



**HAL**  
open science

# Bio-inspired Solutions for Optimal Management in Wireless Sensor Networks

Ado Adamou Abba Ari

► **To cite this version:**

Ado Adamou Abba Ari. Bio-inspired Solutions for Optimal Management in Wireless Sensor Networks. Artificial Intelligence [cs.AI]. Université Paris Saclay (COMUE); Université de Ngaoundéré (Cameroun), 2016. English. NNT : 2016SACLV044 . tel-01473938

**HAL Id: tel-01473938**

**<https://theses.hal.science/tel-01473938>**

Submitted on 22 Feb 2017

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

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

THESE DE DOCTORAT  
DE  
L'UNIVERSITE DE NGAOUNDERE  
ET DE  
L'UNIVERSITE PARIS-SACLAY  
PREPAREE A L'UNIVERSITE DE VERSAILLES SAINT-QUENTIN-EN-  
YVELINES

ÉCOLE DOCTORALE N°580  
Sciences et Technologies de l'Information et de la Communication (STIC)  
Spécialité de Doctorat: Informatique

Par

**M. Ado Adamou ABBA ARI**

Bio-inspired Solutions for Optimal Management in Wireless Sensor Networks

**Thèse présentée et soutenue à Versailles, le 12 juillet 2016.**

**Composition du Jury:**

Mme Oum-El-Kheir AKTOUF	Maître de conférences - HDR	INP Grenoble, France	Rapporteur
M. Adlen KSENTINI	Maître de conférences - HDR	EURECOM, France	Rapporteur
M. Alain BUI	Professeur	UVSQ, France	Examineur
M. Jean-Frédéric MYOUPPO	Professeur	Université de Picardie Jules Verne, France	Examineur
M. Samir TOHMÉ	Professeur	UVSQ, France	Examineur
M. François SPIES	Professeur	Université de Franche-Comté, France	Président
M. Abdelhak GUEROUI	Maître de conférences - HDR	UVSQ, France	Directeur de Thèse
M. Blaise Omer YENKÉ	Maître de conférences	Université de Ngaoundéré, Cameroun	Co-Directeur de Thèse
Mme Gladys DIAZ	Maître de conférences - HDR	Université de Paris 13, France	Invitée



# Abstract

During the past few years, wireless sensor networks witnessed an increased interest in both the industrial and the scientific community due to the potential wide area of applications. However, sensors' components are designed with extreme resource constraints, especially the power supply limitation. It is therefore necessary to design low power, scalable and energy efficient protocols in order to extend the lifetime of such networks. Cluster-based sensor networks are the most popular approach for optimizing the energy consumption of sensor nodes, in order to strongly influence the overall performance of the network. In addition, routing involves non negligible operations that considerably affect the network lifetime and the throughput. In this thesis, we addressed the clustering and routing problems by hiring intelligent optimization methods through biologically inspired computing, which provides the most powerful models that enabled a global intelligence through local and simple behaviors. We proposed a distributed clustering approach based on the nest-sites selection process of a honeybee swarm. We formulated the distributed clustering problem as a social decision-making process in which sensors act in a collective manner to choose their cluster heads. To achieve this choice, we proposed a multi-objective cost-based fitness function. In the design of our proposed algorithm, we focused on the distribution of load balancing among each cluster member in order to extend network lifetime by making a tradeoff between the energy consumption and the quality of the communication link among sensors. Then, we proposed a centralized cluster-based routing protocol for wireless sensor networks by using the fast and efficient searching features of the artificial bee colony algorithm. We formulated the clustering as a linear programming problem and the routing problem is solved by proposing a cost-based function. We designed a multi-objective fitness function that uses the weighted sum approach, in the assignment of sensors to a cluster. The clustering algorithm allows the efficient building of clusters by making a tradeoff between the energy consumption and the quality of the communication link within clusters while the routing is realized in a distributed manner. The proposed protocols have been intensively experimented with a number of topologies in various network scenarios and the results are compared with the well-known cluster-based routing protocols. The results demonstrated the effectiveness of the proposed protocols.



# Résumé

Au cours de ces dernières années, les réseaux de capteurs sans fils ont connu un intérêt croissant à la fois au sein de la communauté scientifique et industrielle en raison du large potentiel en terme d'applications offertes. Toutefois, les capteurs sont conçus avec d'extrêmes contraintes en ressources, en particulier la limitation de l'énergie. Il est donc nécessaire de concevoir des protocoles efficaces, évolutifs et moins consommateur d'énergie afin de prolonger la durée de vie de ces réseaux. Le *clustering* est une approche très populaire, utilisée pour l'optimisation de la consommation d'énergie des capteurs. Cette technique permet d'influencer fortement la performance globale du réseau. En outre, dans de tels réseaux, le routage génère un nombre assez élevé d'opérations non négligeables qui affectent considérablement la durée de vie du réseau ainsi que le débit offert. Dans cette thèse, nous nous sommes intéressés d'une part aux problèmes de *clustering* et de routage en utilisant des méthodes d'optimisation inspirées de certaines sociétés biologiques fournissant des modèles puissants qui conduisent à l'établissement d'une intelligence globale en se basant sur des comportements individuels très simples. Nous avons proposé une approche de *clustering* distribuée basée sur le processus de sélection des sites de nidification chez les colonies d'abeilles. Nous avons formulé le problème de *clustering* distribué comme un processus social de prise de décision dans lequel les capteurs agissent d'une manière collective pour choisir des représentants au sein de leurs *clusters* respectifs. Le protocole proposé assure une distribution de l'équilibrage de charge entre les membres de chaque *cluster* afin de prolonger la durée de vie du réseau en faisant un compromis entre la consommation d'énergie et la qualité du canal de communication. D'autre part, nous avons proposé un protocole de routage basé sur des *clusters* en utilisant un algorithme inspiré du phénomène de butinage des abeilles. Nous avons formulé le problème de *clustering* comme un problème de programmation linéaire alors que le problème du routage est résolu par une fonction de coûts. L'algorithme de *clustering* permet la construction efficace des *clusters* en faisant un compromis entre la consommation d'énergie et la qualité du canal de communication au sein des *clusters* tandis que le routage est réalisé de manière distribuée. Les protocoles proposés ont été intensivement expérimentés sur plusieurs topologies dans différents scénarios de réseaux et comparés avec des protocoles bien connus de *clustering* et routage. Les résultats obtenus démontrent l'efficacité des protocoles proposés.



# Acknowledgements

Above all, I give thanks to God for giving me life, strength and inspiration to accomplish this task.

Naturally, the first gratitude is addressed to my Ph.D. supervisors Dr Abdelhak GUEROUI and Dr Blaise Omer YENKÉ. They have provided me with encouragement, inspiration, valuable advice, friendship and sympathy, which enabled me to carry out this thesis. For me, it is a great honor to be your student.

I am very grateful to Dr Nabila LABRAOUI, my co-supervisor, for inviting me to investigate this research area and also for her availability, openness, generosity, fruitful discussions and valuable remarks that helped us in better improving the work done during this Ph.D. program. I hope that this collaboration will remain successful in the future.

I would like to thank the LI-PaRAD Laboratory and the French Ministry of Foreign Affairs and International Development for resources and financial support that helped us to achieve the work done in this thesis.

I would also like to thank Professor David BÉKOLLÉ and Dr Irépran DAMAKOA. Their charisma and experiences have largely encouraged my curiosity and developed in me a great enthusiasm for the scientific research.

I express my gratitude to all my family for the valuable support they provide to me. Especially, I am very grateful to my mother and my father, who raised me with the love of study from my childhood and supported me all times.

I reserve a special gratitude, a real esteem and a particular thought to Djalé LANDI for her outstanding advice and constant encouragement.





# Contents

<b>Abstract</b>	<b>ii</b>
<b>Résumé</b>	<b>iv</b>
<b>Acknowledgements</b>	<b>vi</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Figures</b>	<b>xiv</b>
<b>List of Abbreviations</b>	<b>xvii</b>
<b>List of Algorithms</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Thesis Contributions . . . . .	5
1.3 List of Publications . . . . .	6
1.3.1 Journal Papers . . . . .	7
1.3.2 Conference Papers . . . . .	7
1.4 Other Publications . . . . .	7
1.4.1 Journal Papers . . . . .	7
1.4.2 Conference Paper . . . . .	8
1.5 Organization of the Thesis . . . . .	8
<b>I Literature Review</b>	<b>11</b>
<b>2 WSNs and Bio-inspired Solutions</b>	<b>13</b>

2.1	Introduction . . . . .	13
2.2	Wireless Sensor Networks . . . . .	14
2.2.1	Background . . . . .	14
2.2.2	Architectures of WSNs . . . . .	15
2.2.3	Offered Services in WSNs . . . . .	16
2.2.4	Applications of WSNs . . . . .	18
2.2.5	WSNs' Challenges . . . . .	19
2.3	Bio-inspired Solutions . . . . .	20
2.3.1	Background . . . . .	20
2.3.2	Overview of honeybees swarm intelligence . . . . .	20
2.3.3	Artificial Bee Colony Algorithm . . . . .	22
2.3.4	Collective Decision in Honeybees Nest-sites Selection Process . . . . .	24
2.4	Conclusion . . . . .	26
<b>3</b>	<b>Review of Cluster-based Routing Protocols</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Heuristic based Approaches . . . . .	27
3.2.1	Low Energy Adaptive Clustering Hierarchy . . . . .	27
3.2.2	Energy Efficient Hierarchical Clustering . . . . .	29
3.2.3	Hybrid Energy-Efficient Distributed clustering . . . . .	30
3.2.4	Clustering Algorithm for Efficient Energy Saving in WSNs . . . . .	31
3.2.5	Hierarchical Energy Efficient Routing Protocol . . . . .	32
3.2.6	Energy-Aware Clustering Algorithm . . . . .	33
3.3	Meta Heuristic based Approaches . . . . .	35
3.3.1	Low Energy Adaptive Clustering Hierarchy - Centralized . . . . .	35
3.3.2	Energy-aware Clustering for WSNs using the PSO Algorithm . . . . .	36
3.3.3	Cluster-based WSN Routing using the ABC Algorithm . . . . .	37
3.3.4	Multi-tier Cluster-based Routing Protocol for Sensor Networks . . . . .	38
3.3.5	Energy-Efficient Clustering and Routing Algorithms for WSNs . . . . .	39
3.3.6	PSO protocol for Hierarchical Clustering in WSNs . . . . .	40
3.3.7	Energy-efficient and Scalable Multipath Routing Protocol for WSNs . . . . .	41
3.4	Synthesis . . . . .	42
3.5	Conclusion . . . . .	42

<b>II</b>	<b>Contributions</b>	<b>45</b>
<b>4</b>	<b>A Self-organized Clustering Algorithm for WSNs</b>	<b>47</b>
4.1	Introduction . . . . .	47
4.2	Network model . . . . .	48
4.2.1	Assumptions . . . . .	48
4.2.2	Energy model . . . . .	49
4.3	Design Goals . . . . .	50
4.4	Proposed Clustering Protocol . . . . .	50
4.4.1	Overview of our approach . . . . .	51
4.4.2	Initialization step: CH candidate . . . . .	52
4.4.3	CH competition step: dance strength determination . . . . .	53
4.4.4	Scheduling step and CH election strategy . . . . .	54
4.4.5	Proposed clustering algorithm . . . . .	54
4.4.6	Complexity analysis . . . . .	54
4.5	Simulation and Discussion . . . . .	54
4.6	Conclusion . . . . .	63
<b>5</b>	<b>A Power Efficient Cluster-based Routing Algorithm for WSNs</b>	<b>65</b>
5.1	Introduction . . . . .	65
5.2	Network Model and Vocabulary . . . . .	66
5.2.1	Energy Model . . . . .	66
5.2.2	Vocabulary . . . . .	67
5.2.3	Assumptions . . . . .	68
5.3	Algorithms Design . . . . .	68
5.3.1	Design goals . . . . .	68
5.3.2	ABC-SD clustering mechanism . . . . .	70
5.3.3	ABC-SD routing mechanism . . . . .	78
5.4	Performance Evaluation . . . . .	83
5.4.1	Theoretical analysis of energy consumption . . . . .	84
5.4.2	Metrics of the performance evaluation . . . . .	88
5.4.3	Simulation and discussion . . . . .	89
5.5	Conclusion . . . . .	101

<b>6 Conclusion</b>	<b>103</b>
6.1 Conclusion . . . . .	103
6.2 Future Research Directions . . . . .	105
<b>Bibliography</b>	<b>107</b>

# List of Tables

3.1	Synthesis of cluster-based protocols . . . . .	43
4.1	The used Notations in the NEST Algorithm . . . . .	48
4.2	Messages in the NEST Algorithm . . . . .	49
4.3	Considered analogies between a Bee Colony and a Sensor Network . . . . .	49
4.4	Adopted Simulation Parameters for the NEST . . . . .	56
4.5	Adopted Swarm Parameters for the NEST . . . . .	57
4.6	Fitness functions of compared protocols with the NEST . . . . .	59
4.7	Mean of time at which the first sensor dies ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ). . . . .	60
4.8	Average residual energy of each node ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ) during 10 runs. . . . .	61
4.9	Mean of PDR ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ) during 10 runs. . . . .	61
5.1	Energy consumption of the different states in the Chipcon CC2420. . . . .	67
5.2	Descriptions of the fields in the packet format. . . . .	79
5.3	The used messages in the ABC-SD routing algorithm. . . . .	79
5.4	The used ABC parameters for simulating the ABC-SD. . . . .	90
5.5	Adopted Simulation parameters in the ABC-SD. . . . .	90
5.6	Average sensor's energy consumption ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ). . . . .	92
5.7	Mean of throughput ( $\bar{\mu}(\times 10^6)$ ) and standard deviation ( $\sigma(\times 10^6)$ ). . . . .	96
5.8	Mean of packet loss rate ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ). . . . .	97
5.9	Average sensor's energy consumption ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ) - sensing area of $500 \times 500 m^2$ . . . . .	99
5.10	Mean of throughput ( $\bar{\mu}(\times 10^6)$ ) and standard deviation ( $\sigma(\times 10^6)$ ) - sensing area of $500 \times 500 m^2$ . . . . .	101



# List of Figures

1.1	Cluster-based WSN: a scheme of delivery of gathered data by sensors to the BS. . . . .	3
2.1	WSNs' flat architectures. . . . .	15
2.2	WSNs' hierarchical architectures. . . . .	16
2.3	Classification of WSNs' applications. . . . .	18
4.1	Box plot of CH number in the NEST during 10 runs. . . . .	58
4.2	Number of CHs and number of CH elections in the NEST during 10 runs. . . . .	58
4.3	Network lifetime . . . . .	59
4.4	Amount of energy consumed . . . . .	60
4.5	Amount of packets received by the BS . . . . .	62
4.6	Average of end-to-end delay during 10 runs. . . . .	63
5.1	ABC-SD: Design scheme. . . . .	69
5.2	The main steps of the proposed ABC-SD protocol. . . . .	70
5.3	Cluster presented like a k-way tree where the <i>CH</i> is the root and at each level of the tree, each sensor $s_i \in c_1$ has no more than k children. . . . .	75
5.4	An illustration for the construction of the <i>TDMA</i> frame for the cluster $c_1$ . . . . .	75
5.5	The adopted unique packet format with its eight fields for the data transfer over the network. . . . .	78
5.6	Route discovery in the ABC-SD. . . . .	82
5.7	Steady-state: a sequence of actions in data transfer and acknowledgement mechanism in the ABC-SD. . . . .	83
5.8	Packet retransmission process in the ABC-SD. . . . .	83
5.9	Flow chart of the ABC-SD cluster-based routing protocol. . . . .	84
5.10	Energy consumed by each node in each value of $\alpha$ . . . . .	92



5.11 Energy consumption . . . . .	93
5.12 Energy efficiency . . . . .	94
5.13 Time at which the first sensor dies. . . . .	95
5.14 Amount of data packets received by the <i>BS</i> . . . . .	96
5.15 Packets loss. . . . .	97
5.16 The communication overhead in ABC-SD. . . . .	98
5.17 Amount of unclustered sensors. . . . .	99
5.18 Energy consumption in a sensing area of $500 \times 500 m^2$ . . . . .	100
5.19 Amount of data packets received by the <i>BS</i> in a sensing area of $500 \times 500 m^2$ . . . . .	100

# List of Abbreviations

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
AIS	Artificial Immune System
BFO	Bacterial Foraging Optimization
BFOA	Bacterial Foraging Optimization Algorithm
B-MAC	Berkeley Media Access Control
BS	Base Station
CDMA	Code Division Multiple Access
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CA	Carrier Sense Multiple Access with Collision Detection
CH	Cluster Head
CHs	Cluster Heads
EADC	Energy-Aware Clustering Algorithm
EEHC	Energy Efficient Hierarchical Clustering
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-C	Low Energy Adaptive Clustering Hierarchy - Centralized
LP	Linear Programming
GA	Genetic Algorithm
GPS	Global Positioning System
HBMO	Honey Bee Mating Optimization
HEED	Hybrid Energy-Efficient Distributed clustering
HEERP	Hierarchical Energy Efficient Routing Protocol
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
UML	Unified Modeling Language
MAC	Medium Access Control
NLP	Non Linear Programming
NP	Non Polynomial
PSO	Particle Swarm Optimization

PSO-C	Particle Swarm Optimization - Centralized
QoS	Quality of Service
SA	Simulated Annealing
SI	Swarm Intelligence
S-MAC	Sensor Medium Access Control
RSSI	Received Signal Strength Indicator
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network
WSNs	Wireless Sensor Networks

# List of Algorithms

2.1	Algorithm simulating the ABC	23
4.1	Algorithm Simulating the NEST	55
5.1	ABC-SD: Clustering Algorithm	72
5.2	The ABC-SD route discovery algorithm	81



# Chapter 1

## Introduction

### 1.1 Motivation

With the rapid progress of wireless communication technologies, the popularity of Wireless Sensor Networks (WSNs) has attracted the attention of both Industry and Academy in researching and developing promising and tremendous real life solutions during the last few years. Sensor-based infrastructure such as Wireless Sensor Network (WSN) provides promising Green Computing techniques for enhancing information gathered on a specific environment to the end users.

A WSN is a powerful *infrastructureless* network consisting of ten to some thousands autonomous low power sensors organized in an ad hoc manner. These sensors are able to gather and process information from an environment, and communicate with each other. In this kind of network, sensors are randomly or manually deployed on a physical environment that acts as the sensing layer of the Internet of Things (IoT) and have a wide area of applications. WSNs are usually used in both civilian and military applications to instrument, observe and react to an occurred event or phenomena, on a remote or inaccessible environment [1, 2]. These applications are both in tracking such as military strategies or seismic measurements; and in monitoring such as prevention of natural disasters or agricultural irrigation management.

In a WSN, each sensor is equipped with a sensing unit, a radio transmission module, a power supply, a processing and data storage devices. However, these components are designed with huge resource constraints like energy and processing capabilities [3, 4]. The sensor power supply limitation is the main constraint in WSNs. In addition, due to the unattended and hostile nature of the sensing environment, it is not easy to replace the batteries of thousands of deployed sensors. Therefore, the energy saving of sensors is a challenging issue that needs investigation in order to prolong the network lifetime from months to years.

Certainly, a number of issues has been studied for improving power management strategies in WSNs as well as sensors' architectures. These research include energy consumption and low power devices [5, 6, 7], Medium Access Control (MAC) protocols for WSNs [8, 9], clock synchronization techniques [10], secure data aggregation [11, 12], and efficient routing techniques [13, 14, 15, 16].

In spite of their energy constraints, the limited computing, storage and communication link capacities, sensors are deployed with the objective of gathering data and transmitting them to a given Base Station (BS) in a collective and self-organized manner [17]. This extreme resource limitation requires robust and low power mechanisms of collecting and routing data. Many designs of sensor networks allow sensors to transmit their gathered data to a mobile or static BS in a multihop routing [18, 19]. However, a robust and low power mechanism of collecting and routing data towards the BS in an efficient way remains a challenge. Therefore, the design and maintenance of such large WSNs requires efficient management strategies and scalable architectures.

