of mobile robots using virtual angular forces". In: *Computer Communications* 35.9 (2012), pp. 1038–1046.

[15] Jiming Chen, Shijian Li, and Youxian Sun. "Novel deployment schemes for mobile sensor networks". In: *Sensors* 7.11 (2007), pp. 2907–2919.

[16] Cisco. *Internet of Things*. http://www.cisco.com/web/solutions/trends/iot/portfolio.html. Accessed March 22, 2017.

[17] Council. *Internet of Things council*. http://www.theinternetofthings.eu. Accessed November 18, 2015.

[18] Frederick Ducatelle et al. "Self-organized cooperation between robotic swarms". In: *Swarm Intelligence* 5.2 (2011), pp. 73–96.

[19] GaoJun Fan and ShiYao Jin. "Coverage problem in wireless sensor network: A survey." In: *JNW* 5.9 (2010), pp. 1033–1040.

[20] Marco Fiore. *NS-2.29 Wireless Update Patch*. http://perso.citi.insa-lyon.fr/mfiore/research.html. Accessed November 19, 2015.

[21] Willow Garage. *PR2 robot*. http://www.willowgarage.com/pages/pr2/overview. Accessed March 15, 2017.

[22] Michele Garetto et al. "A distributed sensor relocatlon scheme for environmental control". In: *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*. IEEE. 2007, pp. 1–10.

[23] Maria Carmela De Gennaro and Ali Jadbabaie. "Decentralized control of connectivity for multi-agent systems". In: *Decision and Control, 2006 45th IEEE Conference on*. Citeseer, 2006, pp. 3628–3633.

[24] Zoubin Ghahramani. "Graphical models: parameter learning". In: *Handbook of brain theory and neural networks* 2 (2002), pp. 486–490.

[25] Almero Gouws. "A Python implementation of graphical models". PhD thesis. Stellenbosch University, 2010.

[26] M Ani Hsieh et al. "Maintaining network connectivity and performance in robot teams". In: (2008).

[27] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2014.

[28] Josna Jose, S Manoj Kumar, and Joyce Jose. "Energy efficient recoverable concealed data aggregation in wireless sensor networks". In: *Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), 2013 International Conference on*. IEEE. 2013, pp. 322–329.

[29] Caner Komurlu and Mustafa Bilgic. "Active inference and dynamic gaussian bayesian networks for battery optimization in wireless sensor networks". In: *Proceedings of AAAI workshop on artificial intelligence for smart grids and smart buildings*. 2016.

[30] Frank R Kschischang, Brendan J Frey, and H-A Loeliger. "Factor graphs and the sum-product algorithm". In: *IEEE Transactions on information theory* 47.2 (2001), pp. 498–519.

[31] Santosh Kumar, Ten H Lai, and Anish Arora. "Barrier coverage with wireless sensors". In: *Proceedings of the 11th annual international conference on Mobile computing and networking*. ACM. 2005, pp. 284–298.

[32] Van Tuan Le. "Cooperation in multi-robot systems: a contribution to maintaining the connectivity and to dynamic allocation of roles". Theses. Université de Caen, Oct. 2010. URL: https://hal.archives-ouvertes.fr/tel-01075290.

[33] Geunho Lee and Nak Young Chong. "A geometric approach to deploying robot swarms". In: *Annals of Mathematics and Artificial Intelligence* 52.2-4 (2008), pp. 257–280.

[34] Jun Li et al. "An extended virtual force-based approach to distributed self-deployment in mobile sensor networks". In: *International Journal of Distributed Sensor Networks* 2012 (2012).

[35] G Mahadevan. "On domination theory and related concepts in graphs". In: (2015).

[36] Francesco Marcelloni and Massimo Vecchio. "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks". In: *The Computer Journal* 52.8 (2009), pp. 969–987.

[37] Enrico Natalizio and Valeria Loscrí. "Controlled mobility in mobile sensor networks: advantages, issues and challenges". In: *Telecommunication Systems* (2013), pp. 1–8.

[38] W. Hong S. Madden M. Paskin P. Bodik C. Guestrin and R. Thibaux. *Intel Lab Data*. http://www.select.cs.cmu.edu/data/labapp3/index.html. Accessed July 20, 2016.

[39] Lynne E Parker. "Multiple mobile robot systems". In: *Springer Handbook of Robotics*. Springer, 2008, pp. 921–941.

[40] Parth D Patel, Pranav B Lapsiwala, and Ravindra V Kshirsagar. "Data aggregation in wireless sensor network". In: *International Journal of Managment, IT and Engineering* 2.7 (2012), pp. 457–472.