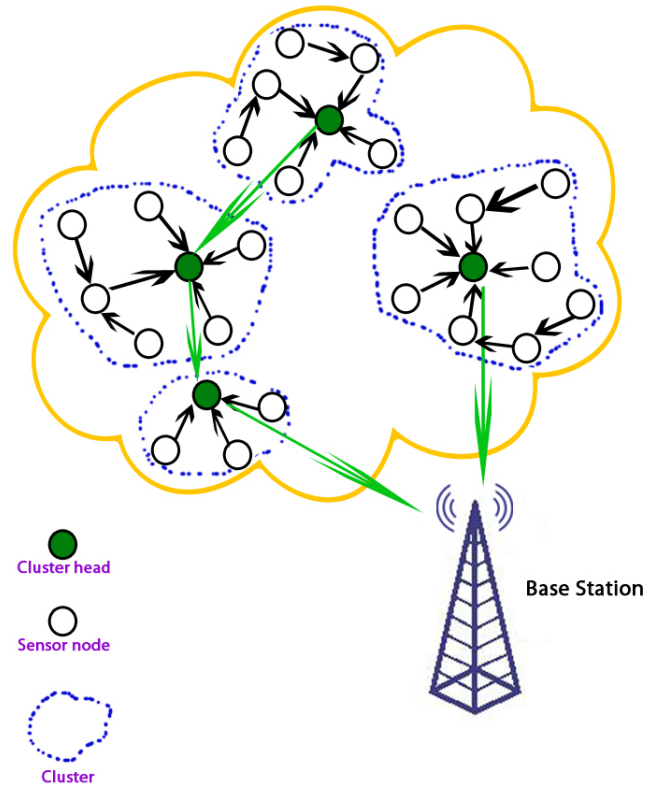
Applications running on WSNs need a network with a long lifetime. To increase network lifespan, clustering is the most widely used technique for efficiently managing network energy consumption and scalability. Moreover, designing low power clustering algorithms for WSNs remains a challenging issue that would allow a significant enhancement of the lifetime of the network [20].

In WSNs, clustering techniques consist of partitioning the network into a varied number of sensor groups called clusters. In each cluster, a leader called Cluster Head (CH) is elected either in a distributed manner by sensors themselves [1, 21] or by a centralized control algorithm [22, 23]. In each cluster, sensors gather data and send them to their corresponding Cluster Heads (CHs). The CH aggregates all received data and transmits them to the BS. The above described functional scenario of a cluster-based sensor network is depicted in Figure 1.1.

Data aggregation achieved by CHs would enable the reduction of redundant data and so, a veritable decrease in the communication overhead from sensors to the BS [12]. Also, sensors distribution is a key factor involved in improving network performance apart from the choice of the head sensor of the cluster. In fact, when sensors are well distributed in the sensing area, the network coverage is well maintained and the network lifetime can be extended. However, maintaining a good coverage is well achieved in the case of a pre-deployed network in which the location of each sensor is known or in the case of a randomly deployed sensor network with a certain number of mobile sensors that assure the coverage [24].

In a cluster-based WSN, communication among sensors are in two kinds. The first kind of communication is known as the intra-cluster communication. The second type of communication is called inter-cluster communication. Otherwise, in some designs, the gathered

Figure 1.1: Cluster-based WSN: a scheme of delivery of gathered data by sensors to the BS.



data are transferred to the BS by using intermediate sensors in a flat routing approach [1]. Unlike this way of designing, the cluster-based routing algorithms should allow only CHs to transfer collected data to the BS. This approach helps to reduce the rate of communication towards the BS and also permits to avoid collisions among communications generated by clusters.

Usually, clustering algorithms are grouped into two main techniques: distributed and centralized. In the distributed clustering, after the sensors deployment and the network initialization, CHs are elected by sensors themselves. The clustering algorithm is executed by all sensors in a collective self-organized manner. Some centralized approaches periodically swap the CH [25, 26]. However, a distributed clustering solution that elects a CH when it is needed would be less costly in terms of energy consumption introduced by the amount of communication in the CH election process. In centralized control approaches,



the BS executes the clustering algorithm in order to build clusters and select the CHs. Then, the BS informs each sensor in the network, the cluster in which it belongs [13, 23, 27]. Furthermore, as it is necessary that the intra-cluster communication be carried out in a multihop manner within the cluster, it is also important and efficient that CHs' communications, i.e., the inter-cluster communication, be performed in a multihop manner [28]. Note that, the intra-cluster communication is not always carried out by the multihop approach. In detail, some approaches exclusively allow sensors to directly transmit their gathered data to their respective CHs even if the communication range between them is very large [25]. Similar to the way in these protocols, CHs also directly communicate the aggregated information to the BS in a single hop approach. However, in real cases, the BS as well as CHs may be located far away from certain sensors in the sensing environment or even, the BS may be positioned far away from the sensing environment [2, 28].

In these cases, the BS may not be reachable by CHs in a single hop communication. In addition, according to the widely used energy model proposed in [27], large distances affect the signal propagation and allow a lot of power wastage. Unfortunately, in homogeneous networks, the power supply as well as the communication range of sensors is limited. So, in order to avoid direct communication between a CH and the BS, intermediate sensors should participate in the forwarding of data towards the BS [1, 21]. Another way is to use special sensors called gateways provisioned with extra energy, which act as relay sensors in the inter-cluster communication [16]. However, it seems not realistic for large scale randomly deployed sensor networks. Therefore, it seems beneficial to route packets by using an efficient and low power multihop communication model, both for intra-cluster and inter-cluster communication.

Clustering sensors provides many advantages for an efficiency management of WSNs. It allows power saving in sensors since CHs are capable of eliminating redundant and false positive data by a given aggregation process [12]. By this way, the communication bandwidth is preserved. It also allows a good management of the inter-cluster routing since only CHs maintain the routing information towards the BS.

However, clustering is known as a non-deterministic polynomial NP-hard problems for a WSN since it has been proven that, finding an optimal set of CHs in a WSN is not an easy task [16, 27, 29, 30]. Moreover, since a WSN can be seen as an undirected graph or a tree, there is at least one path (usually a number of paths) to reach a given destination, such as the CH location. So, the path establishment process of the routing algorithm has to produce a set of selected low cost paths among a number of possibilities. This has to be done both for the intra-cluster and the inter-cluster communications. Unfortunately, it is not evident to use a deterministic polynomial algorithm for routes establishment [31].

Swarm Intelligence, which is the study of the collective behavior of social individuals' communities, provides efficient metaheuristic tools and algorithms that deal with a lot

of desirable and interesting properties applied in WSNs [19, 28, 32]. Moreover, swarm intelligence can be seen as analogies between computing methods and biological behaviors of swarms in which collective intelligence can emerge. The considered swarms are colonies of social insects, bird flocking and fish schooling [33]. Each kind of swarm is constituted of simple and autonomous individuals, who cooperate with each other to achieve some tasks necessary for the survival of the colony.

In the current research inclination, hiring of biological solutions to solve and optimize different aspects of artificial systems', i.e., real life problems has been shaped into an important field called bio-inspired computing [34]. In the context of WSNs, combining the simple individuals' output and robust behaviors has been successfully applied by identifying and modelling some analogies between the two systems in order to form a single problem solving solution [17].

Clustering and routing in WSNs are well known optimization problems that are well investigated for developing many swarm intelligence based algorithms. For more than a decade, bio-inspired solutions progressively draw the attention of researchers because of its efficiency in the solving of a number of optimization problems in wide domains. The way of selecting CHs and building clusters by using bio-inspired solutions governs our contributions in this thesis.

## 1.2 Thesis Contributions

The design of low power, scalable and energy efficient protocols in order to extend the lifetime of WSNs has been widely studied in the literature but it remains an active area of research. Routing involves non-negligible operations, which considerably affect the network lifetime and the throughput. The clustering technique with data aggregation on CHs has an influence on the overall performance of the network since it is favoring a maximum network lifetime. A number of novel protocols, architectures, algorithms and applications has been proposed and implemented. In this thesis, we aim at designing low power clustering and routing protocols for large scale WSNs by using bio-inspired solutions. To achieve this goal, we realized a twofold contribution:

1. In the first contribution, we proposed a distributed consensus approach based on the nest-sites selection process of a honeybee swarm to achieve clustering in randomly deployed WSNs. We formulated the distributed clustering problem as a social decision-making process in which sensors act in a collective manner to choose their respective CHs. The proposed clustering algorithm called NEST allows an efficient collective and self-organized building of clusters by making a tradeoff between the energy consumption and the quality of the communication link among sensors. The cluster-reformation process is not performed periodically but when it is needed. In

the design of our proposed algorithm, we focused on the distribution of load balancing among each cluster member in order to extend network lifetime. In order to choose a CH, each sensor evaluates candidates to a CH position by determining their dance strength. The dance strength determination is reduced to a multi-objective cost-based fitness function. We performed intensive simulation on the proposed protocol. We compared our proposal with three existing well known protocols. The comparison is conducted by using several performance metrics including network lifetime, amount of energy consumption, amount of data packets received by the BS and so on. The performance of the proposed approach was evaluated by performing extensive experiments and the results demonstrated that our algorithm delivers better performance in terms of network lifetime, amount of packets received by the BS, end-to-end delay, energy consumption and efficiency.

2. The second contribution consists of designing a centralized cluster-based routing protocol called ABC-SD. The proposed protocol exploits the biologically inspired fast and efficient searching features of the Artificial Bee Colony (ABC) metaheuristic to build low power clusters. For the choice of CHs, a multi-objective fitness function is designed by using a Linear Programming (LP) formulation. The routing problem is addressed by a Cost-based Function (CF) that makes a trade-off between the energy efficiency and the number of hops of the path. The clustering process is achieved at the BS with a centralized control algorithm, which exploits energy levels and the neighborhood information of location-unaware sensors. As for the routing of gathered data, it is realized in a distributed manner. Furthermore, unlike the existing protocols in the literature, a realistic energy model based on the characteristics of the Chipcon CC2420 radio transceiver's data sheet is adopted in the considered network model. The proposed protocol is intensively experimented with a number of topologies in various network scenarios and the results are compared with five well-known cluster-based routing protocols that include the swarm intelligence based protocols. The obtained results demonstrate the effectiveness of the proposed protocol in terms of network lifetime, network coverage and the amount of packets delivered to the BS.

### 1.3 List of Publications

The following publications by the author are the most relevant to the work presented in this thesis.

### 1.3.1 Journal Papers

- [1] **Ari, A. A. A.**, Yenke, B. O., Labraoui, N., Damakoa, I., and Gueroui, A. (2016) 'A power efficient cluster-based routing algorithm for wireless sensor networks: Honeybees Swarm Intelligence based approach', *Journal of Network and Computer Applications*, Vol. 69, pp.77-97. Elsevier. doi:10.1016/j.jnca.2016.04.020. Thomson Reuters Journal Citation Reports 2015 - **Impact Factor: 2.229**.
- [2] **Ari, A. A. A.**, Labraoui, N., Yenke, B. O., and Gueroui, A. 'Clustering algorithm for wireless sensor networks: the honeybee swarms nest-sites selection process based approach', *International Journal of Sensor Networks* (2016). Inderscience. Accepted. Thomson Reuters Journal Citation Reports 2015 - **Impact Factor: 0.92**.
- [3] **Ari, A. A. A.**, Gueroui, A., Labraoui, N., and Yenke, B.O. (2015) 'Concepts and evolution of research in the field of wireless sensor networks', *International Journal of Computer Networks & Communications*, Vol. 7 No.1, pp.81-98. doi:10.5121/ijcnc.2015.7106.

### 1.3.2 Conference Papers

- [1] **Ari, A. A. A.**, Gueroui, A., Yenke, B. O., and Labraoui, N. (2016 January). 'Energy efficient clustering algorithm for wireless sensor networks using the ABC metaheuristic', In *Computer Communication and Informatics (ICCCI), 2016 International Conference on* (pp.1-6). Coimbatore, Tamilnadu, India. IEEE. doi:10.1109/ICCCI.2016.7480010.
- [2] **Ari, A. A. A.**, Gueroui, A., Labraoui, N., Yenke, B. O., Titouna, C. and Damakoa, I. (2016 September). 'Adaptive Scheme for Collaborative Mobile Sensing in Wireless Sensor Networks: Bacterial Foraging Optimization approach', In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on* (pp.1-6). Valencia, Spain. IEEE. Accepted

## 1.4 Other Publications

The others research activities conducted by the author are given in the following.

### 1.4.1 Journal Papers

- [1] Yenke, B. O., **Ari, A. A. A.**, Mbeuyo, C. D., and Voundi, D. A. (2015). 'Virtual Machine Performance upon Intensive Computations'. *GSTF Journal on Computing (JoC)*, Vol. 4 No.3, pp.98-107.
- [2] Yenke, B. O., **Ari, A. A. A.**, Sambo, D. W., and Gueroui, A. 'MMEDD: Multi-threading Model for an Efficient Data Delivery in wireless sensor networks'. *International Journal of Communication Networks and Information Security*. (2016). Accepted.

[3] Titouna, C., Gueroui, M., Aliouat, M., **Ari, A. A. A.**, and Adouane, A. 'FDRA: Fault Dection and Recovery Algorithm for wireless sensor networks'. Submitted on 13.06.2016 in *International Journal of Sensor Networks*.

### 1.4.2 Conference Paper

[1] Yenke, B. O. and **Ari, A. A. A.** (2013 December). 'Virtual Machines Performance in HPC Environments'. In *Conférence de Recherche en Informatique on* (pp.1-4). Yaoundé, Center, Cameroon.

## 1.5 Organization of the Thesis

The thesis is organized as follows.

- Chapter 2 gives a background knowledge about WSNs and biologically inspired solutions. We first give an overview of WSNs by presenting a background that allows the understanding of WSNs, followed by the presentation of WSNs' architectures, offered services, supported applications and challenges in WSN research. Then we present bio-logically inspired computing solutions by starting with a comprehensive overview of swarm intelligence that provides intelligent optimization tools and algorithms, which allow an optimal management of WSNs. This comprehensive discussion includes the presentation of the Artificial Bee Colony algorithm and the collective decision enabled in honeybees nest-sites selection process. These two bio-inspired solutions are applied to formulate and solve clustering and routing problem described in Chapter 4 and Chapter 5.
- Chapter 3 presents a comprehensive review of the most relevant cluster-based routing protocols. The review of these solutions is organized into two categories namely: the heuristic based approaches and the meta heuristic based approaches. Then a synthesis of these solutions is tabulated in order to have a brief view.
- Chapter 4 presents the proposed distributed clustering approach called NEST that is based on the intelligent behaviors enabled by honeybees swarm in their process of selecting a nest-site. To describe the proposal, some assumptions and the adopted network model are presented before giving the design goals. Then, the details of the proposed clustering approach are analyzed and this is followed by the simulation and discussion.
- Chapter 5 presents the proposed centralized cluster-based routing protocol called ABC-SD, which is inspired by foraging behaviors of honeybees. The proposal allows to solve the routing problem by modelling a cost-based function. The ABC-SD

uses the fast and efficient features of the Artificial Bee Colony algorithm to build clusters and select CHs by making a tradeoff between the energy consumption and the quality of the communication link. The chapter starts with the presentation of the adopted network model before presenting the design goals as well as the proposed clustering and routing algorithms. This is followed by the performance evaluation and discussion.

- Chapter 6 concludes this research work and presents our perspectives and the envisioned works that deserve further investigation.



Part I

Literature Review





## Chapter 2

# Wireless Sensor Networks and Bio-inspired Solutions

### 2.1 Introduction

The field of WSNs is experiencing a resurgence of interest and a continuous evolution in the scientific and industrial community. The use of this particular type of ad hoc network is becoming increasingly important in many contexts, regardless of geographical position and so, according to a set of possible applications [2]. WSNs offer interesting low cost and easily deployable solutions to perform a remote real time monitoring, target tracking and recognition of physical phenomenon. The uses of these sensors organized into a network continue to reveal a set of research questions according to particularities of target applications. Biologically inspired solutions can be seen as analogies between computing methods and biological behaviors of swarms in which collective intelligence can emerge [17]. The considered swarms are colonies of social insects, bird flocking and fish schooling. Because of the complexity introduced by large scale sensor networks, biologically inspired solutions are progressively applied in WSNs in order to optimise the whole management of such networks. In this chapter, we give a background knowledge about WSNs and swarm intelligence based solutions.

The rest of the chapter is organized as follows: Section 2.2 gives an overview of WSNs; Section 2.3 presents biologically inspired solutions; and we conclude this chapter in Section 2.4.

## 2.2 Wireless Sensor Networks

### 2.2.1 Background

A WSN is a powerful *infrastructureless* ad hoc network that consists of a number of distributed sensor nodes. WSNs are seen as technological platforms increasingly useful in a wide and tremendous number of civilian and military sectors for environmental monitoring and tracking; seismic monitoring; pollution monitoring; disaster management; automated biomedical health care; security and tactical surveillance; critical infrastructure surveillance; and irrigated agriculture [2].

These networks typically consist of a number of autonomous less-costly sensors manually or randomly deployed on a given area, which communicate through wireless channels and cooperate to process and share information [11, 13]. In most cases, sensors are statically deployed over the sensing area. However, they can also be mobile and capable of interacting with the environment [24, 35, 36].

Sensors are usually highly resource-constrained and are capable of gathering physical information from the environment and transmitting the gathered data to an end-user or a BS by utilizing intermittent wireless communication [37]. In WSNs, a node consists of a sensor unit, a processing and data storage unit, a wireless transmission module and a power management unit. Each sensor is able to gather, process physical information and communicate with neighboring sensors in order to transmit these gathered data to the BS. However, wireless sensors are designed with huge resource constraints: a limited amount of energy; reduced computing capacity; limited memory size and storage; short-range of communication and reduced bandwidth. So, it results some problems in network architectures, QoS, coverage, security, fault tolerance and routing [38]. Therefore, the energy saving of sensors is a challenging issue that needs investigation in order to prolong the network lifetime from months to years. In a WSN, energy consumption depends on network architecture, environment in which the network is deployed, the underlying application and the adopted routing strategy. The routing is a key process to be considered in WSNs for a better energy conservation [13, 29].

In a large scale WSN, the network is designed in such a way that it allows sensors to transmit their gathered data to a mobile or static BS in multi-hop routing [1, 15]. The establishment and maintenance of paths towards the BS is not an evident task because of energy constraints and the limited transmission range. It is therefore necessary to design optimal mechanisms for routing data towards the BS. Clustering is the most widely used technique for efficiently managing network energy consumption and scalability in order to increase network lifespan. This technique allows grouping sensors into clusters and electing a *boss* namely a Cluster Head (CH) in each cluster. Each CH is capable to collect data from the cluster members, aggregate and transfer the aggregated data to the BS.

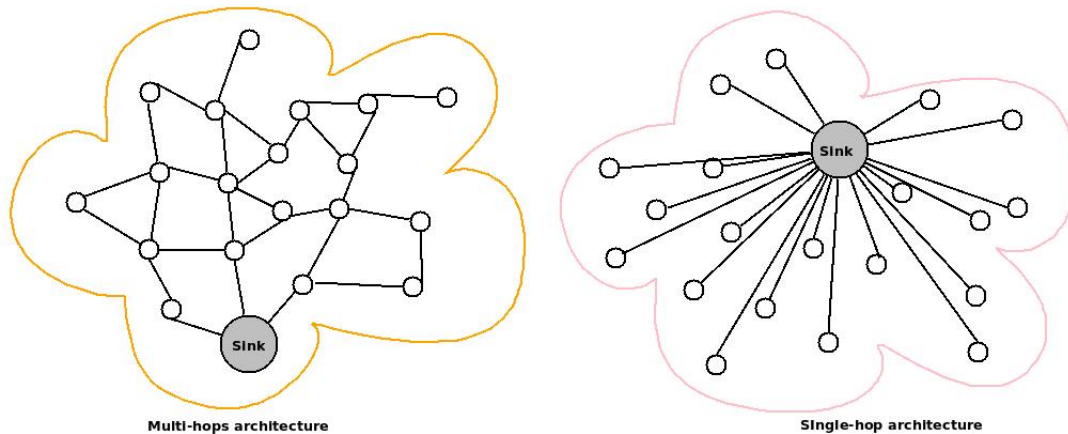
### 2.2.2 Architectures of WSNs

There are two main types of network architectures for WSNs namely the flat architecture and the hierarchical architecture.

#### 2.2.2.1 Flat architectures

In the flat architecture, except for the sink node, the other sensors are identical, they have the same capacity in terms of energy and computing. Also, they have the same role in the sensing task. A sensor can directly communicate with a sink either in a single or a multi hop manner [39]. The simplicity presented in this architecture enables low communication latency. Furthermore, when the network becomes denser, the scaling problem arises mainly as regards the routing. Figure 2.1 presents WSNs' flat architectures.

Figure 2.1: WSNs' flat architectures.

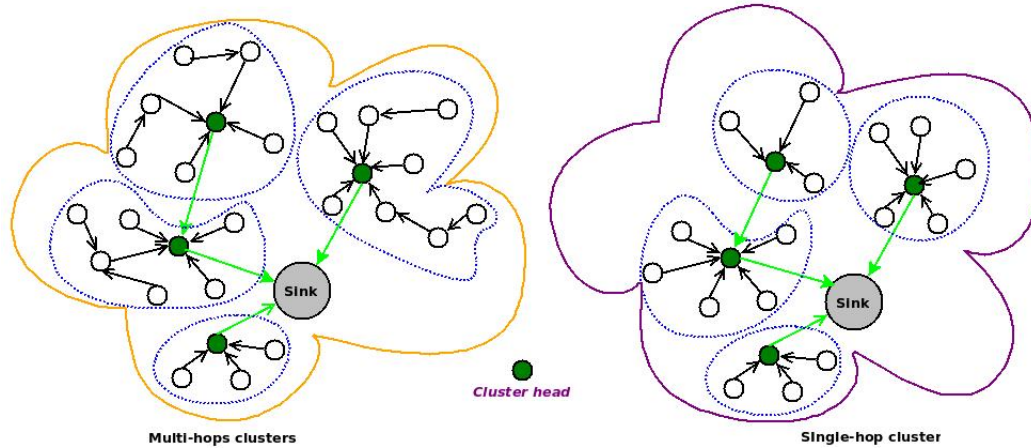


#### 2.2.2.2 Hierarchical architectures

In most cases, hierarchical architectures are used in large sensor networks [40]. In such architectures, the network is divided into several groups or clusters which are the organizational unit of the network. Depending on cases, a more expensive sensor type and more powerful than other sensors or a normal sensor in the cluster is designated as group leader namely the CH that is responsible for the coordination of the sensors under its responsibility and acts as a gateway to another cluster. The CH is responsible for the aggregation and/or compression of all the collected data in order to route it to the sink [41]. This allows the reduction of the transmission data within the network. However, there may be more latency in communications due to the density of the network and higher energy

consumption for the CHs. Figure 2.2 highlights the clustered architectures for multi hop and single hop clusters.