[41] Javier Del Prado Pavon and Sunghyun Choi. "Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement". In: *Communications, 2003. ICC'03. IEEE International Conference on*. Vol. 2. IEEE. 2003, pp. 1108–1113.

[42] Riccardo Petrolo, Valeria Loscri, and Nathalie Mitton. "Towards a smart city based on cloud of thoings". In: *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities, WiMobCity*. ACM. 2014, pp. 61–66.

[43] Arun Prasath et al. "Self-organization of sensor networks with heterogeneous connectivity". In: *Sensor Networks*. Springer, 2010, pp. 39–59.

[44] Amanda Prorok, M Ani Hsieh, and Vijay Kumar. "Fast Redistribution of a Swarm of Heterogeneous Robots". In: *International Conference on Bio-inspired Information and Communications Technologies*. 2015.

[45] Ou Qinghai et al. "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*. Vol. 2. IEEE. 2003, pp. 1293–1303.

[46] ABI Research. *Internet of Robotic Things*. https://www.abiresearch.com. Accessed November 2, 2015.

[47] Martin Riedmiller. "Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms". In: *Computer Standards & Interfaces* 16.3 (1994), pp. 265–278.

[48] Nexter Robotics. *Wifibots*. http://www.wifibot.com/. Accessed March 15, 2017.

[49] Lorenzo Sabattini et al. "Distributed control of multirobot systems with global connectivity maintenance". In: *Robotics, IEEE Transactions on* 29.5 (2013), pp. 1326–1332.

[50] Michael Schuresko and Jorge Cortés. "Distributed motion constraints for algebraic connectivity of robotic networks". In: *Journal of Intelligent and Robotic Systems* 56.1-2 (2009), pp. 99–126.

[51] Surender Kumar Soni, Narottam Chand, and Dhirendra Pratap Singh. "Reducing the data transmission in WSNs using time series prediction model". In: *Signal Processing, Computing and Control (ISPCC), 2012 IEEE International Conference on*. IEEE. 2012, pp. 1–5.

[52] Kannan Srinivasan and Philip Levis. "RSSI Is Under-Appreciated". In: *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*. May 2006.

[53] Tossaporn Srisooksai et al. "Practical data compression in wireless sensor networks: A survey". In: *Journal of Network and Computer Applications* 35.1 (2012), pp. 37–59.

[54] Ethan Stump, Ali Jadbabaie, and Vijay Kumar. "Connectivity management in mobile robot teams". In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE. 2008, pp. 1525–1530.

[55] Liansheng Tan and Mou Wu. "Data reduction in wireless sensor networks: A hierarchical LMS prediction approach". In: *IEEE Sensors Journal* 16.6 (2016), pp. 1708–1715.

[56] Chaohui Wang, Nikos Komodakis, and Nikos Paragios. "Markov random field modeling, inference & learning in computer vision & image understanding: A survey". In: *Computer Vision and Image Understanding* 117.11 (2013), pp. 1610–1627.

[57] Guiling Wang, Guohong Cao, and Thomas F La Porta. "Movement-assisted sensor deployment". In: *IEEE Transactions on Mobile Computing* 5.6 (2006), pp. 640–652.

[58] Thomas Watteyne et al. *Peach project*. https://project.inria.fr/peach/. Accessed May 31, 2017.

[59] Mou Wu, Liansheng Tan, and Naixue Xiong. "Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications". In: *Information Sciences* 329 (2016), pp. 800–818.

[60] Jonathan S Yedidia, William T Freeman, and Yair Weiss. "Understanding belief propagation and its generalizations". In: *Exploring artificial intelligence in the new millennium* 8 (2003), pp. 236–239.

[61] Seokhoon Yoon et al. "Coordinated locomotion and monitoring using autonomous mobile sensor nodes". In: *IEEE Transactions on Parallel and Distributed Systems* 22.10 (2011), pp. 1742–1756.

[62] Michael M Zavlanos and George J Pappas. "Distributed connectivity control of mobile networks". In: *Robotics, IEEE Transactions on* 24.6 (2008), pp. 1416–1428.

[63] Wei Zhao and Yao Liang. "A systematic probabilistic approach to energy-efficient and robust data collections in wireless sensor networks". In: *International Journal of Sensor Networks* 7.3 (2010), pp. 162–175.

[64] Yao Zou and Krishnendu Chakrabarty. "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*. Vol. 2. IEEE. 2003, pp. 1293–1303.