Figure 2.2: WSNs' hierarchical architectures.



### 2.2.3 Offered Services in WSNs

The offered services in WSNs are organized into provisioning, control and sensors management [4, 42]. Coverage and localization are the provisioning services. Control and sensors management enable WSN middle-ware to deliver services such as coverage; localization; synchronization; data compression and data aggregation; security; and fault tolerance.

#### 2.2.3.1 Coverage

It involves the placement of sensors in a deployment area to ensure coverage of the entire area by the network [43]. Coverage depends on the application, the number of sensors and localization of these sensors.

#### 2.2.3.2 Localization

Localization consists of finding position of a sensor in the network [44]. In sensor networks, three methods of localization are usually used [45]:

- the GPS is the easiest but the cost of energy consumption is high and cannot work on dense environment as forest;
- the anchor node approach that consists of a sensor called anchor, which knows its location and helps its neighboring sensors to evaluate their own;

- close localization which consist of a sensor that using the neighboring sensors to determine their locations and then become an anchor node of other sensors.

### 2.2.3.3 Synchronization

Synchronization is an important service in WSNs that allows the realization of an activity or a process at the same time as other sensors. Synchronization is very important especially in routing [46].

### 2.2.3.4 Data compression and aggregation

Data compression reduces the cost of communication and increases the reliability of data transfer. Compression involves the strict reduction of the amount of data to be routed towards the BS. Data compression requires decompression, which can reproduce compressed data in their original format. Usually, the decompression of data is achieved at the BS [47].

The data aggregation can greatly help to conserve the scarce energy resources by eliminating redundant data, thus achieving a longer network lifetime [48, 49]. Data aggregation process usually achieved by CHs, consists of collecting data from all cluster members, applying an aggregation function to all collected data and transmitting just one value as measured data [12].

### 2.2.3.5 Security

WSNs are vulnerable to attacks that consist of compromising a sensor, altering the integrity of the data, listening the network to retrieve messages or injecting false messages in order to create resources' wastage [50]. Despite the difficult task of properly securing a WSN because of the significant limitation of resources, security protocols exist and ensure the security service in WSNs [11, 51].

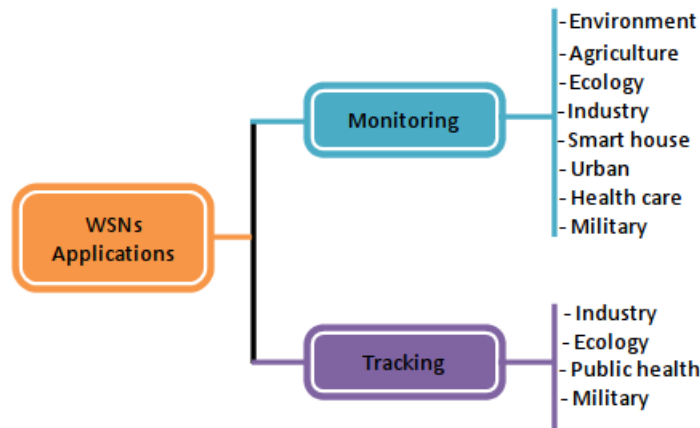
### 2.2.3.6 Fault tolerance

The main goal of fault tolerance in WSNs is to achieve the network scheduling even if there is a faulty sensor in the network. Indeed, fault tolerance in sensor networks can be seen as an ability to maintain the network running without interruption in the presence of failure of a sensor [3]. Fault tolerance is generally implemented in routing and transport protocols. Usually, it consists in the realization of the following two steps: detection of failure and limitation of effect; recovery and treatment of failure [52, 53].

### 2.2.4 Applications of WSNs

The rapid evolution of sensor's technology has led to the design of very small and smart sensors, which allow WSNs to be employed in various applications. WSNs' applications can be categorized mainly into two categories namely monitoring and tracking [4]. In Figure 2.3, we summarize the classification of WSNs' applications.

Figure 2.3: Classification of WSNs' applications.



#### 2.2.4.1 Monitoring

Monitoring is used to analyse, supervise and carefully control operations of a system or a process in real time [2]. WSN-based monitoring applications are various. Below some of them are briefly presented:

1. Environment: monitoring of water quality, weather, pressure, temperature, seismic phenomena, vibration and monitoring of forest fires.
2. Agriculture: irrigation management and humidity monitoring.
3. Ecology: monitoring of animals in their natural environments.
4. Industry: supply chain, inventory monitoring, industrial processes, machinery and productivity.
5. Smart house: monitoring any addressable device in the house.
6. Urban: self-identification, parking management, transport and circulation systems.
7. Health care: organs monitoring, wellness and surgical operation.
8. Military: intrusion detection.

### 2.2.4.2 Tracking

Tracking in WSN is generally used to follow an event, a person, animal or even an object. Existing applications in tracking can be found in various fields.

1. Industry: traffic monitoring, fault detection.
2. Ecology: tracking the migration of animals in various areas.
3. Public health: monitoring of doctors and patients in a hospital.
4. Military: a WSN can be deployed on a battlefield or enemy zone to track, monitor and locate enemy troop movements.

### 2.2.5 WSNs' Challenges

The rapid development of wireless sensor technology is mainly due to the different needs in terms of applications. The main challenge in WSNs is the designing of a network by taking into account a limited energy capacity; resource constraints; random and large deployment; and dynamic and not controlled environment.

Enable reliable communication while maintaining the **Energy Efficiency** in the network is a challenging issue in WSNs research since sensors are very limited in their capacity of storing energy. Besides the efficiency in terms of energy conservation, the **Throughput** is another important requirement of WSNs. Indeed, the amount of packets received by the BS allows the evaluation of the data delivery reliability and therefore the throughput. Then, the capacity of a WSN to handle a great amount of sensors, which is called the **Scalability**, is a critical requirement in the design of large scale sensor networks. Finally, the network **Coverage**, which is an important characteristic in sensor networks, especially in WSNs requiring a high availability of gathered information, is another challenging issue in a static WSN.

To achieve these challenges in a routing protocol, clustering allows sensors to efficiently coordinate their local interactions for the achievement of global goals namely energy efficiency, throughput, coverage and scalability [14, 54]. Due to the limited transmission range of sensors while sensors are usually deployed in large areas of interest, it is not evident to use a deterministic polynomial algorithm for routes establishment [31]. Moreover, bio-inspired solutions are progressively applied in WSNs by proposing some metaheuristic based algorithms for routing in WSNs [1, 13, 16, 21, 23, 55, 56]. Swarm Intelligence (SI), Artificial Immune System (AIS), Genetic Algorithm (GA) and other nature-inspired phenomenon in which self-organization and collective intelligence can emerge, are increasingly used in the fields of sensor networks in order to optimize the whole management of WSNs [57, 58].



## 2.3 Bio-inspired Solutions

### 2.3.1 Background

Biologically inspired solutions also known as nature inspired solutions are metaheuristic solutions based on the collective behavior of social individual communities. These solutions provide efficient tools and algorithms that deal with a lot of desirable and interesting properties applied in WSNs [13, 19, 28]. These solutions enable a set of extremely effective sensory systems that are structurally simple, functionally versatile and powerful, and highly distributed, as well as noise and fault tolerant [59]. The natural behaviors featured by these individuals can lead to the optimization of real life problems [60]. For instance, the Ant Colony Optimization (ACO) metaheuristic [61] has been applied for multipath cluster-based routing in [31]; the Particle Swarm Optimization (PSO) algorithm [62] has been used in [29, 56] to optimize clustering; A Genetic Algorithm (GA) has been used in [63] to ensure target coverage in wireless sensor deployment; and the Artificial Bee Colony (ABC) metaheuristic [64] has been successfully used for clustering and routing in [13, 65, 23]. In addition, metaheuristic algorithms have been applied in many scientific and engineering problems. For instance, Simulated Annealing (SA) has been used for modelling empirical paths loss for *IEEE 802.11n* wireless networks at *2.4 Ghz* in rural regions in Cameroon [66]; Bacterial Foraging Optimization Algorithm (BFOA) has been successfully applied in power system reconfiguration [67] and in photovoltaic parameter estimation [68]; a honey-bee social foraging has been applied for feedback control of smart lights in [69]; and a modified version of the ABC by internal feedback information has been applied in the bioinformatic domain for the prediction of protein secondary structures in [70].

In this section, we present an overview of honeybees swarm intelligence and two used Bio-inspired solutions: the Artificial Bee Colony (ABC) and the collective decision making in honeybees nest-sites selection process. These solutions will be applied to formulated problems in Chapter 4 and Chapter 5.

### 2.3.2 Overview of honeybees swarm intelligence

Honeybees are fascinating and highly organized insects capable of individual cognitive abilities and self-organization. These insects grouped in a colony and living within a hive, show impressive auto-solving problem capabilities by exhibiting a combination of individual traits and social cooperation that is unparalleled in the animal kingdom [71]. These individual capabilities lead to unanimous intelligent decisions in a collective self-organized manner and a decentralized control mechanism. In a honeybee colony, individuals are organized to perform various tasks that are necessary to the survival of the colony. Indeed, they are engaged in several activities: foraging, mating, waggle dance, polarization,

defending the hive and swarming. In this work, we use the honeybees swarm intelligence, which is focused on the collective behaviors that result from local and simple interactions of individuals of a honeybee *Apis mellifera* colony with their environment. Particularly, we are interested in their foraging behaviors.

The foraging behaviors of honeybees *Apis mellifera* consist of a remarkable set of simple and organized tasks in which individuals each with a role, are engaged. Indeed, in a honeybee colony, each individual has a defined role and some qualifications. There are three types of individuals: the queen, the workers and the drones. The queen bee is a mated female bee and there is only one queen per hive, i.e., per colony. The queen has the main role of reproduction and cohesion, necessary to perpetuate the colony. Drones are male bees that have the main role of mating with the queen for the reproduction of the colony. Workers that are female bees as to them, are responsible for the foraging. They perform the most important task of the hive by making trips in order to supply water, pollen, nectar and other substances from blooming plants, within 3 to 4 km around the hive [72]. Workers make the link between the colony and the ambient environment.

The swarm intelligence of honeybees *Apis mellifera* foraging behavior has been studied by Karaboga [64]. The author presents three essential components on which he has built a minimal model of foraging behaviors that leads to the emergence of a collective intelligence. These components are given hereinafter:

1. The food sources. The profitability of a food source depends on many factors including its closeness to the hive, its richness or the concentration of the energy within, and the facility of extracting this energy. In order to simplify its model, Karaboga takes goodness to qualify these qualities.
2. The employed foragers. These worker bees are associated with a particular food source, which is currently being exploited until its exhaustion. They carry information about that particular food source, i.e., the direction toward the food source, its distance from the hive and the profitability of the food source. So, these foragers share these information with other foragers who remained in the hive.
3. The unemployed foragers. They are continually looking out for a food source to exploit. Unemployed foragers are in two types: scouts that are engaged in the search of new food sources around the hive, and onlookers, who wait in the hive and establish a food source through the information shared by employed foragers.

In these components, we can see that the exchange of information among bees on explored food sources is important for maintaining a collective knowledge. These information are exchanged by performing a special dance of bees called *waggle dance*. Indeed, an onlooker bee will decide by itself to employ the most profitable food source according to a certain

probability, simply by watching the dances and then choosing the best dancer source [64]. The self-organization that emerges from the foraging activity of honeybees relies on four characteristics:

1. Positive feedback: it occurs for instance when the amount of nectar in a source increases or the number of onlookers visiting a food source increases proportionally.
2. Negative feedback: the exploitation process of exhausted food sources is stopped by bees.
3. Fluctuations: the scouts perform a random search process for discovering new food sources.
4. Multiple interactions: Employed bees share their knowledge about food sources with onlookers waiting in the hive.

The aforementioned bio-inspired characteristics of honeybees are used in solving optimization problems. The well known algorithm in honeybees metaheuristic is the Artificial Bee Colony algorithm introduced by Karaboga [64].

### 2.3.3 Artificial Bee Colony Algorithm

The Artificial Bee Colony algorithm is a swarm based metaheuristic algorithm developed for numerical optimization. The ABC algorithm is based on a minimal modelling of the intelligent foraging behaviors of honeybees [64]. The proposed model consists of three essential components that are previously described, and of two leading modes. The components are three groups: employed foragers, unemployed foragers and food sources. The first are associated with particular food sources and the second are in two types: onlookers and scouts. At all times, onlookers and scouts are in the search of a rich food source to exploit. In fact, the first half of the colony consists of employed bees and the second half consists of onlooker bees. Onlookers bees are foragers who watch the *waggle dance* of employed foragers within the hive, for a decision of choosing a food source. That decision is taken according to the quality of the food source. The scout bees as for them, have a job of searching new food sources.

The self-organization and the collective intelligence in ABC are achieved by the leading modes. Thereof consists of the recruitment of foragers to rich food sources and abandoning poor food sources. According to the previously described four characteristics on which self-organization of honeybees relies, the first is assimilated to a positive feedback and the second causes a negative feedback. Concretely, the proximity and richness of a food source represents a good solution to a given optimization problem and the goodness of a food source is the fitness of the associated solution. For a global optimization problem, the

ABC algorithm consists of looking the best food source, i.e., the best solution. The ABC scheme given in *Algorithm 2.1* consists of four phases: employed bees phase, onlooker bees phase, scout bees phase and the memorization of the best solution achieved so far [64].

---

**Algorithm 2.1** Algorithm simulating the ABC

---

Begin

1. \*Initialization Phase\*
  2. Generate initial population  $X_i, i = 1, \dots, SN$
  3. **Repeat**
  4. \*Employed Bees Phase\*
  5. For each employed bee
  6.     Produce new solutions  $v_i$  according to Equation 2.1
  7.     Calculate the fitness according to Equation 2.2
  8.     Apply the greedy selection process
  9. \*Onlooker Bees Phase\*
  10. For each onlooker bee
  11.     Choose a solution  $x_i$  depending on  $p_i$
  12.     Produce new solutions  $v_i$  according to Equation 2.1
  13.     Calculate the fitness according to Equation 2.2
  14.     Apply the greedy selection process
  15. \*Scout Bees Phase\*
  16.     When solutions can't be improved  
        (reach a limit parameter), then replace it with  
        a new solution produced by a formula given by Equation 2.3
  17. \*Memorize the best solution so far\*
  18. **Until** Maximum Cycle Number
- End
- 

Firstly, scout bees have the job of searching the positions of all food sources. In the ABC algorithm, initial positions of food sources are randomly generated. Each employed bee is associated to one randomly generated food source. For simplicity reasons, the number of employed bees is equal to the number of food sources ( $SN$ ). During employed bees phase, each employed bee determines a new food source with a good amount of nectar, within the neighborhood food source according to Equation 2.1. Then, each employed bee calculates the fitness, i.e., the goodness of the food source. If the current employed bee finds that the obtained fitness value according to the formula given in Equation 2.2 is higher than the previous one, it moves to the new food source and abandons the old one, otherwise it remains on its old food source.

$$v_{ij} = x_{ij} + \psi_{ij} \times (x_{ij} - x_{kj}) \quad (2.1)$$

Where:

- $\psi_{ij} = rand(-1; 1)$ ;
- $x_k$  is a neighbor solution;
- $j \in \{1, 2, \dots, D\}$  is a randomly chosen parameter index,  $D$  is the dimension of the solution vector.

The  $i^{th}$  food source position is represented by the vector  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})^T$  and the fitness of the food source located at position  $X_i$  is  $f(X_i)$ . At the end of this phase, employed bees share their knowledge, i.e., the information on the fitness value, with the onlooker bees. Each onlooker has to select a food source according to the probability given by Equation 2.2. That probability depends on the goodness of the food source. This scheme allows good food sources to have a greater chance of being chosen.

$$p_i = \frac{f(X_i)}{\sum_{n=1}^{SN} f(X_n)} \quad (2.2)$$

When selecting their food sources, each onlooker selects a new food source within the neighborhood food source and evaluates its fitness. The onlooker will memorize the new position if the obtained fitness value is higher than the previous one and will forget the old, otherwise it keeps the current food source.

Then, food sources are exploited by employed bees and onlooker bees until their exhaustion. The best food source is looked up until reaching a certain number of cycles, i.e., the *limit parameter*. Hence, if the fitness value is not improved during the number of cycles, the food source is assigned as abandoned and the employed bee becomes a scout bee, then it is now the scout bees phase. A new solution is randomly generated by the scout according to the formula in given Equation 2.3.

$$x_{ij} = l_j + rand(0; 1) \times (u_j - l_j) \quad (2.3)$$

Where:

- $x_i$  is the abandoned food source;
- $l_j \leq x_{ij} \leq u_j$ .

### 2.3.4 Collective Decision in Honeybees Nest-sites Selection Process

Collaborative decision-making is an important goal of distributed processes. In a bee swarm, a collective decision emerges by individual actions that lead to a global intelligence over the colony. This natural feature of a bee is a very interesting tool for the

achievement of distributed self-organized sensor networks. In this section, we present honeybee behaviors and the nest-sites selection process.

The nest-sites selection process is the last and the most complex part of the house hunting process of the honeybee [73]. The selection of a site where the new hive will be established is based on a distributed non-hierarchical decision making process. Indeed, when the size of a colony (members of the same hive) becomes too large, a young queen coming from a fertilized egg is enthroned by the old queen. Therefore, the old queen, some drones and about a half of the colony leave the hive and look for a new site near or far off the old hive. Meanwhile, the young queen continues to manage the remaining bees.

Passino and Seeley [74] have defined the nest-sites selection in honeybees as a process of social decision making in which the scout bees in a swarm locate several potential nest-sites, evaluate them and select the best one by means of competitive signaling. Indeed, after leaving the hive, the new colony settles on a temporary site and from there, a number of workers called scout bees start looking for a new site. Once a scout finds a site with sufficient quality, it returns to the other bees who remained in the temporary site and performs a series of movements called waggle dance. Other bees follow these dances and may fly to the site, inspect it themselves, and dance in their turn. The dancer communicates the proposed nest-site direction and distance to her swarm mates [75]. The number and intensity of the performed waggle dances by a scout denote the quality of site that has been found. All other scouts follow the same process and a thousand of honeybees remain in the temporary site, make a collective decision for one among many discovered nest-sites [76]. The site chosen by the other bees will be that one for which the scout will have made the most waggle dance.

Passino and Seeley [74] developed a stochastic discrete time model of the nest-sites selection process that occurs on the temporary site and also the quorum sensing at the nest-site. In particular, we are interested in models of site quality assessment and dance strength determination. In fact, authors define nest-site quality assessment  $S^i(k)$  (see Equation 2.4) by  $i^{th}$  scout bee and  $L^{ij}(k)$  (see Equation 2.5) represents how the  $i^{th}$  scout bee will dance for the  $j^{th}$  site after the  $k^{th}$  expedition.

$$S^i(k) = \begin{cases} J(\theta^i(k)) + \omega^i(k) & \text{if } J(\theta^i(k)) + \omega^i(k) > \epsilon_t \\ 0 & \text{if } J(\theta^i(k)) + \omega^i(k) \leq \epsilon_t \end{cases} \quad (2.4)$$

where:

- $J(\theta^i(k))$  represents the combined overall assessment of nest-site quality for a site at location  $\theta^i$  of bee  $i$  during expedition  $k$ .
- $\omega^i(k)$  represents the noise due to errors made in the scout assessing a nest-site.

- $\epsilon_t$  is the quality or dance threshold for an acceptable site.

$$L^{ij}(k) = \begin{cases} \max\{(\gamma S^i(k_j^i) - \epsilon_s(k - k_j^i)), 0\} & \text{if } S^i(k_j^i) > \epsilon_t \\ 0 & \text{if } S^i(k_j^i) \leq \epsilon_t \end{cases} \quad (2.5)$$

where:

- $k_j^i \geq 1$  ( $k \geq k_j^i$ ) is the time index of the expedition that nest-sites explorer  $i$  first finds and assesses the quality of nest-site  $j$ .
- $\gamma$  is a parameter that represents the proportionality between nest-site quality and initial dance strength the first time that the scout bee dances for the site.
- $\epsilon_s > 0$  is a parameter used to represent the decrease in dance strength between successive expeditions.

According to these equations and others that represent the models of nest-sites selection process, Passino and Seeley [74] have studied the design of nest-sites selection process. Also, their model was validated by various simulations. We were inspired by this model for the election of CHs in our contribution to clustering sensor networks in Chapter 4.

## 2.4 Conclusion

In this chapter, we provided an overview of WSNs by highlighting some aspects of this particular type of ad hoc networks. Moreover, swarm intelligence based algorithms that promise more innovative solutions for designing distributed auto-organised WSNs have been defined and the used bio-inspired solutions for optimizing the routing in WSNs have been presented.

## Chapter 3

# Review of Cluster-based Routing Protocols

### 3.1 Introduction

Clustering sensors is usually adopted in large scale networks. Like reported by Liu [18], it is proved that the cluster-based networks, also known under the names of hierarchical or multi-tier networks, are more scalable, provide more reliability, better coverage, good fault tolerance and are energy efficient. Moreover, Moussaoui et al. [77] argue that the clustering can be seen as a graph partitioning problem with some added constraints since the size and the geometric form of the obtained graphs (clusters) is not known in advance. Thus, because of these constraints, the clustering is known to be a NP-Hard optimization problem [22]. Nevertheless, the hierarchical routing issue for WSNs has been investigated a lot and many protocols have been developed. These protocols include heuristic and meta heuristic (bio-inspired) based approaches [2, 19, 78]. In this chapter, we present a comprehensive review of the most relevant cluster-based routing protocols.

The rest of the chapter is organized as follows: Section 3.2 presents the heuristic based approaches; Section 3.3 presents the meta heuristic based approaches; A synthesis of these approaches is proposed in Section 3.4 and we conclude this chapter in Section 3.5.

### 3.2 Heuristic based Approaches

#### 3.2.1 Low Energy Adaptive Clustering Hierarchy

Heinzelman et al. [25] proposed a Low Energy Adaptive Clustering Hierarchy (LEACH), which is the most popular and widely referenced periodical data gathering cluster-based routing protocol for WSNs. The adopted completely distributed clustering mechanism



allows the random creation of CHs in order to prolong the network's lifetime. In LEACH, the network's operation is divided into rounds. Each round includes two phases, namely the setup phase and the steady-state phase. The CHs are selected and clusters are built during the setup phase and these CHs have the responsibility of aggregating and delivering the data gathered by sensors to the BS.

During the setup phase, each sensor independently decides to elect itself or not as a CH for the current round according to a certain probability. This decision is based on the suggested percentage of CHs for the network that is determined a priori and the number of times the sensor  $n$  has been a CH so far. This decision is made by the sensor by choosing a random number between 0 and 1. The sensor becomes a CH for the current round if the selected number is less than a threshold  $T(n)$  given in Equation 3.1. When a sensor is selected as a CH, it broadcasts an advertisement message to the rest of the sensors by using the CSMA MAC protocol. The non CH sensors decide which cluster they will join for this round according to the RSSI value of the advertisement. In the data transfer phase, LEACH uses the TDMA to avoid inter-cluster and intra-cluster collisions.

$$T(n) = \begin{cases} \frac{P}{1 - P \times \left( r \times \text{mod} \frac{1}{P} \right)} & \text{if } n \in G \\ 0 & \text{Otherwise} \end{cases} \quad (3.1)$$

Where  $P$  is the desired percentage of CHs. The optimal number of CHs is estimated to be about 5% of the total number of sensors, i.e.,  $P = 0.05$ ;  $r$  is the current round; and  $G$  is the set of sensors that have not been elected as CHs in the last  $\frac{1}{P}$  rounds.

Furthermore, the data routing is performed in a single-tier, i.e., in a single-hop manner. That is not realistic for large scale WSNs. It causes the quick death of sensors and therefore, it largely affects the lifetime. Otherwise, the LEACH algorithm allows the selection of a sensor to a CH position even if its remaining energy is too low. In addition, the CH status dynamically rotates between sensors in each round according to a given probability. This mechanism limits the proposal since a sensor may die right away after it is selected as a CH. Also, the adopted dynamic scheme for the construction of clusters causes extra overheads.

Unfortunately, since the selection of CHs is randomly done, the distance between a given sensor and its corresponding CH may be too long. This causes a weak distribution of CHs over the sensing area and hence a lot of power wastage. de Fr in et al. [36] introduced efficient CH trajectories for collaborative mobile sensing. The proposed approach is similar to LEACH by considering the challenges presented in the use of smartphones as a sensing platform. Their proposal increases the power lifetime associated with sensor data transmissions of the mobile handsets in a LEACH-like scenario by 20-40% without incurring a sensing accuracy penalty.

### 3.2.2 Energy Efficient Hierarchical Clustering

Bandyopadhyay and Coyle [79] proposed an Energy-Efficient Hierarchical Clustering (EEHC) algorithm for WSNs, which is a distributed randomized clustering algorithm designed to organize sensors into clusters. The EEHC protocol assumes that communication environment is contention and error free and does not demand clock synchronization between the sensors. Moreover, the CHs in EEHC collect the gathered data from the sensing nodes and transmit an aggregated report through the hierarchy of CHs to the BS. The EEHC algorithm allows the generation of a hierarchy of CHs and in order to increase the energy saving. To become a CH, each sensor in the network has to compute the probability  $p$  given in Equation 3.2 and send an announcement message to indicate itself as a CH named volunteer CH, to the sensors within its neighborhood. The announcement message is forwarded to all the sensors that are located not more than  $k$  hops away from the CH. The value of  $k$  is given by the formula given in Equation 3.3.

$$p = \left( \frac{1}{3c} + \frac{\sqrt[3]{2}}{3c \times \sqrt[3]{(2 + 27c^2 + 3\sqrt{3c}\sqrt{27c^2 + 4})}} + \frac{\sqrt[3]{(2 + 27c^2 + 3\sqrt{3c}\sqrt{27c^2 + 4})}}{3c} \times \frac{1}{\sqrt[3]{2}} \right)^2 \quad (3.2)$$

Where  $c = 3.06a\sqrt{\lambda}$ ;  $2a$  denotes the length of a side of the sensing square; and  $\lambda$  is the intensity of the Poisson process.

$$k = \left\lceil \frac{1}{r} \times \sqrt{\frac{-0.917 \times \ln(\frac{\alpha}{7})}{p\lambda}} \right\rceil \quad (3.3)$$

Where  $r$  is the radio range of each sensor and  $\alpha$  is a parameter to be properly chosen.

Any sensor that is not itself a CH that receives such an announcement message joins the closest CH. Since the forwarding of the announcement message by sensors is limited to  $k$  hops, when a sensor does not receive an announcement message during a given time  $t$  that is a defined time required for data to reach a CH from any sensor  $k$  hops away, it becomes a CH named a forced CH. Furthermore, the limit on the number of hops thus allows CHs to well schedule their transmissions. In fact, since all sensors within a cluster are at the worst cases  $k$  hops away from the CH, the CH should transmit the aggregated data to the BS after every  $t$  time. Hence, the energy consumed in the network for the information gathered by the sensors to reach the processing center will depend on two parameters:  $p$ , which is the probability of a sensor to become a CH at each level in CHs hierarchy and  $k$ , which is the maximum number of hops allowed between a sensor and its corresponding CH. These optimal clustering parameters are obtained through hierarchical clustering in order to minimize the overall amount of consumed energy.

Unfortunately, the hierarchical CHs approach adopted in the design of the EEHC allows

CHs to consume relatively more energy than other sensors since CHs have more data and communication to handle. This leads to battery run out in CH sensors faster than in sensing nodes. Thus, the EEHC algorithm can be run periodically for load balancing or triggered as the energy levels of the CHs fall below a certain threshold [80].

### 3.2.3 Hybrid Energy-Efficient Distributed clustering

Younis and Fahmy [26] have extended the clustering mechanism of the LEACH by introducing the Hybrid Energy-Efficient Distributed clustering (HEED). The proposed protocol aims at providing an energy-efficient clustering protocol with explicit consideration of energy. The HEED was designed with four primary goals: prolonging network lifetime by balancing the energy consumption; terminating the clustering process within a constant number of iterations; minimizing and controlling the overhead; and producing well-distributed CHs and compact clusters. To achieve these goals, the HEED protocol uses two parameters: the residual energy and the sensor degree or density. The first parameter has been introduced to increase the energy efficiency and hence prolong the network's lifetime. The second parameter deals with the cost of the intra-cluster communication based on the density of sensors in a considered neighborhood, in order to allow a power balancing in the selection of CHs.

The HEED algorithm periodically selects CHs probabilistically by calculating the probability of becoming a CH that includes the residual energy and by evaluating the cost of the intra-cluster communication, a combination of two clustering parameters. Like in the LEACH, an initial percentage of CHs in the network  $C_{prob}$ , is predefined. To become a CH, each sensor evaluates the probability  $CH_{prob}$  with the formula given in Equation 3.4.

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \quad (3.4)$$

Where  $E_{residual}$  is the estimated current residual energy in this sensor and  $E_{max}$  is the maximum energy corresponding to a fully charged battery, which is typically identical for all sensors in the network.  $CH_{prob}$  value is not allowed to be lower than a certain threshold  $p_{min}$  that is selected to be inversely proportional to  $E_{max}$ . In other terms, the  $CH_{prob}$  value must be greater than a minimum threshold  $p_{min}$ .

Typically, there are two types of status that a sensor could announce to its neighborhood: tentative status and final status. A CH is either a tentative CH, if its  $CH_{prob} < 1$ , or a final CH, if its  $CH_{prob} = 1$ . During each round, if a sensor never heard from a CH, it elects itself to become a CH according to the probability  $CH_{prob}$  and it sends an announcement message to its neighborhood. In general, if a sensor is selected to a CH candidate position, it broadcasts an announcement message as a tentative CH or a final CH. A given sensor that hears the CH list, has to join the CH that computed the lowest communication cost.

Then, each node doubles its  $CH_{prob}$  and goes to the next iteration until its  $CH_{prob}$  value reaches 1. If the  $CH_{prob}$  value of a sensor has reached 1, that sensor permanently becomes a CH.

In HEED, the balancing of energy consumption extends the lifetime of all the whole network, thereby sustaining stability of clusters. Contrary to the LEACH, CHs in HEED send the aggregated data to the BS in a multi-hop manner. In addition, cluster members automatically update their neighborhood in multi-hop networks by periodically sending and receiving messages. However, in certain cases, more CHs are generated in the HEED than the expected number and this fact leads to an unbalanced energy consumption [81]. The clustering mechanism adopted in the HEED improves network lifetime compared to the LEACH since the LEACH randomly selects CHs without energy level consideration that resulted in faster death of sensors. Unfortunately, because of a great number of iterations in its execution, the HEED algorithm generates maximum overheads.

### 3.2.4 Clustering Algorithm for Efficient Energy Saving in WSNs

Moussaoui et al. [77] proposed a clustering algorithm for efficient energy saving that uses a distributed approach to setup non-overlapping clusters while performing CHs rotation as well as carrying out other energy intensive tasks. The proposed algorithm combines parameters such as the cluster size, the transmission power and the energy level of sensors with certain weighting factors properly chosen according to the system needs. The proposal allows only sensors with a sufficient energy level to be selected as CHs while those with low energy prolong their lifetime by performing tasks that require low energy costs. The execution of the proposed algorithm is performed in three phases namely the cluster setup, CHs election and the schedule creation of CHs. At this point, the cluster formation takes place in five steps:

1. Each sensor  $s_i$  constructs its neighbor set  $NS(s_i)$  that is the set of sensors within its transmission range  $tx_{range}(s_i)$ , like given in Equation 3.5

$$NS(s_i) = \{s_j \in V \mid d(s_i, s_j) < tx_{range}(s_i)\} \quad (3.5)$$

Where  $V$  represents the set of sensors and  $d(s_i, s_j)$  is the euclidean distance between sensors  $s_i$  and  $s_j$ .

2. Each sensor  $s_i$  computes its equivalency classes  $EC(s_i)$  (see Equation 3.6), which is the set of sensors that belong to  $NS(s_i)$ . Two sensors  $s_j$  and  $s_k$  belong to the same  $EC(s_i)$  if  $s_j$  and  $s_k$  are neighbors to each other.

$$EC(s_i) = \{s_j \in NS(s_i) \mid \forall s_k \in EC(s_i) \Rightarrow s_j \in NS(s_k)\} \quad (3.6)$$

3. For each equivalency class  $EC$ , each sensor computes the combined weight  $W_{EC}$ , like defined in Equation 3.7.

$$\begin{aligned}
 W_{EC(s_i)} = & \alpha \cdot |Card(EC(s_i)) - m| + \frac{2 \cdot \beta}{\left( (Card(EC(s_i)))^2 - Card(EC(s_i)) \right)} \times \sum_{s_j, s_k \in EC(s_i)} d(s_j, s_k) \\
 & + \gamma \cdot \sum_{s_j \in EC(s_i)} \frac{1}{C_e(s_j)}
 \end{aligned} \tag{3.7}$$

Where  $\alpha$ ,  $\beta$  and  $\gamma$  are the weight factors;  $Card(EC(s_i))$  is the size of  $EC(s_i)$ ;  $m$  is a defined number that represents the maximum number of sensors within a cluster and  $C_e(s_j)$  is the actual energy level of sensor  $s_j$ . This combined weight metric  $W_{EC(s_i)}$  is composed by different components reflecting the number of sensors enclosed by the cluster, the distance among sensors and the energy level of each sensor.

4. Choose equivalency class with the smallest  $W_{EC(s_i)}$  as cluster.
5. Repeat the steps 2-4 for the remaining sensors that do not belong to any cluster yet.

After the clustering procedure, sensors are grouped into clusters. Each sensor belongs to only one cluster. Then, each cluster is represented by a CH, which is elected from the nodes set. Thereby, cluster members communicate each other only through their CH. Since CHs perform functions that consume more energy, it is important to re-elect. Schedule creation is the last major issue related to the setup phase. The proposed algorithm utilizes the TDMA scheduling scheme to minimize collisions among sensors. Simulation results proved that the number of clusters decreases with the increase in both the cluster size threshold and the transmission range. However, the average latency increases as the cluster size threshold increases. Thus, a high system throughput can be achieved by limiting or optimizing the size of each cluster.

### 3.2.5 Hierarchical Energy Efficient Routing Protocol

Nesrine and Ben Jemaa [82] proposed the Hierarchical Energy Efficient Routing Protocol (HEERP) that aims at designing a hierarchy-based multi-path and multi-hop routing algorithm, which guarantees the fiability, simplicity and energy efficiency in order to prolong the network lifetime. The HEERP algorithm allows the formation of hierarchical relations, where sensors can build autonomous relationships without any centralized control mechanism such as cluster and CHs selection. The steps involved in the HEERP algorithm include the construction of the network hierarchy and neighbor tables, data transmission and network maintenance.

The first step enables the hierarchy setup and the neighbor table construction for each sensor. To achieve this, the BS initiates formation of hierarchy by broadcasting an LCREQ packet. Other sensors choose LCREQ packets from sensors with lesser hop count. In this way, sensors store records and keep flooding the packet until the network is constructed. During this process, each sensor keeps in its Parents-information-table the identities of the parent sensors with the less hop towards the BS. After the network construction step, each sensor is ready to send data to its parents. During the data transmission a given sensor risks to run out its battery or to be damaged due to some environmental factors. Therefore, the sensor may lose its path towards the BS and thus data loss will occur. To overcome this problem, the HEERP algorithm introduces some control to ensure reliability. The maintenance step ensures the routing availability by removing entry of faulty sensors. Compared to the LEACH, the HEERP consumes less energy while improving the delay in the packet delivery to the BS.

### 3.2.6 Energy-Aware Clustering Algorithm

Yu et al. [83] proposed a cluster-based routing protocol for WSNs with nonuniform node distribution, which includes an Energy-Aware Clustering Algorithm (EADC) and a cluster-based routing algorithm. The EADC algorithm constructs clusters of even sizes using competition range ( $R_c$ ) in order to well balance the energy consumption among cluster members. The problem of imbalanced energy consumption among CHs due to the nonuniform node distribution is solved by a cluster-based routing algorithm, which disseminates data forwarding task of densely spaced CHs to sparsely spaced CHs with higher energy to balance the energy consumption among CHs by adjusting the energy consumed in the intra-cluster and inter-cluster communication. The proposal supports both multi-hop and single-hop transmission by CHs. The whole process of the proposal is divided into four phases namely the information collection phase, the CH competition phase, the cluster formation phase and the cluster-based routing algorithm.

During the information collection phase each sensor broadcasts within its radio range  $r$ , a *Node\_Msg* containing the sensor *id* and its residual energy. In the same manner it receives the *Node\_Msg* messages from its neighborhood, according to which, each sensor  $s_i$  calculates the average residual energy  $E_{ia}$  of its neighbors according to the formula given in Equation 3.8. Also, each sensor calculates its waiting time  $t_i$  for broadcasting *Head\_Msg*, according to the formula given in Equation 3.9.

$$E_{ia} = \frac{1}{d} \times \sum_{j=1}^d E_{jr} \quad (3.8)$$

Where  $E_{jr}$  denotes the residual energy of  $s_j$  that is a neighbor of sensor  $s_i$  and  $d$  is the

number of all neighbors of sensor  $s_i$ .

$$t_i = \begin{cases} T_2 \cdot V_r \times \frac{E_{ia}}{E_{ir}} & \text{if } E_{ir} \geq E_{ia} \\ T_2 \cdot V_r & \text{if } E_{ir} < E_{ia} \end{cases} \quad (3.9)$$

Where  $T_2$  is the duration of the CH competition phase,  $E_{ir}$  is the residual energy of  $s_i$  and  $V_r$  is a real value uniformly distributed in  $[0.9, 1]$ , which is introduced to reduce the probability that two sensors send *Head\_Msg* at the same time.

During the CH completion phase, if sensor  $s_i$  receives no *Head\_Msg* messages when timer  $t_i$  expires, it broadcasts the *Head\_Msg* within radio range  $R_c$  to advertise that it will be a CH. Otherwise, it gives up the competition. In the cluster formation phase each non-CH sensor chooses the nearest CH and sends the *Join\_Msg* that contains the *id* and residual energy of this sensor. According to the received *Join\_Msg* messages, each CH creates a sensor schedule list including the *Schedule\_Msg* for its cluster members, the *Schedule\_Msg* is used for telling the cluster members when they can transmit their data to the CH and in other time interval they can alter their state to asleep in order to reduce the energy consumption.

After the formation of clusters, each CH constructs a routing tree among the elected CHs. Therefore, each CH determines whether to communicate directly with the BS or in a multihop manner, based on the threshold value *DIST\_TH*. In fact, if the distance from the CH  $s_i$  to the BS is less than *DIST\_TH*,  $s_i$  communicates with the BS directly, and sets the BS as its next hop. Otherwise, the CH  $s_i$  communicates with the BS through its next-hop CH  $s_j$  based on an indicator called  $relay(s_i, s_j)$  given in Equation 3.10. Then, each CH chooses the neighbor CH with the largest relay value and closer to the BS as its next hop. Simulation shows that CHs in EADC are evenly distributed across the sensing area either in the case of randomly deployed or non-uniformly deployed sensing area because of the completion range  $R_c$ . The proposal significantly improves the network lifetime.

$$relay(s_i, s_j) = \alpha \times \frac{E_{jr}}{E_{max}} + (1 - \alpha) \times \frac{1}{S_{j(cm-num)}} \quad (3.10)$$

Where  $E_{jr}$  is the residual energy of cluster head  $s_j$ ,  $E_{max}$  as the maximum initial energy of sensors in the network,  $S_{j(cm-num)}$  is the number of cluster members of  $s_j$  and  $\alpha$  is a real value uniformly distributed in  $[0, 1]$ .

### 3.3 Meta Heuristic based Approaches

#### 3.3.1 Low Energy Adaptive Clustering Hierarchy - Centralized

Unlike the LEACH that is a completely distributed protocol, the LEACH-C [27], which is an amelioration of the LEACH, uses a centralized control algorithm for the selection of CHs and the formation of clusters. At the start of each network round, i.e., during the setup phase, when receiving sensors' locations and their energy levels during the network initialization step, the BS launches the clustering process. To ensure even distribution of the energy load among all the sensors in the network so that there are no overly-utilized sensors that will run out of energy before the others, the BS computes an average energy level and whichever sensors have energy below this average, cannot be eligible to become CHs. The clusters are chosen to minimize the amount of energy for the non-CH sensors to transmit their data to the CH, by minimizing the total sum of squared distances among all the non-CH and the closest CH according to formula given in Equation 3.11.

$$f = \sum_{i=1}^N \min_{s_k \in K} \left( d^2(s_i, s_k) \right) \quad (3.11)$$

Where  $N$  is the number of sensors;  $K$  is the set of CHs; and  $d(s_i, s_k)$  is the euclidean distance between the sensor  $s_i$  and the CH  $s_k$ .

The clustering algorithm in the LEACH-C uses the Simulated Annealing (SA) approach with sensors' locations and their available power information to build a predetermined number of CHs and so, a fixed number of clusters, each with a certain number of member. In addition, clusters are built such a way that the amount of energy consumed by cluster members in the transmission of their gathered data to their corresponding CH is minimized. Each sensor should choose to become a CH at round  $r$  with probability  $P_i(t)$  given in Equation 3.12.

$$P_i(t) = \begin{cases} \frac{k}{N - k \times \left( r \times \text{mod} \frac{N}{k} \right)} & \text{if } C_i(t) = 1 \\ 0 & \text{if } C_i(t) = 0 \end{cases} \quad (3.12)$$

Where  $r$  is the current round;  $k$  is the expected number of CHs for the current round; and  $C_i(t)$  is the indicator function determining whether or not a sensor  $s_i$  has been a CH in the most recent  $(r \times \text{mod} \frac{N}{k})$  rounds.  $C_i(t) = 0$  if the sensor  $s_i$  has been a CH and  $C_i(t) = 1$  otherwise.

Furthermore, once CHs and their associated clusters are found, the BS broadcasts a message that contains the CH  $id$  for each cluster members. If a CH  $id$  matches its own  $id$ , the sensor is a CH; otherwise, the sensor determines its TDMA slot for data transmission



and goes to sleep until it is time to transmit data. The steady-state phase of LEACH-C is identical to that of LEACH [25]. The LEACH-C approach produces a better positioning of CHs on the sensing area, it slightly reduces the power consumption and data loss compared to LEACH. But the adopted single-hop routing mechanism remains a limitation. However, the LEACH-C protocol improves the LEACH in terms of data delivery at the BS and power consumption.

### 3.3.2 Energy-aware Clustering for WSNs using the PSO Algorithm

Latiff et al. [22] used the Particle Swarm Optimization (PSO) algorithm [62], which is based on social behaviors of bird flocking or fish schooling to design the PSO-C protocol. The proposed protocol implements a centralized clustering algorithm at the BS. It is supposed that, each sensor knows its location. In order to assign sensors to a cluster, the BS, i.e., the clustering algorithm, considers the energy level and the euclidean distance between a sensor and selected CHs. The PSO-C protocol operates in rounds. Each round begins with a setup phase at which clusters are built, followed by a steady state phase.

During the setup phase, sensors send information about their current energy levels and their locations to the BS. Based on these information, the BS computes the average energy level of all sensors. Only sensors with an energy level above the average are eligible to be candidates for a CH position for the current round. A defined cost function that aims at minimizing the intra-cluster distance while optimizing the power consumption of sensors is adopted. This allows the selection of the best set of CHs that can maximize the cost function. Then, the BS runs the PSO algorithm to determine the best  $K$  CHs that can minimize the cost function given in Equation 3.13.

$$\begin{aligned}
 f &= \beta \times f_1 + (1 - \beta) \times f_2 \\
 f_1 &= \max_{1 \in [1, K]} \left( \sum_{s_i \in C_{p,k}} \frac{d(s_i, CH_{p,k})}{|C_{p,k}|} \right) \\
 f_2 &= \frac{\sum_{i=1}^N E(s_i)}{\sum_{k=1}^K E(CH_{p,k})}
 \end{aligned} \tag{3.13}$$

Where  $\beta$  is a user-defined weight factor;  $N$  is the number of sensors;  $|C_{p,k}|$  is the number of sensor that belong to cluster  $C_k$  of particle  $p$ ;  $f_1$  is the maximum average euclidean distance of sensors and to their associated CHs; and  $f_2$  is the ratio of total initial energy of all sensors with the total current energy of CHs' candidates in the current round.

Therefore, when the BS has identified the optimal set of CHs and their associated cluster members, it informs each sensor the cluster and the *id* of the CH in which it belongs. The sensors that becomes a CH acts as a local control centre to coordinate the data transmission in its cluster. The CH sets up a TDMA schedule for its members to avoid

collisions among data messages, allowing the radio devices of each member to be turned off at all times, except during their transmission time, to further reduce sensors' energy consumption. When receiving all gathered data by its cluster members, the CH aggregates and transmits the aggregated data to the BS by using a fixed spreading code and CSMA approach.

Authors compared the PSO-C protocol with the above described LEACH and LEACH-C protocols. The results show that PSO-C outperforms the compared protocols in terms of network lifetime and throughput. In their performance comparison, authors found that the PSO-C protocol outperforms the Genetic Algorithm (GA) and K-means based clustering protocols in terms of the convergence of the defined cost function, the data delivery and the network lifetime. In spite of the promising performance outputted by this swarm intelligence based algorithm, the PSO-C protocol assumes that sensors know their location. This fact seems not realistic for randomly deployed sensors and thus, non-scalable.

### 3.3.3 Cluster-based WSN Routing using the ABC Algorithm

The ABC algorithm [64], which is inspired by foraging behaviors of honeybee swarms has been successfully used to design clustering algorithms [13, 23]. Karaboga et al. [23] proposed the ABC-C protocol, a cluster-based routing algorithm having no global positioning system, inspired by the ABC metaheuristic. The clustering mechanism of the proposed protocols is based on the clustering technique of the LEACH [25] protocol where CHs perform data aggregation. CHs use TDMA MAC in intra-cluster communication and CDMA MAC communication with the BS. The clustering and routing strategy is performed at the initialization phase, as a centralized control mechanism at the BS and information messages about the configuration and cluster organization are broadcast over the network. At this stage, information about the distances between all sensors and their energy status are gathered. Indeed, to obtain the values of distances, sensors send advertisement messages to the network. When receiving these advertisement messages at various signal strengths, each sensor computes distances  $d_{ij}$  between sensors  $s_i$  and  $s_j$  according to the formula given in Equation 3.14.

$$d_{ij} = c \times \frac{\sqrt{P^s}}{4\pi \cdot f} \times \frac{1}{\sqrt{P^r}} \quad (3.14)$$

Where  $f$  is the communication frequency;  $c$  is the speed of the light;  $P^r$  is the received signal strength; and  $P^s$  is the sender signal strength.

After receiving and calculating cross-distance values, sensors send these values to the BS. These information are used in the selection process, which is achieved with a centralized program using the ABC algorithm. In this algorithm, the selection process of CHs is

achieved using the fitness function obtained analytically in which the communication energy is considered as the significant factor. In the first step, cluster organization is made by selecting the CHs for the current round. Then, sensors are joined to the nearest CHs. After this selection process, periodical data from the network is gathered by cluster members and sent to CHs as the second step. Similar to the LEACH approach, the ABC-C performs the election of CH periodically. This choice introduces overhead communication in the network. However, because in their proposed semi-distributed approach, the CH election step is achieved in a centralized manner by the BS, the energy cost of the CH election process is minimized on sensor nodes.

### 3.3.4 Multi-tier Cluster-based Routing Protocol for Sensor Networks

da Silva Rego et al. [84] introduced the Bee-C that is a multi-tier cluster-based routing protocol for sensor networks. Based on LEACH-C algorithm, the Bee-C is a bio-inspired algorithm inspired by the Honey Bee Mating Optimization (HBMO) [85], which is based on the reproduction behaviors of honeybees. Like LEACH [25] and LEACH-C [27] protocols, the main idea in the Bee-C design is the energy saving. To achieve this goal, authors designed a network with a continuous data dissemination, principally for the energy saving purpose. Besides the use of a metaheuristic algorithm, Bee-C implements new objective functions that aim at optimizing the energy consumption. The Bee-C protocol works in rounds, the time of each round is defined as a parameter of the protocol. Each round is composed of two phases: cluster formation and data transfer.

The construction of CHs phase is carried out similarly to LEACH-C, where each sensor sends its location and its amount of energy to the BS. According to the data, the BS computes the average consumed energy. Only the sensors with energy level over the average are eligible to become CHs. To find the best clusters, the BS performs the HBMO-based algorithm, to minimize or maximize the objective functions of the algorithm. The first objective function (*see Equation 3.15*) is calculated by obtaining the sum of the averages of the remaining energy of sensors in each cluster. The second objective function given in Equation 3.16 is the combination of three sub-objective functions where one of the sub-objective functions is the objective function of the LEACH-C algorithm.

$$F_1 = \sum_{k=1}^p \frac{E_{T_k} - E_{C_k}}{N_k} \quad (3.15)$$

Where  $p$  is the number of clusters;  $E_{T_k}$  is the total energy of the cluster  $k$ ;  $E_{C_k}$  is the total spent energy by the cluster  $k$ ; and  $N_k$  is the number of sensors in the cluster  $k$ .

$$F_2 = \left( \sum_{i=1}^N \min_{s_k \in K} (d^2(s_i, s_k)) \right) \times \left( \frac{\sum_{i=1}^N E(s_i)}{\sum_{k=1}^K E(s_k)} \right) \times \left( \sum_{k=1}^K d(d(s_k, BS)) \right) \quad (3.16)$$

Where  $K$  is the set of CHs;  $E(s_i)$  is the energy of the sensor  $s_i$ ;  $E(s_k)$  is the energy of the CH  $k$ ;  $d(s_i, s_k)$  is the euclidean distance between sensor  $s_i$  and the CH  $s_k$ ; and  $d(s_k, BS)$  is the euclidean distance from the CH  $k$  to the BS.

The Simulation of the Bee-C showed the gains with the two objective functions, in regards to the number of packets sent to the BS, the network lifetime and the total network coverage time. Unfortunately, while increasing the number of sensors, the Bee-C protocol delivers poor results, it simply means that the protocol is not scalable.

### 3.3.5 Energy-Efficient Clustering and Routing Algorithms for WSNs

A semi-distributed clustering strategy based on Non Linear (NLP) and linear Programming (LP) formulations of the clustering and routing problem is proposed by Kuila and Jana [16]. These problems are solved by proposing algorithms based on the PSO. The routing algorithm is developed with a trade-off between transmission distance and number of data forwarded with an efficient particle encoding scheme and multi-objective fitness function. The clustering algorithm is presented by considering energy conservation of sensors through load balancing.

The proposal considers a WSN model where all the sensors are randomly deployed along with a few gateways and once they are deployed, they become stationary. A sensor can be assigned to any gateway if it is within the communication range of the sensor. The network setup is performed in three phases namely: the bootstrapping, the route setup and the clustering. During the bootstrapping process, all sensors and gateways are assigned unique identifiers. Then all sensors and the gateways broadcast their identifiers using CSMA/CA MAC layer protocol. Therefore, the gateways can collect the identifiers of the sensors and the other gateways which are within their communication range and finally send the local network information to the BS. Now, using the received information of the network, the BS executes the routing and clustering algorithm. Then, all the gateways are informed about their next hop relay sensor towards the BS and the sensors are also informed about the *id* of the gateway they belong to. Then the gateways provide a TDMA schedule to their sensor members for intra cluster communication. Gateways use slotted CSMA/CA MAC protocol to communicate with its next hop relay sensor.

The routing problem is solved by minimizing the maximum transmission distance between two sensors in the routing path and maximum hop count. The first objective (*see Equation 3.17*) is to minimize the maximum distance ( $Max_{Dist}$ ) between two sensors and the second objective (*see Equation 3.18*) is to minimize the maximum number ( $Max_{Hop}$ ) of hops used by the gateways.

$$Max_{Dist} = \max_{1 \leq i \leq M} d(g_i, NextHop(g_i)) \quad (3.17)$$

Where  $g_i$  is the  $i^{th}$  gateway;  $NextHop(g_i)$  denotes the next gateway towards the BS; and  $d(g_i, NextHop(g_i))$  is the distance between the gateway  $g_i$  and its subsequent.

$$Max_{Hop} = \max_{1 \leq i \leq M} (HopCount(g_i)) \quad (3.18)$$

Where  $HopCount(g_i)$  denotes the number of next hops required to reach to the BS from  $g_i$ .

The proposed fitness function enables a good balancing of consumed energy by CHs. Like the approach presented in [23], the cluster formation is done in a centralized manner. The proposed algorithms improve the network lifetime, but it seems not realistic for randomly deployed environments.

### 3.3.6 PSO protocol for Hierarchical Clustering in WSNs

Elhabyan and Yagoub [29] proposed a centralized PSO based protocol for hierarchical clustering (PSO-HC) in WSNs, which uses a realistic energy model. The clustering algorithm of the PSO-HC protocol is implemented at the BS. The PSO-HC includes a PSO-based clustering protocol with a trade-off between energy efficiency, network coverage and data transmission reliability; and a PSO-based routing protocol with a novel particle encoding scheme for complete routing tree solution and derivation of efficient multi-objective fitness function.

In the PSO-HC, each round consists of two phases, the setup phase and the steady state phase. In the setup phase, the network is configured. Indeed, the setup phase consists of a Neighbor discovery and transfers the collected information on the network to the BS. Based on the information the BS received, the BS will compute the average energy level of all sensors. Only sensors with an energy level above the average are eligible to be candidates to a CH position for this round to ensure that only nodes with sufficient energy are selected as CHs. The BS will use these information to find the optimal set of CHs and their associated cluster members as well as relay nodes and optimal routes from those CHs to the BS.

The clustering mechanism takes into consideration the sensor's received signal strength and its energy level. The received signal strength helps in the evaluation of the communication link quality between two sensors. To improve the scalability, a two-tier cluster is adopted in such a way that the coverage is maximized. A multi-objective fitness function in both the clustering and routing problem is considered in the PSO-HC.

Like PSO-C [22], LEACH [25] and LEACH-C [27] protocols, the network operating time is splitted into rounds in which CHs are selected. In the steady-state phase, each non-CH sensor uses its TDMA schedule to transmit its data to its respective CH. When a CH receives this data, it uses its next relay sensor to forward the data to the BS. When

a non-CH sensor finishes its data transmission slot, it enters the sleep state to save its energy.

The PSO metaheuristic is exploited in the selection of a minimal number of active CHs. In fact, since the whole coverage of the network is not well guaranteed in the case of a fixed number of CHs, authors consider a predetermined percentage of CHs according to the network density. The performance analysis shows that PSO-HC outperforms PSO-C, LEACH-C and LEACH protocols in terms of the average consumed energy and the throughput.

### 3.3.7 Energy-efficient and Scalable Multipath Routing Protocol for WSNs

Cai et al. [21] proposed an amelioration of Bee-C called Bee-Sensor-C. Principally designed for event driven sensor networks, the Bee-Sensor-C protocol uses a dynamic clustering scheme in order to reduce the routing overhead and improves the scalability. Authors adopted an on-demand multipath cluster-based routing mechanism inspired by some foraging behaviors of a bee swarm. Indeed, the Bee-Sensor-C builds a cluster structure and selects the CH when an event occurs. Bee-Sensor-C is mainly divided into three phases namely: the cluster formation, the multipath construction and the data transmission. The first phase is to build cluster structure when an event happens. The second phase consists of the construction of multipath between a CH and relative sink, followed by carrying data to sink through the stochastically selected path. The last phase consists of the route maintenance.

Concretely, when an event occurs in a place of the sensing environment, all sensors close to that event may need to gather and transfer the data. So, the first sensor that declares the event becomes the CH and other sensors have to follow him and by this way, a cluster is constructed. The multipath construction between CH and destination sensor is done by computing a reward value according to the formula given in Equation 3.19.

$$R_{jp} = \frac{w_1 E_{avg} + (1 - w_1) E_p^j}{H_{jp}} \quad (3.19)$$

Where  $E_{avg}$  is the average remaining energy of the path;  $w_1$  is the control parameter and  $H_{jp}$  is the number of hops to reach the sensor  $s_p$  from sensor  $s_j$ . At the last phase, when the backward scout with unique path  $id$  arrives at the CH, it recruits foragers and data is transmitted through the stochastically selected path by computing the dance number which represents the quality of the path according to the formula given in Equation 3.20.

$$DN_{pid} = \left\lceil \frac{(\beta - H_{pid} \times (w_2(E_{initial} - E_{avg}) + (1 - w_2)E_{min}))}{\gamma} \right\rceil \quad (3.20)$$

Where  $w_2$  is the weight value;  $\gamma$  and  $\beta$  are user-defined constants;  $E_{initial}$  is the initial

energy of sensor;  $E_{min}$  is the minimum residual energy in the path,  $E_{avg}$  is the average residual energy of sensors in the path; and  $H_{pid}$  is the hop count of the path.

However, the adopted on demand cluster construction will introduce a great overhead in a case of high event-driven and large-scale networks. Nevertheless, compared to Bee-C, Bee-Sensor-C improves the whole network performance.

### 3.4 Synthesis

The aforementioned protocols and many others in the literature excepting those in [13, 29], adopted the first order energy model initially proposed by Heinzelman et al. [27] for the LEACH-C protocol. Of course, it is the first stable theoretical modelling of energy consumption for radio sensors, but that model seems not realistic. Indeed, in this energy model, authors haven't taken into account the energy of listening that is known to be non-negligible in power discharging of a wireless sensor battery [7]. Moreover, similar to the previous PSO-C protocol, most of the existing clustering and routing protocols suppose that sensors know their locations. This implicitly implies that sensors are equipped with localization hardware. It is not realistic for large scale networks and also it leads to the use of costly sensors. Furthermore, that energy model is essentially based on the distance between the sender and the receiver for evaluating the link quality. Yet, a number of research demonstrated that the distance is not a valid criteria for the evaluation of the quality of a link in a WSN. The most convincing work that justifies this assertion is the empirical study of low-power wireless achieved by Srinivasan et al. [6].

Table 3.1 presents a comparison of the aforementioned clustering protocols.

### 3.5 Conclusion

In this chapter, we provided a comprehensive review of the most relevant heuristic and metaheuristic cluster-based routing protocols. Some of them will be used principally for the comparison purpose with contributions presented in Chapter 4 and Chapter 5.

Table 3.1: Synthesis of cluster-based protocols

Clustering protocol	method	approach	CHs <sup>1</sup>	Energy model	Link quality based on	Location awareness	Connectivity to the BS	Network type	EE <sup>2</sup>	TH <sup>3</sup>	SC <sup>4</sup>	LQ <sup>5</sup>
LEACH [25]	Distributed	Prob./Random	Variable	First order	Distance	No	Single-hop	Ho. <sup>7</sup>	✓	✗	✗	✗
EEHC [79]	Distributed	Prob./Energy	Fixed	First order	Distance	No	Single-hop	Ho./He. <sup>8</sup>	✓	✗	✗	✗
HEED [26]	Distributed	Prob./Energy	Variable	First order	Distance	No	Multi-hop	Ho.	✓	✗	✗	✗
CAEES [77]	Distributed	Link/Energy	Variable	First order	Distance	No	Single-hop	Ho.	✓	✓	✗	✗
HEERP [82]	Distributed	Link	-	First order	Distance	No	Multi-hop	Ho.	✓	✗	✓	✗
EADC [83]	Distributed	Prob./Energy	Variable	First order	Distance	No	Multi-hop	Ho./He.	✓	✓	✗	✗
LEACH-C [27]	Centralized	SA	Fixed	First order	Distance	Yes	Single-hop	Ho.	✓	✗	✗	✗
Bee-C [84]	Centralized	HBMO	Fixed	First order	Distance	Yes	Multi-hop	Ho.	✓	✗	✗	✗
PSO-C [22]	Centralized	PSO	Fixed	First order	Distance	Yes	Single-hop	Ho./He.	✓	✗	✗	✗
ABC-C [23]	Centralized	ABC	Fixed	First order	Distance	No	Single-hop	Ho./He.	✓	✓	✗	✗
EECRA [16]	Centralized	PSO	Fixed	First order	Distance	No	Multi-hop	Ho./He.	✓	✓	✗	✗
PSO-HC [29]	Centralized	PSO	Fixed	CC2420	RSSI	No	Multi-hop	Ho./He.	✓	✗	✓	✓
Bee-Sensor-C [21]	Distributed	PSO	Fixed	First order	RSSI	No	Multi-hop	Ho.	✓	✗	✓	✗

<sup>1</sup>The number of CHs<sup>2</sup>Energy Efficiency<sup>3</sup>Throughput<sup>4</sup>Scalability<sup>5</sup>Link Quality<sup>6</sup>Probabilistic<sup>7</sup>Homogeneous<sup>8</sup>Heterogeneous





**Part II**

**Contributions**



## Chapter 4

# A Self-organized Clustering Algorithm for WSNs: the honeybee swarms nest-sites selection process based approach

### 4.1 Introduction

In this chapter, we proposed a distributed clustering approach based on the nest-sites selection process of a honeybee swarm. In the design of our proposed algorithm called NEST, we focused on the distribution of load balancing among each cluster member in order to extend the network lifetime. The performance of the proposed approach was evaluated by performing extensive experiments and the results demonstrated that our algorithm delivers better performance in terms of network lifetime, energy consumption and amount of packets received by the base station. The main contributions can be summarized as follows: social decision-making process formulation of the clustering problem; proposition of a cost-based fitness function; proposition of a clustering algorithm that includes a tradeoff between the energy consumption and the quality of the communication link; simulation of the NEST to demonstrate its performance compared to some existing protocols.

The rest of the chapter is organized as follows: some assumptions and adopted network model are presented in Section 4.2; in Section 4.3, the design goals are presented; the details of the proposed clustering approach are analyzed in Section 4.4; this is followed by simulation and discussion in Section 4.5; and we conclude this chapter in Section 4.6.

## 4.2 Network model

### 4.2.1 Assumptions

We considered a network with  $N$  stationary randomly deployed sensors on a  $M \times M$  square. The set of sensors is represented by  $S = \{s_1, s_2, \dots, s_N\}$ . We also assume that the BS is stationary. Each sensor has a unique identifier  $i \in [1, N]$ . A sensor can communicate with another sensor if the sensor is within its communication range. We consider that all sensors are homogeneous. Also, sensors' batteries cannot be recharged and each sensor has power control capabilities to vary their transmit/receive power.

We suppose that, the location  $(x_i, y_i) \in \mathbb{R}$  of each sensor  $i$  is represented by  $\theta^i$ . We assumed that all sensors are locations unaware, but the location of the BS is known by each sensor. The BS is outside the sensors square, near the middle edge of the square, and has sufficient resources. Each sensor can receive a message from another, if it is in the communication range of the sender sensor. It is supposed that, the communication range  $R_c = 2 \cdot \pi \times d_s$  of the sensor  $s_i$ , is a circular region where the sensor is at the center. The sensing distance is denoted by  $d_s$ . In fact, a given sensor  $s_i$  located on  $\theta^i$  with a sensing range  $d_s$  can sense information on locations  $\theta^m$  if the Euclidean distance between these two points respects the condition given in Equation 4.1. The used notations are summarized in Table 4.1 and messages in Table 4.2.

$$\sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} \leq d_s \quad (4.1)$$

Table 4.1: The used Notations in the NEST Algorithm

Notation	Description
$CL_m$	Cluster members
$CH\_f(i)$	Followers of CH $i$
$CH\_c$	CH candidate
$CH\_c(i)$	Set of $CH\_c$ received by sensor $i$
$\varepsilon_l$	Minimum energy level to be $CH\_c$
$\varepsilon_t$	Quality assessment threshold
$\varepsilon_s$	Dance decay rate
$R_c$	Communication range
$T\_1$	Time of initialization step
$T\_2$	Time of CH election step

In addition, in order to design our approach, we proposed in Table 4.3, a few analogies between some features of the bee colony and sensor network elements. We consider that our WSN consists of a number of bee colonies that are seen as clusters. Also, for simplicity reasons, we suppose that all colonies are closed to each other and are located on a defined network square.

Table 4.2: Messages in the NEST Algorithm

Message	Description
<i>CHc_MSG</i>	<i>Cluster candidate message</i>
<i>CH_FOLLOW</i>	<i>Cluster join message</i>
<i>CH_TDMA</i>	<i>TDMA scheduling message</i>
<i>ELECT</i>	<i>Election message</i>

Table 4.3: Considered analogies between a Bee Colony and a Sensor Network

Bee colony	Sensor network
<i>Bee</i>	<i>Sensor node</i>
<i>Queen</i>	<i>CH</i>
<i>Nest-site</i>	<i>CH location</i>
<i>Colony</i>	<i>Cluster</i>

### 4.2.2 Energy model

We used a simplified energy model introduced by Heinzelman et al. [27] and adopted by most proposals in cluster-based sensor networks like those presented in [16, 23, 55, 56]. In this model, authors describe a free space and multi-path fading channels depending on the distance between the source and the destination. In fact, the transmission energy depends on the threshold distance  $d_0$ . If the distance from the source to the destination is less than  $d_0$ , the free space  $\epsilon_{fs}(pJ/(\text{bit}\cdot\text{m}^{-2}))$  parameter is used, otherwise, the multi-path  $\epsilon_{mp}(pJ/(\text{bit}\cdot\text{m}^{-2}))$  parameter is used. To transmit  $b$ -bits of data over a distance  $d$ , the energy of transmission  $E_T(b, d)$  is given by Equation 4.2:

$$E_T(b, d) = \begin{cases} b \times E_{elec} + b \cdot \epsilon_{fs} \cdot d^2 & \text{if } d < d_0 \\ b \times E_{elec} + b \cdot \epsilon_{mp} \cdot d^4 & \text{if } d \geq d_0 \end{cases} \quad (4.2)$$

Where:

- $E_{elec}$ (nJ/bit) is the sum of electrical energy required by digital coding, modulator and some other electronic circuits, to transmit or receive one bit of data.
- $\epsilon_{fs}$  is the required energy by the amplifier to send data directly to the recipient.
- $\epsilon_{mp}$  is the required energy by the amplifier to send data by using other sensors in a multi-hop way.
- $d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}$

The amount of energy expended by the radio to receive  $b$ -bits of data is given by Equation 4.3.

$$E_R(b, d) = b \times E_{elec} \quad (4.3)$$

In addition to the  $E_{elec}$ , we considered the energy of data aggregation  $E_{agg}$  and the energy of sensing  $E_{sen}$ .

### 4.3 Design Goals

Unlike most existing centralized bio-inspired clustering approaches, we designed a completely distributed approach for clustering in WSNs. In fact, most nature-inspired proposed clustering processes are centralized [16, 20, 23, 55, 56]. In these approaches, the clustering algorithms are performed in a *BS* and generally each sensor knows its location that it communicates to the *BS*. The *BS* retrieves all sensor locations, their energy levels and performs the clustering algorithm. Then, the *BS* sends a message to each sensor in order to inform it the cluster in which it belongs and the *id* of its corresponding *CH*. Designing a honeybee based technique for the choice of the *CH* and thus, for the formation of clusters in a distributed way, i.e., by the sensors themselves, would be a novel approach.

Certainly, the creation of completely distributed clustering protocols in WSNs is a very appealing and important line of research, given the limitations exhibited by centralized approaches. At first sight, the realization of such an approach seems not very interesting for WSNs. Indeed, the collective designation of a *CH* causes a lot of communication in the network, resulting in a lot of overhead. But, given the requirements in terms of applications, it is imperative that the sensor networks become smarter.

To achieve this goal, we proceed according to the BeeAdHoc approach introduced by Wedde et al. [86], which uses some foraging behaviors of honeybees to design an energy efficient routing algorithm for Mobile Ad Hoc Networks. We made a little abstraction by using some features of swarm intelligence in order to simulate a collective decision in WSNs. Especially in our case, it is possible to use just a few steps of the nest-sites selection process in particular: quality assessment and dance strength determination. According to the aforementioned goals, we proposed a clustering approach in which *CHs* are elected in a distributed manner. In the NEST algorithm, there is no periodical cluster-reformation as adopted by most approaches [13, 16, 23, 27, 56], but a new *CH* is elected by the cluster members when the quality of the actual *CH* becomes less than the given threshold.

### 4.4 Proposed Clustering Protocol

In this Section, our distributed bee inspired clustering algorithm is described. First, an overview of our proposed approach is presented and then the details of the constituent

steps are given.

#### 4.4.1 Overview of our approach

Like the categorization of the collective behaviors of social insects achieved by Garnier et al. [73], our approach includes four functions: coordination, cooperation, deliberation and collaboration. The coordination function is achieved in the network initialization, the cooperation and the deliberation functions are achieved in the selection of the *CHs* while the collaboration function consists of network scheduling. Similarly to the existing distributed clustering protocols for WSNs, the proposed scheme at its core follows the same steps as other distributed clustering protocols: sensors competing to be *CHs*, sending their bids, other sensors selecting the best *CHs* by following the best bids.

Even so, the innovation introduced by our approach is on how the best *CHs* are chosen by normal nodes. Indeed, in the honeybees nest-sites selection process, honeybees remain on the temporary site and follow the waggle dance performed by each scout bee that has discovered a potential nest-site, in order to select the best dancer. This selection is called dance strength determination. The *CH* selection approach is based on the dance strength determination, modeled by Equation 4.8 and Equation 4.9 and this is well explained in Section 4.4.3.

Initially, there is the node-competition after which the *CH* is elected, followed by the formation of clusters and these clusters cannot change during the network lifetime and finally, when a *CH* dies, there is the election of a new *CH* within the cluster. This is another innovation of our proposed approach. Indeed, when the *CH* finds that its energy level is below the threshold, it informs its *CH<sub>f</sub>* and initiates the *CH* election process within the cluster. Unlike LEACH [25], HEED [26] and other distributed clustering approaches, each proceeding *CH* election is launched if and only if it is necessary, i.e., when the *CH* energy level is lower than the threshold value  $\varepsilon_t$ . This is an added value in the proposal.

The design of the proposed clustering approach takes into consideration the fact that, sensors are uniformly distributed according to the discrete uniform probability distribution, see Equation 4.4. Each sensor can be a candidate to a *CH* status, if that sensor is eligible to certain criteria. At the initialization step during *T<sub>1</sub>*, each sensor computes its quality assessment according to Equation 4.6. If that quality is above the threshold  $\varepsilon_t$ , then the sensor becomes a *CH<sub>c</sub>* and broadcasts its status by sending a *CH<sub>c</sub>-MSG*.

After the initialization step, all *CH<sub>c</sub>* are known and the election process can start. During *T<sub>2</sub>*, when a sensor receives just one candidature for the *CH* status, that sensor follows the *CH<sub>c</sub>* by sending a *CH<sub>FOLLOW</sub>* message. In case of receiving more than one candidature, each *CH<sub>c</sub>* will execute a waggle dance in order to determine the best candidate. In



fact, the sensor that receives these candidatures will compute the dance strength of each  $CH_c$  according to Equation 4.9. These actions occur during the first part of  $T_2$ . In the second part of  $T_2$ , if a  $CH_c$  does not receive a  $CH\_FOLLOW$  message, i.e., each sensor that is not a  $CH$ , it changes its status to normal node and then it joins a  $CH_c$  for which, it received a  $CH_c\_MSG$  by sending a  $CH\_FOLLOW$  message. This will be done in the same way as actions described in the first part of  $T_2$ .

At the last step ( $T_3$ ), every sensor knows its  $CH$ , each  $CH$  knows its  $CH\_f$  list and all clusters are built. During this steady state phase, each  $CH$  establishes a  $TDMA$  scheduling in order to indicate to each  $CL_m$ , the time in which they will report their gathered data. In fact, as proposed in [12, 23, 55], the  $CHs$  divide the time into frames, and during each frame, each  $CL_m$  has one reserved slot: a broadcast slot, in which a  $CL_m$  sends its gathered data to its  $CH$ . This based scheduling method is adopted here in order to avoid collisions among data messages throughout the network. The inter-cluster and  $CH$  to  $BS$  communication is achieved by using  $CDMA$  while intra-cluster communication is done by  $TDMA$ .

#### 4.4.2 Initialization step: CH candidate

Each sensor has the same chance to become a  $CH_c$ . By taking into consideration the fact that all sensors are uniformly distributed, it is therefore necessary to use a constant probability distribution in order to give the same chance to each sensor to be a  $CH$ . We used the discrete uniform probability distribution. The probability density function for the discrete uniform distribution is given by  $P(i|N)$ , which is the probability of choosing one sensor among  $N$  sensors, see Equation 4.4. Since the number of sensors is fixed, the uniform discrete probability is a constant.

$$P(i|N) = \frac{1}{N} I_{1,\dots,N}(i) \quad (4.4)$$

The quality of the  $CH_c$   $i$ , i.e., the sensor located at  $\theta^i$  is calculated by the function  $J(\theta^i)$ , see Equation 4.5. In our case, because sensors are locations unaware, we brought back the computation of the quality  $J(\theta^i)$  to the residual energy of the sensor  $i$ .

$$J(\theta^i) = \frac{\text{Actual Energy of sensor } i}{\text{Initial Energy of sensor } i} \quad (4.5)$$

So, each sensor computes its quality assessment  $S^i(k)$ , see Equation 4.6, where  $k$  is an ordering number that counts the number of participations of a sensor in the  $CH$  election process.

$$S^i(k) = \begin{cases} 1 & \text{if } P(i) \times J(\theta^i(k)) + \Omega^i(k) > \varepsilon_t \\ 0 & \text{else} \end{cases} \quad (4.6)$$

Where  $\Omega^i(k)$  is a certain error/noise due to the transmission channel. As considered in [74], the sensor  $i$  randomly and uniformly takes this error into  $[-0.1, 0.1]$ , i.e., in the order of 10%.

Therefore, the set of potential nest-site  $NS_p$ , i.e., the list of  $CH\_c$  is given by Equation 4.7.

$$NS_p(k) = \{i \mid S^i(k) = 1\} \quad (4.7)$$

Each sensor  $i \in NS_p(k)$  broadcasts a  $CH\_c\_MSG$ .

#### 4.4.3 CH competition step: dance strength determination

At  $CH$  election time  $k$ , when a sensor receives more than one candidature, it has to proceed to a dance strength determination in order to choose the best nest-site, i.e., the best  $CH\_c$ . Each  $CH\_c$  has to perform a waggle dance and the best  $CH\_c$  will be the one that will perform a maximum number of waggle dances. For simplicity reasons, the dance strength is determined by the sensor that receives more than one candidate. To achieve this, the sensor  $i$  has to choose the best candidate  $b \in CH\_c(i)$  that will execute the maximum dance strength  $L^j(k)$ ,  $j \in CH\_c(i)$ . The corresponding cost-based fitness function that takes into account the energy objective and the received signal strength objective is given by Equation 4.8 and Equation 4.9.

$$j \in CH\_c(i) \mid L^j(k) = \max_{m \in CH\_c(i)} \{L^m(k)\} \quad (4.8)$$

$$L^j(k) = \frac{S^j(k)}{\gamma_{ij} - \varepsilon_s \cdot (k - 1)} \quad (4.9)$$

Where  $\gamma_{ij}$  is the strength of the received signal by  $s_i$  from  $s_j$ . This parameter represents the quality of the communication link between  $s_i$  and  $s_j$ . For simplicity reasons, in our implementation we consider  $\gamma_{ij}$  as the distance between  $s_i$  and  $s_j$ .

When the best  $CH\_c$  is determined, the sensor  $i$  sends a  $CH\_FOLLOW$  message. If there are  $CH\_c$ , which have not received a  $CH\_FOLLOW$  message, they have to follow a  $CH\_c$  in the same manner as described previously. After the reception of all following messages, each  $CH$  constructs its followers' list  $CH\_f(i)$ . These lists represent the clusters of the network. Each cluster is constituted by the  $Cl\_m$ .

#### 4.4.4 Scheduling step and CH election strategy

During the network scheduling time, each *CH* builds a *TDMA* and informs each *CL<sub>m</sub>*, the moment they must send their gathered data. Thus, to save energy, each sensor turns off its radio and activates it only for transmission when its slot time comes. Concerning the proceeding *CH* election, each *CH* which finds that its energy level is lower than the threshold  $\varepsilon_l$ , launches the above described election process for a new *CH* election within the same cluster.

Furthermore, in simulation, the threshold  $\varepsilon_l$  is fixed in a way that when the *CH* launches the election of the new *CH*, its energy is not depleted. When the new *CH* is elected, the old one will act like a follower as the others and it will perform the sensing task and the sending of its gathered data to its new *CH*.

#### 4.4.5 Proposed clustering algorithm

Our proposed distributed clustering approach inspired by the biological principles of nest-sites selection process of honeybee is resumed in the following algorithm. This algorithm is executed locally by each sensor and maintains the four functions categorized in [73] that permit a self-organization and self-adaptive features on the network.

#### 4.4.6 Complexity analysis

In this section, the complexity analysis of the proposed algorithm (Algorithm 4.1) is performed. Since all sensors are assumed to be the same in terms of enabled features, we evaluate the complexity of the execution of the NEST algorithm on one sensor. During the clustering process, i.e., in steps 1 and 2 of Algorithm 4.1, the algorithm complexity is  $O(m)$  if the current sensor is a *CH* and  $O(1)$  if the current sensor is a follower of a given *CH*. Where  $m$  is the number of followers of the current sensor. While in the data routing process, i.e., in the scheduling step, in which each sensor has the main function of sending or receiving the gathered data, the convergence time is in the order of  $O(m \cdot b)$  for *CH* sensors and  $O(1)$  for followers. Where  $b$  represents the size of the data packet.

### 4.5 Simulation and Discussion

To evaluate the performance of the proposed algorithm, we analyzed three principle metrics. First, the network re-clustering is observed, i.e., the number of *CH* elections during the network scheduling time. Then, we observed the network lifetime that is an important characteristic usually analyzed in the performance evaluation of sensor networks by measuring the power consumption, the energy efficiency, the time at which the first sen-

---

**Algorithm 4.1** Algorithm Simulating the NEST
 

---

**BEGIN**

**Step 1:** CH candidate.  
 /\* Step duration:  $T_1$  \*/

1.  $k = 1$   
 /\*  $k$  is the number of participations in the election process \*/
2. STATE = 1
3. /\* if STATE is set to 1, the sensor is in the election process \*/
4. **1.1.** Compute the quality assessment according to Equation 4.6.
5. **1.2.** if  $S^i(k) == 1$  **then**
6.     Broadcast a *CHc-MSG*

**Step 2:** Reception and confirmation of candidature  
 /\* Step duration:  $T_2 = t_2 + t_2^*$  \*/

**2.1.** First part  
 /\* duration:  $t_2$  \*/

7.     if  $S^i(k) == 1$  **then**
8.         Wait for a *CH-FOLLOW* message
9.     **else**
10. **2.2.** if  $length(CH_c(i)) \geq 2$  **then**
11.     Proceed to a dance strength determination according to Equation 4.8 & 4.9
12.     Send a *CH-FOLLOW* message
13.     **else**
14.         Send a *CH-FOLLOW* message to the best dancer

**2.3.** Second part  
 /\* duration:  $t_2^*$  \*/

15.     if  $length(CH_f(i)) == 1$  **then**  
        Perform 2.2.
16.     STATE = 0

**Step 3:** Scheduling  
 /\* Until the end of lifetime \*/

17. **3.1.** if  $length(CH_f(i)) \neq 0$  **then**  
 /\* The current sensor is CH \*/
18.     Construct TDMA
19.     Send a *CH-TDMA* to each cluster member,  $f \in CH_f(i)$
20.     **while** (CH energy level  $> \epsilon_l$ )
21.         Receive sensed data from each  $f \in CH_f(i)$
22.         Aggregate these data and send it to the BS by using CDMA
23.     **if** CH energy level  $< \epsilon_l$  **then**  
 /\* Launch the CH election amongst CH\_m \*/
24.         Send *ELECT* message to each  $f \in CH_f(i)$
25.          $k = k + 1$
26.         STATE = 1  
        /\* Start the election process \*/
27.         Execute Steps 1 to 3
28. **3.2.** **else**  
 /\* The current sensor is a CH\_f \*/
29.     Receive *CH-TDMA*
30.     **while** (not receive *ELECT* message)
31.         Send sensed data to CH
32.     **if** receive *ELECT* message **then**
33.          $k = k + 1$
34.         STATE = 1
35.         Execute Steps 1 to 3

**END**

---

sensor will deplete all its power and the end-to-end delay. Finally, in an efficient WSN, the throughput is expected to be maximal. So, the amount of packets received by the *BS* is observed by measuring the rate of successful packets delivered to the *BS*.

The proposed distributed clustering approach was implemented by using a discrete model developed in Matlab. We evaluated the NEST algorithm by conducting a series of intensive experiments. Unlike the semi-distributed biological inspired clustering approach proposed in [13, 16, 56], as well as the completely centralized clustering algorithms introduced in [23, 55], all sensors in our proposed approach have almost the same features. All sensors are homogeneous and can perform any task over the network. There is no special sensor with a particular behavior. All sensors are locations unaware. The only particular feature is that, 20% of sensors randomly considered, have more energy than the rest.

Simulation is performed by taking into consideration a WSN with 100 stationary sensors randomly and uniformly deployed on a  $500 \times 500$   $m^2$  sensing area during 5000 iterations. The *BS* is located at (250, 575) and 20% of randomly considered sensors have 100 *Joules* of initial energy, while the remaining sensors have 40 *Joules* of initial energy. Different energy levels are adopted in order to be closer to the reality. The transmission range of each sensor is fixed to 87 meters. We used energy parameters introduced in [27]. Energy loss due to the communication channel is assumed to be proportional to the distance square  $d^2$ . The simulation parameters are summarized in Table 4.4.

Table 4.4: Adopted Simulation Parameters for the NEST

Parameter	Value
Square ( $m^2$ )	$500 \times 500$
<i>BS</i> location	(250, 575)
Number of sensor nodes	100
Initial Energy (J)	40 (80% of sensors) 100 (20% of sensors)
Number of iterations	5000
$E_{elec}(nJ/bit)$	50
$\epsilon_{fs}(pJ/(bit.m^{-2}))$	10
$\epsilon_{mp}(pJ/(bit.m^{-4}))$	0.0013
$d_0(m)$	87
$E_{agg}(nJ/bit)$	5
$E_{sen}$	0
Packet size (Kbit)	512
Message size (kbit)	2
Bandwidth of channel (Kbit/s)	250

Like the modeling and the analysis of the nest-sites selection process achieved in [74], the noise  $\Omega$  due to the communication channel in the assessment of *CH<sub>c</sub>* quality is taken in  $[-0.1, 0.1]$ . The thresholds of the quality assessment and the dance decay rate presented in Table 4.5 have been taken at their best values after performing various experimentations. In these experiments, the network is up until the number of dead sensors reaches 80% of all sensors.

Table 4.5: Adopted Swarm Parameters for the NEST

Parameter	Value
$\Omega$	$[-0.1, 0.1]$
$\varepsilon_l$	20
$\varepsilon_t$	0.15
$\varepsilon_s$	0.6

The proposed semi-distributed PSO-based clustering approach in [16, 56] and the centralized PSO algorithm used in performance comparisons achieved by Latiff et al. [55] concluded that, PSO-based protocols perform successful results. In the same mind, Karaboga et al. [23] as well as Ari et al. [13] have proposed centralized clustering approaches implemented at the *BS*, based on efficient and fast searching features of the ABC optimization algorithm. They showed that their proposed protocols optimize the construction of clusters and the selection of *CHs*. Unlike these centralized and semi-distributed approaches, the NEST algorithm is completely distributed and self-organized.

In simulation, we considered the same fitness for PSO and ABC, as adopted in [23], in order to make a reliable comparison. Also, the number of *CHs* is set to 5 for PSO and ABC, and the probability value for a given sensor to be chosen as a *CH* is set to 0.05 for the LEACH. In the NEST protocol, the number of *CHs* cannot be fixed because the election of *CHs* is achieved in a distributed self-organized manner, i.e., by sensors themselves.

Figure 4.1 shows a box plot of the number of *CHs* during 10 runs of the proposed protocol. It can be seen that the median of the *CH* number is about 5. Also, in order to demonstrate the advantage of the proposed *CH* election strategy, Figure 4.2 shows the number of *CHs* and the number of *CH* elections during 10 runs of the NEST. This test shows the improvement achieved in the number of *CHs*' elections since the results show that there is a lesser number of *CH* changes within clusters. In contrast of the aforementioned LEACH, PSO and ABC clustering approaches that achieve *CH* election at each iteration time, it is clear that, the number of *CH* elections, i.e., the number of re-clustering, is really optimized in NEST. This low number of *CH* elections is an important issue that prolongs the network lifetime.

For the performance evaluation of our proposed approach, we made comparisons with some of well known WSN cluster-based protocols:

- The LEACH [25] that is the most popular, widely used and completely distributed clustering algorithm for WSNs.
- The PSO [87] based clustering approach that is a completely centralized bio-inspired approach.
- The ABC [23] cluster-based WSN routing approach that implements a centralized

Figure 4.1: Box plot of CH number in the NEST during 10 runs.

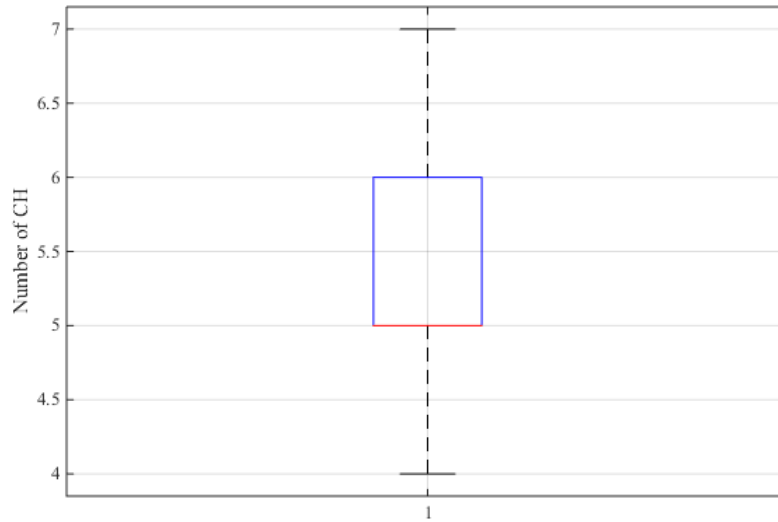
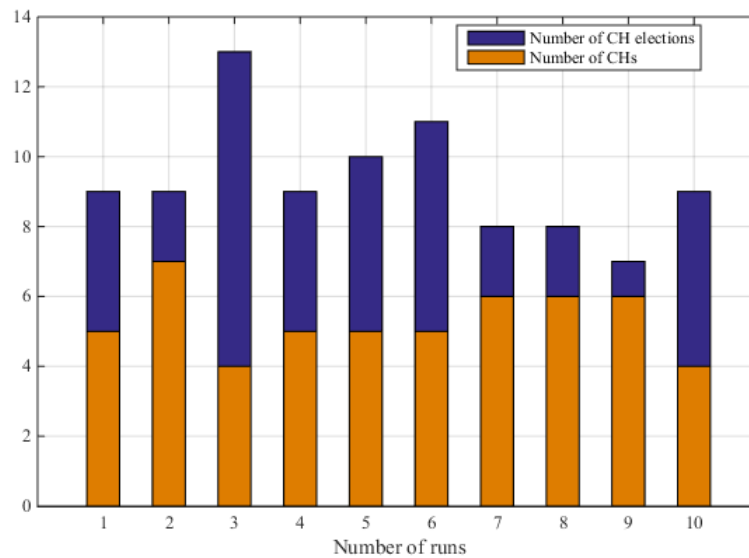


Figure 4.2: Number of CHs and number of CH elections in the NEST during 10 runs.



clustering strategy inspired by the foraging behaviors of honeybees.

In Table 4.6, equations of fitness functions of these protocols are briefly presented.

To measure the efficiency of our distributed clustering algorithm, we evaluated the network

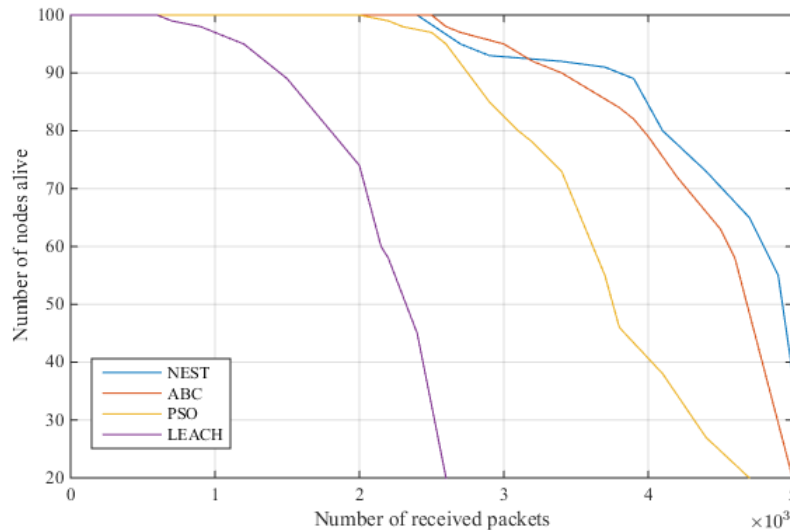
Table 4.6: Fitness functions of compared protocols with the NEST

Fitness	Protocol
$L^j(k) = \frac{S^j(k)}{\gamma_{ij} - \varepsilon_s \cdot (k-1)}$	NEST
$T(i) = \begin{cases} \frac{P}{1 - P \cdot [r \cdot \text{mod}(\frac{1}{P})]} & \text{if } i \in G, \\ 0 & \text{otherwise} \end{cases}$	LEACH
$f^{ICWAQ} = \delta \cdot f^{ICWA} + (1 - \delta) \cdot f^{qos}$	PSO ABC

lifetime, the remaining network energy and the delivery time, versus the amount of packets received by the *BS*. Performance comparisons of these protocols with the proposed protocol are depicted in Figures 4.3, 4.4 and 4.5.

Figure 4.3 shows the number of sensors alive versus the amount of packets received by the *BS*. It can clearly be observed that, our approach improves the network lifetime. In the NEST, when the number of dead sensors reaches 60% of all sensors, the *BS* receives two times the number of packets received by the LEACH, 20% more than the packets received by PSO and 3% more than the packets received by the ABC. So, it is clear that the NEST protocol allows the minimum death of sensors.

Figure 4.3: Network lifetime



Moreover, we evaluated the network lifetime by observing the time at which the first sensor dies. To achieve that, we simulated the network lifetime until the first sensor becomes out of battery. We have taken the mean and the standard deviation of results of all the runs



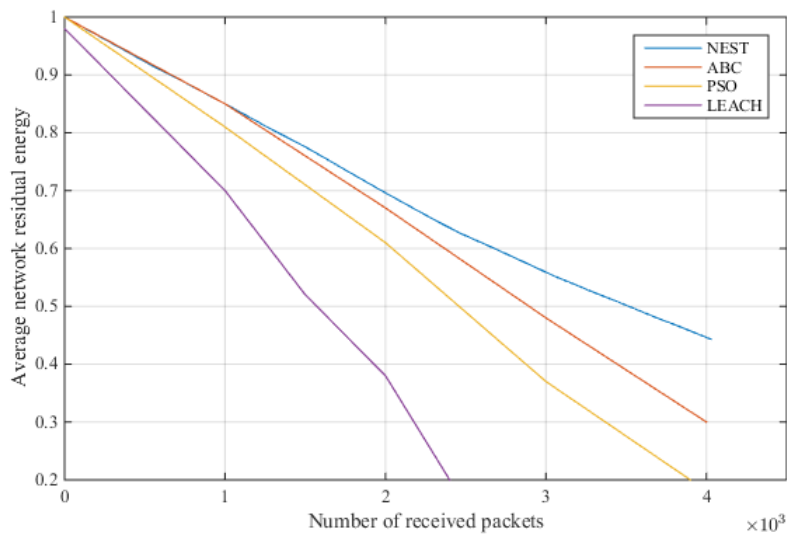
of the considered algorithms. In Table 4.7, it can be seen that, the first dead sensor in the NEST occurs long after the first dead in the PSO and LEACH algorithms. However, even if the NEST algorithm has the smallest standard deviation, in the ABC based clustering algorithm, the first dead sensor occurs at almost the same time as in the NEST.

Table 4.7: Mean of time at which the first sensor dies ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ).

	NEST	ABC	PSO	LEACH
$\bar{\mu}$	2003	2017	1009	403
$\sigma$	6	13	21	17

The amount of energy consumption is highlighted in Figure 4.4 by plotting the average network residual energy versus the number of received packets. It can be observed that, our approach performs a great number of operations while minimizing the energy consumption. At  $4 \times 10^3$  packets received by the *BS*, about 45% of the network remaining energy is still available while the network remaining energy for ABC is about 30%. PSO and LEACH are less performant than others. The results obtained here in addition to the early mentioned results for the network lifetime and the time at which the first sensor becomes out of battery, demonstrate the effectiveness in the distribution of the energy consumption among the network in the NEST algorithm. The reason for this significant energy conservation in the NEST is due to the intelligence introduced by the adopted bee swarm behaviors in the *CH* selection process.

Figure 4.4: Amount of energy consumed



Furthermore, in order to show that the NEST algorithm allows an efficient energy-balanced

topology, we evaluated the energy efficiency by computing the average of the residual energy of each node in the network at the end of the simulation time. The results presented in Table 4.9 show the comparison of the NEST algorithm and other protocols in terms of the average residual energy of each node and the associated standard deviation. It was noted that, in the case of large scale sensor network, the average residual energy of each node at the end of simulation time in the NEST algorithm is higher than ABC, PSO and LEACH. This is mainly due to the adopted quality assessment and dance strength models in the NEST.

Table 4.8: Average residual energy of each node ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ) during 10 runs.

	NEST	ABC	PSO	LEACH
$\bar{\mu}$	<b>0.447</b>	0.302	0.174	0.000
$\sigma$	<b>0.017</b>	0.074	0.031	0.000

In Figure 4.5, it is seen that, the ABC performs greater amounts of packets received by the *BS* according to the iteration time. Certainly, in the ABC protocol, the *BS* receives a bit more packets, but, in our proposed clustering approach, until 1500 iterations are made, the *BS* in the NEST receives slightly more packets than in the ABC and PSO. Also, at 4000 iterations, the number of received packets by the *BS* in the NEST tends to reach that of ABC. Unfortunately, in the LEACH the amount of received packets by the *BS* is worst. This is explained by the poor management of the cluster election in the LEACH algorithm. In fact, each living sensor participates in the election at each iteration. Whereas in our approach, a given sensor takes part in the CH election process only if its *CH* launches the process.

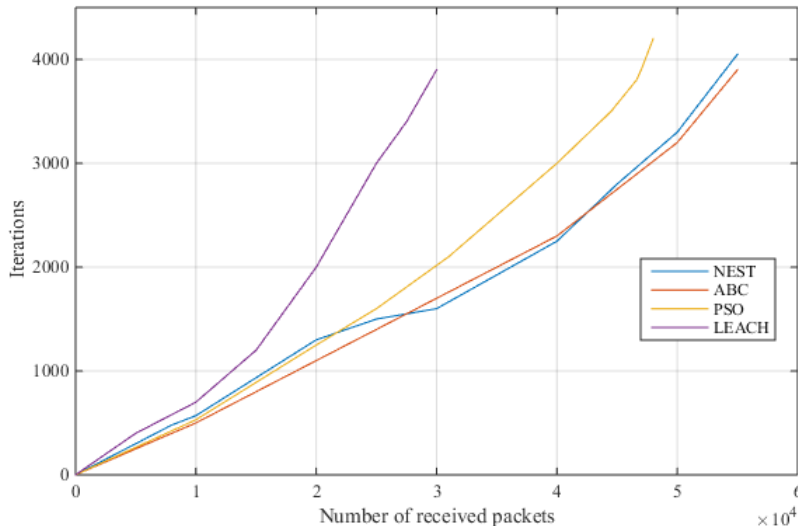
Besides, over and above of the measurement of the throughput, we computed the mean as well as the standard deviation of the Packet Delivery Rate (PDR) for data packets received by all the *CHs* from their respective followers during 10 runs of the algorithm. The results are presented in Table 4.9. It can be observed that the proposed protocol has significantly better PDR than the compared protocol. Indeed, the mean of PDR allowed by the NEST algorithm is maximum with almost minimum deviations in the mean of PDR.

Table 4.9: Mean of PDR ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ) during 10 runs.

	NEST	ABC	PSO	LEACH
$\bar{\mu}$	<b>0.898</b>	0.801	0.628	0.469
$\sigma$	<b>0.007</b>	0.031	0.027	0.020

Figure 4.6 plots the average of end-to-end delay encountered by packets in the NEST during the simulation time compared to ABC, PSO and LEACH. The curves present

Figure 4.5: Amount of packets received by the BS

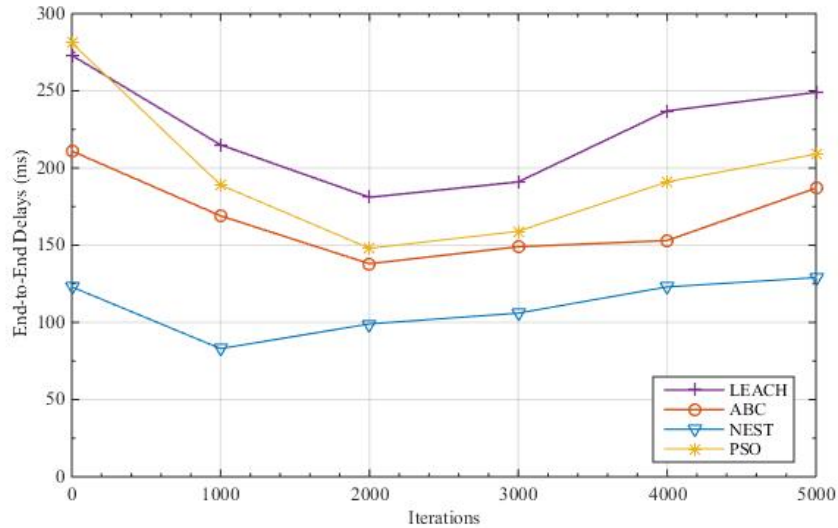


the average end-to-end delay of ten runs of each protocol. We observed that the NEST algorithm has lower average end-to-end delay than the compared protocol. This visible performance is favored by two main factors. First, the NEST forwards data on relatively shortest path from each follower to its corresponding *CH*, hence, reduces queuing and propagation delays. Then, the observed outcome is allowed by the adopted values of the noise parameter  $\Omega \in [-0.1, 0.1]$  due to the communication channel in the assessment of the quality of candidates to a *CH* position. However, LEACH algorithm enabled major delays. Indeed, LEACH allows nodes to transmit their gathered data at their slot time in the entire network TDMA schedule in a single hop communication. It introduces major delays due to the period during which failed paths can be recovered.

In the clustering approach adopted in the LEACH and other more recent clustering approaches such as PSO protocol for hierarchical clustering developed by Elhabyan and Yagoub [15], the *CH* election for the entire network is achieved at every iteration. This introduces greater overhead and so, the amount of energy wastage is high. In the ABC and PSO protocols, the clustering operation is performed at the *BS* at the beginning of each iteration. Then, information about clusters and *CLm* is transmitted to each sensor during each iteration. To avoid this additional overhead in our approach, after network initialization, the *CHs* are elected by sensors themselves during iteration if it is necessary. The *CH* election is performed within clusters, which have a *CH* with insufficient energy with respect to the threshold value  $\varepsilon_l$ .

The advantage of using the bio-inspired model proposed here over the most extended

Figure 4.6: Average of end-to-end delay during 10 runs.



ABC algorithm introduced by Karaboga [64] and successfully applied in WSN clustering in [13, 23], is based on the fact that, with ABC it is not possible to choose the *CH* in a distributed manner. It is only done in a centralized manner. Indeed the ABC allows a periodical choice of the *CH* at the *BS*. However good, designed for large-scale networks, the NEST allows the election of the *CHs* by sensors themselves, in a distributed self-organized way. This is an important goal of clustering in sensor networks.

## 4.6 Conclusion

In this chapter, we addressed the issue of clustering in WSNs by using social behaviors of honeybees. In particular, we have given an importance to the *CH* election process in order to reduce energy wastage introduced by the most existing clustering and routing algorithms. In the NEST, *CHs* are elected in a distributed and a self-organized manner by sensors themselves and each proceeding *CH* election is done inside the cluster, if the *CH* election is necessary. Thus, the energy consumption of the network is significantly balanced within all cluster members. This avoids the rapid death of *CHs*. The results of experiments showed that, our approach is more performant than the most existing clustering solutions in terms of the network lifetime, energy consumption and amount of packets received by the *BS*.



## Chapter 5

# A Power Efficient Cluster-based Routing Algorithm for WSNs: Honeybees Swarm Intelligence based approach

### 5.1 Introduction

This chapter presents a novel cluster-based routing protocol called ABC-SD. The proposed protocol exploits the biologically inspired fast and efficient searching features of the ABC metaheuristic to build low-power clusters [65]. For the choice of cluster heads, a multi-objective fitness function is designed by using a LP formulation. The routing problem is addressed by a cost-based function that makes a trade-off between the energy efficiency and the number of hops of the path. The clustering process is achieved at the BS with a centralized control algorithm, which exploits energy levels and the neighborhood information of location-unaware sensors. As for the routing of gathered data, it is realized in a distributed manner. Furthermore, a realistic energy model is adopted in the considered network model. The proposed protocol is intensively experimented with a number of topologies in various network scenarios and the results are compared with the well-known cluster-based routing protocols that include the swarm intelligence based protocols. The obtained results demonstrate the effectiveness of the proposed protocol in terms of network lifetime, network coverage and the amount of packets delivered to the BS. The main contributions can be summarized as follows: a LP formulation of the clustering problem; the routing problem addressed by a CF; proposition of an ABC-based clustering algorithm with a tradeoff between the energy consumption and the quality of the communication link; the introduction of a pre-established routing mechanism in which

routing paths are less costly in terms of power consumption; the integration of a realistic energy model and realistic network settings; the simulation of the proposed protocol to demonstrate its performance compared to some existing protocols.

The rest of the chapter is organized as follows: the network model is presented in Section 5.2; in Section 5.3, the design goals as well as the proposed clustering and routing algorithms are described; this is followed by the performance evaluation and discussion in section 5.4; and the conclusion and directions for future work are presented in section 5.5.

## 5.2 Network Model and Vocabulary

In this Section, we present the used realistic energy model based on the signal strength indicator. Moreover, we describe the used vocabulary and the adopted assumptions.

### 5.2.1 Energy Model

Like done by Elhabyan and Yacoub [29], our energy model uses sensors that include the realistic features of Chipcon CC2420 radio transceiver of which the data sheet is given in [88]. The Chipcon CC2420 is a low-cost highly integrated 2.4 GHz IEEE 802.15.4 compliant and ZigBee™ radio transceiver designed for low power WSNs' applications. The CC2420 features include a hardware support for packet handling, data buffering, multiple transmissions, clear channel assessment, link quality indication and packets synchronization.

In consonance with the IEEE 802.11 implementation of the Received Signal Strength Indicator (RSSI) that is a measurement of the power present in a received radio signal, in a wireless environment, the received signal strength is assumed to be a realistic approach for quantifying power consumption in WSN environments [5]. To exploit that in our energy quantification, it is assumed that a sensor battery has linear charge and discharge characteristics. So, the energy  $E_i$  consumed by sensor  $s_i$ , is equivalent to the accumulation resulting from that of its components [7]. The energy consumption of components contains energy of event execution and the energy spent in the transition between states. The energy  $E_S$  spent by a sensor within states is given in Equation 5.1:

$$E_S = \sum_j (p_j \times t_j) \quad (5.1)$$

Where:

- The index  $j$  refers to one of the four states of the Chipcon CC2420: idle, sleep, reception or transmission. The values of power consumption in each of these states is given in Table 5.1.

- $p_j$  is the mean of the power consumed in each state  $j$ .
- $t_j$  is the operation time in the corresponding state.

Then, the energy  $E_T$  spent in transition between states is given in Equation 5.2:

$$E_T = \sum (e_t \times c_t) \quad (5.2)$$

Where:

- $c_t$  is the occurrence frequency.
- $e_t$  is the mean of the consumed energy in each state transition.

Hence, the total energy consumed by the sensor  $s_i$ ,  $E_i$  is given by Equation 5.3. The values of  $p_j$  and  $e_t$  are given in the data sheet of the Chipcon CC2420, which can be found in [88].

$$\begin{aligned} E_i &= \sum (E_S + E_T) \\ &= \sum \left( \sum_j (p_j \times t_j) + \sum (e_t \times c_t) \right) \end{aligned} \quad (5.3)$$

Table 5.1: Energy consumption of the different states in the Chipcon CC2420.

State	Consumption ( $\mu\text{A}$ )
<i>idle</i>	426
<i>sleep</i>	20
<i>receive (rx)</i>	$19,7 \times 10^3$
<i>transmit (tx)</i>	$17,4 \times 10^3$

### 5.2.2 Vocabulary

The considered expressions are given hereinafter:

- The sensing environment is a 2-dimensional square field denoted by  $\mathcal{D} \subseteq \mathbb{R} \times \mathbb{R}$ .
- $S = \{s_1, s_2, \dots, s_N\}$ , the set of  $N$  sensor nodes.
- The identifier of the *BS* is 0 and each sensor node has a unique identifier  $id \in [1, N] \cap \mathbb{N}$ .
- A Candidate to a *CH* position is represented by  $CH_c$
- The set of *CHs* is represented by  $\beth(ch) = \{c_1, c_2, \dots, c_m\}$ , where  $m$  is the number of *CHs*, i.e., the size of  $\beth(ch)$ .



- The field of cluster represented by the *CH*  $c_j$ ,  $j \in [1, m] \cap \mathbb{N}$ , is denoted by  $\mathcal{A}_j$ .
- The followers' list of the *CH*  $c_j$ ,  $j \in [1, m] \cap \mathbb{N}$  is represented by  $f(c_j)$ .
- The cluster member is represented by  $Cl_m$ .
- The number of  $Cl_m$  in the cluster represented by  $c_j$ ,  $j \in [1, m] \cap \mathbb{N}$  is denoted by  $N(\mathcal{A}_j)$ ,  $j \in [1, m] \cap \mathbb{N}$ .
- The  $x^{th}$   $x \in [1, N(\mathcal{A}_j)] \cap \mathbb{N}$  cluster member of the cluster represented by the head  $c_j$ ,  $j \in [1, m] \cap \mathbb{N}$  is represented by  $Cl_{m_j}(x)$ .
- The *RSSI* value received by sensor  $s_i$  from sensor  $s_j$ ,  $i, j \in [1, N] \cap \mathbb{N} : i \neq j$ , is represented by  $\mathfrak{R}(i, j)$ .
- The remaining energy of sensor  $s_i$ ,  $i \in [1, N] \cap \mathbb{N}$  is represented by  $E_r(s_i)$ .
- The neighbor table of sensor  $s_i$ ,  $i \in [1, N] \cap \mathbb{N}$  is represented by  $\mathcal{N}(s_i)$ .

### 5.2.3 Assumptions

Let us consider a network with  $N$  stationary randomly deployed sensors on a two-dimensional square field. We assume that the *BS* is stationary and located at the coordinate  $(0, 0)$ . Each sensor in the network has a unique identifier  $i \in [1, N] \cap \mathbb{N}$ . Let's consider that all sensors are homogeneous and the computation and power capabilities of all of them are the same. Each sensor can communicate with another sensor if that sensor is within its communication range. It is assumed that communication links are bidirectional. Sensors' batteries cannot be recharged and each sensor is equipped with a power control device that has capabilities to vary their transmit/receive power. We assume that sensors are locations unaware. However, the location of the *BS* is known by each sensor. We assume that all sensors maintain a synchronization protocol, which allows a consistent time synchronization among the sensors in the network.

## 5.3 Algorithms Design

In this Section, we first present the design goals of the proposed ABC-SD approach and then, we describe specifications and the used mechanisms in the proposed clustering and the routing algorithms.

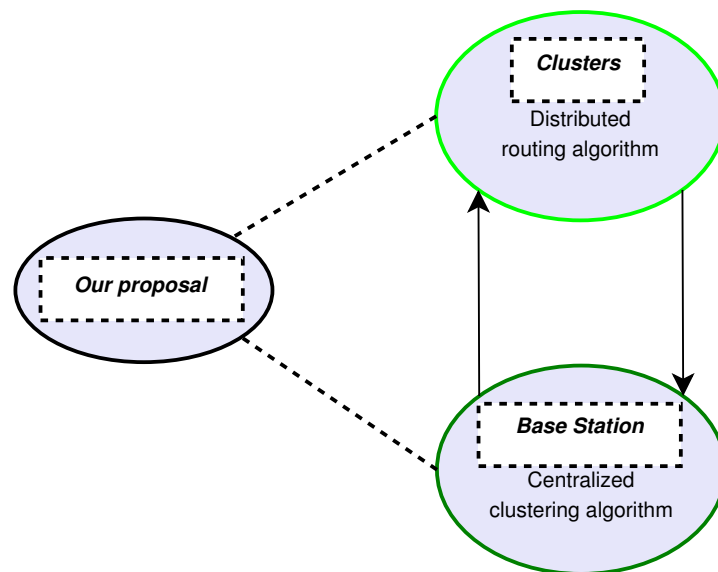
### 5.3.1 Design goals

The requirements of sensor-based applications require that sensor networks become smarter. Since sensors are designed with limited resources, networking these sensors remains a very

appealing and important issue of WSNs research. Thus, in this paper, we are positioned in WSNs' applications that require the use of energy efficient routing strategies that aim at prolonging the network lifetime while taking into account the delivery of a good network QoS, especially in terms of throughput and data delivery. Hence, given the requirements in terms of the applications' needs, designing an energy efficient protocol for routing the gathered data remains actual, given the resource limitations exhibited by sensor's design. To achieve this goal, we designed a bio-inspired cluster-based routing in which the clustering algorithm uses some foraging behaviors of honeybees.

Since the main goals are the achievement of a good scalability and the extension of the network lifetime, our proposed protocol design aims at providing a good load balancing within clusters by using a multi-hop communication among sensors and their corresponding *CHs*. Indeed, the proposed approach schematized in *Figure 5.1* is designed in such a way that the clustering algorithm is performed in a centralized manner at the *BS* while the routing is performed in a distributed manner inside clusters and among *CHs*. So, the protocol acts in a semi-distributed manner.

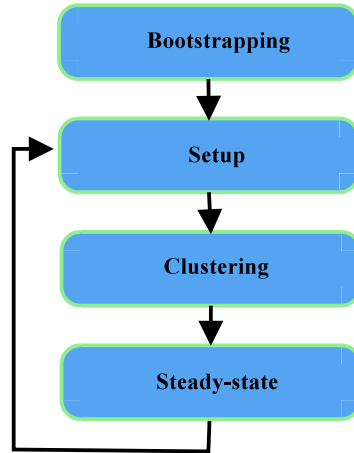
Figure 5.1: ABC-SD: Design scheme.



In fact, when receiving sensor's information at the start of each round, the *BS* will build clusters by paying attention to the choice of the *CH*, which is well achieved by the position update of the ABC algorithm and by doing so, the energy expenditure is reduced as we can see in the results of simulation (*see section 5.4.3*). When clusters are built, the *BS* sends a message to each sensor in order to inform it about the cluster in which it belongs and the *id* of its corresponding *CH*. Then, the distributed part of the proposed algorithm, i.e., the routing mechanism, starts within each cluster by paths discovery followed by data gathering and transferring operations. These operations of transferring data to *CHs*

follow a convergecast tree, i.e., *many-to-one* within the cluster. Transferring aggregated data to the *BS* follows also the same process in which *many* represent CH sensors and *one* represents the *BS*. Since a lot of traffic is generated, particularly during the network initialization and the route discovery, our design implements a mechanism that filters out redundant data. The whole steps performed in the execution of the ABC-SD protocol is highlighted in *Figure 5.2*.

Figure 5.2: The main steps of the proposed ABC-SD protocol.



### 5.3.2 ABC-SD clustering mechanism

The clustering process is preceded by the setup of the network that consists of bootstrapping and neighbor discovery.

#### 5.3.2.1 Bootstrapping and setup

When the sensors deployment is carried out, each sensor  $s_i$ ,  $i \in [1, N] \cap \mathbb{N}$  and also the *BS* broadcasts a *H\_MSG* packet that is received by each node positioned in the communication range of the *hello* sender. Each node maintains two informations about its neighbors in its neighbor table: the *id* of the neighbor and the value  $\mathfrak{R}(i, j)$  of the *RSSI* received. Whenever a given sensor receives a *H\_MSG* packet, it adds an entry in its neighbor's table. At the end of the neighbor discovery time, each sensor  $s_i$ ,  $i \in [1, N] \cap \mathbb{N}$  sends its triplet  $(id, E_r(s_i), \mathcal{N}(s_i))$  to the *BS* by using a controlled flooding mechanism that forces a sensor not to forward a packet if it already forwarded it. The flooding operation is executed during a certain time limit. When receiving the triplets, the *BS* starts the clustering process.

### 5.3.2.2 Clustering process

Before giving the description of our proposed clustering algorithm, we present some mathematical concepts used for the clustering process. Let  $\lambda_d$  (see Equation 5.4),  $d \in \mathbb{N} \setminus \{0\}$  represents the  $d$ -dimensional Lebesgue measure defined on subsets of  $\mathbb{R}^d$ :

$$\lambda_d(\mathcal{A}) = \int_{\mathcal{A}} 1 \, dx \quad (5.4)$$

Since the considered sensing environment  $\mathcal{D}$  is a 2-dimensional square field, the area of  $\mathcal{A}_j \subseteq \mathbb{R} \times \mathbb{R}$  is given by  $\lambda_2(\mathcal{A}_j)$ ,  $j \in [1, N] \cap \mathbb{N}$ . Then, let us consider that, sensors are deployed by a random process that positions the sensors at random points in  $\mathcal{D}$ . Now, for each  $\mathcal{A}_j \subseteq \mathcal{D}$ :  $j \in [1, N] \cap \mathbb{N}$ , let us consider the number  $N(\mathcal{A}_j)$  of sensors in  $\mathcal{A}_j$ . Then,  $N(\mathcal{A}_j)$  obeys the following axioms:

**Axiom 5.1.**  $\{N(\mathcal{A}_j) : \mathcal{A}_j \subseteq \mathcal{D}, j \in [1, N] \cap \mathbb{N}\}$ , which is the set of random variables that represents clusters, is a Poisson process on the sensing environment  $\mathcal{D}$  with the density parameter  $r > 0$ .

**Axiom 5.2.**  $N(\mathcal{A}_j)$  has the Poisson distribution with parameter  $r\lambda_2(\mathcal{A}_j)$ . The probability mass function of  $N(\mathcal{A}_j)$  is given by Equation 5.5 (see [89]):

$$p(k) = \frac{(r\lambda_2(\mathcal{A}_j))^k}{k!} \times e^{-r\lambda_2(\mathcal{A}_j)} \quad (5.5)$$

**Axiom 5.3.** Since areas  $(\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_j)$  are such that  $\bigcup \mathcal{A}_j \subseteq \mathcal{D}$  and  $\bigcap \mathcal{A}_j = \emptyset$ , the collection of variables  $(N(\mathcal{A}_1), N(\mathcal{A}_2), \dots, N(\mathcal{A}_j))$  is a sequence of independent random variables.

Then, after the bootstrapping and setup phase, the *BS* has to compute the set  $\beth(ch)$  of *CHs* and construct the clusters. The process starts by the construction of the set  $S_{CH_c}$  of candidates to a *CH* position according to Equation 5.6. Having sufficient energy is the main condition to be eligible as a  $CH_c$ . In other words, only the sensor that has energy level above the threshold value  $\varepsilon_t$ , can be taken as candidate.

$$S_{CH_c} = \{s_i, i \in [1, N] \cap \mathbb{N} : E_r(s_i) \geq \varepsilon_t\} \quad (5.6)$$

Next, the ABC algorithm presented in Chapter 2 is used in the assignment of followers of each *CH* in  $\beth(ch)$ . So, the initial population of the ABC algorithm is generated according to the number of *CHs*. Each individual containing a sequence of  $m$  randomly selected  $c_j \in \beth(ch)$ ,  $j \in [1, m] \cap \mathbb{N}$  in such a way that each *CH* can directly communicate with at least one *CH* and at least one *CH* can directly communicate with the *BS* by exploiting their neighbor tables. The number of elements of  $\beth(ch)$  is not greater than 5% of the

whole sensors. For the followers assignment, a population of bees is used to find the best  $CH$  where each bee flies in the  $m$ -dimensions search space. Each  $CH$  is assimilated to an employed bee.

The energy wastage that can be introduced by data transferring is intended to be reduced in a way that, the distance as well as the number of hops among sensors and their corresponding  $CH$ s is kept as small as possible. Since our proposed protocol aims at being a multi-hops cluster-based routing protocol, the number of hops within a cluster is not reduced but the path that takes the data towards the  $CH$  is chosen in a way that the number of crossed nodes is smallest as possible. This is explained in Section 5.3.3. So, to achieve the distance shortening, each sensor  $s_i \in S, i \in [1, N] \cap \mathbb{N}$ , becomes a follower of a given  $c_j \in \mathcal{C}(ch), j \in [1, m] \cap \mathbb{N}$ , i.e., an element of the list  $f(c_j)$ , by considering the best  $RSSI$  value  $\mathfrak{R}_b$  (see Equation 5.7) among the  $CH$  received  $\mathfrak{R}(i, j)$  values. See illustration 5.1 for a good understanding.

$$\mathfrak{R}_b = \max_{j \in [1, m] \cap \mathbb{N}} \left( \mathfrak{R}(i, j) \right) \quad (5.7)$$

**Illustration 5.1.** *If the  $CH$ s  $c_1$  and  $c_2$  receive  $RSSI$  values from sensor  $s_3$ ,  $s_3$  will be assigned as a follower of the  $CH$  that received the best  $RSSI$  from  $s_3$ . Thereby, for each sensor  $s_i, i \in [1, N] \cap \mathbb{N}$ , sensors are evaluated and assigned to their corresponding heads.*

So, when the followers are built, the best  $CH$ s are selected according to the fitness function depicted in Equation 5.15, and the position update process of the ABC algorithm (see Algorithm 2.1). The proposed clustering algorithm is given thereafter (see Algorithm 5.1). The main relationship between the proposed ABC-SD protocol and the original ABC protocol lies in the position update process of the ABC algorithm, which is used in the optimization of sensors assignment to clusters in ABC-SD.

---

#### Algorithm 5.1 ABC-SD: Clustering Algorithm

---

Begin  
1. Construct the set  $\mathcal{C}(ch)$   
2. Generate the initial population  $X_j, j \in [1, m] \cap \mathbb{N}$   
3. **Repeat**  
4.   For each sensor  $s_i, i \in [1, N] \cap \mathbb{N}$   
5.     Evaluate  $RSSI$  values received by each  $CH$  in  $X_j$   
6.     Assign the sensor  $s_i$  to a  $CH$  according to Equation 5.7  
7.     Compute the fitness function according to Equation 5.15  
8.   Launch ABC algorithm for position update  
9. **Until** Maximum Cycle Number  
End

---

Besides that, since the number of clusters in this approach follows the Poisson distribution, and as demonstrated in [89], at the end of followers assignment, the number of sensors within each cluster will follow a geometric distribution, particularly in the case of data

packets retransmission (see Section 5.4.1.2). That combination of the Poisson and the geometric distributions is known as the geometric Poisson distribution that is usually used in the description of clusters and their members.

At the end of the network configuration, the *BS* informs each sensor, the cluster in which it belongs, i.e., the *id* of its corresponding *CH*, again by using a controlled flooding mechanism. Also, each *CH* receives the number of sensors in its cluster, the *id* of its next hop (i.e., the next *CH* or the *BS*) and optionally the *id* of its child *CH*. Again, the controlled flooding operation is executed during a certain time and after that time, each  $Cl_m$  starts the route discovery process that is highlighted in Section 5.3.3. Then, when that operation is completed, in order to save energy and as adopted by Labraoui et al. [12], the *CH* builds the TDMA slots time of each  $Cl_m$ . This is described in the following Section.

### 5.3.2.3 Network scheduling: energy saving method

The *MAC* layer protocol is quite important in the design of sensor networks, especially for energy saving. Fairness, latency and throughput are the main challenges in traditional *MAC* protocols. In addition to these challenges, sensor networks aim at being designed with power efficiency and scalability. The S-MAC protocol, which is a complex and powerful *MAC* protocol designed for WSNs, which allows sensors to periodically sleep, has been introduced in order to achieve these aims. Of course, in the S-MAC protocol, the self-configuration of sensors is well performed and energy saving is significant, but latency and throughput remain problems. However, adaptive listening improves it significantly [90].

In networks with clusters in which communications are achieved in a multi-hop manner, usually where there are  $N$  sender sensors and one receiver sensor, i.e., *many-to-one*, a good time synchronization among all sensors is important particularly for synchronous networks, i.e., there is not idle time for sensors, all sensors can send/receive packets in the same time. In the design of the proposed clustering approach, sensors' radios are not always turned on, as adopted by a wide number of clustering algorithms for sensors networks [14, 21, 23]. Also, sleep periods can be different from a sensor to another. Because of this fact, communications in the proposed sensor network especially inside the constructed clusters are asynchronous. So, in order to save energy, each sensor has a *TDMA* slot time in which it wakes up its radio for sending or receiving packets.

A number of works showed that asynchronous networks do not need complex and expensive synchronization needs. Interested readers may refer to [10]. In addition, because low-power operations are preferred in large-scale sensor networks, our design expects a medium access protocol that includes effective and low-power operations. That fact allows, the extension of the number of sensors. In this light, Polastre et al. [9] proposed a very simple and powerful implementation of *MAC* protocols for ad hoc networks called B-MAC and the

performance evaluation achieved by Karaba and Calle [8] shows its values added compared to others. Moreover, B-MAC alternates sleep and wake up periods during the network lifetime like other simplified medium access protocols designed for sensor networks.

Furthermore, B-MAC implements a sleep/wake up mechanism known as the low power listening and the protocol is reconfigurable by upper layers through the proposed flexible application interface. Our proposed protocol exploits the B-MAC features in the construction of *TDMA* slot time. However, for simplicity reasons, we assume that sensors can only access the channel during their allocated slot time unlike the B-MAC in which a method of periodically checking the channel for radio activity is implemented. Moreover, since our routing protocol is a cluster-based one, we introduced a mechanism in which each sensor can only receive packets from its  $Cl_m$  (see Section 5.3.3), this allows the avoidance of collisions. The implemented *TDMA* mechanism for energy saving is highlighted in the following paragraphs.

In each round, at the end of the clustering process, the network scheduling step takes place. After the route discovery process (see Section 5.3.3), it is now the time for *CH*s to build *TDMA* frames. In fact, during the steady state phase (see Section 5.3.2.4), each sensor has to gather information and transmit it to its *CH*, at its *TDMA* slots time. Of course, in WSNs design, the way as well as the time of transmitting the data depends on the application requirements. Indeed, tracking applications such as bush fire must immediately relay the reported temperature that is higher than the fixed threshold, whereas soil monitoring application, e.g., in irrigated agriculture, sensors can just need to report gathered data during a defined time. In our approach, when each cluster completed the route discovery process, each *CH* builds distinct *TDMA* slots for its followers by dividing the time into a different size of a fixed number (the number of  $Cl_m$  of each cluster) of slots, each slot unit lasts  $s_t \mu s$ .

Indeed, since sensors can communicate by using a multi-hop path, before switching off its radio, a sensor that is in the path of a given sensor has to wait a little extra time in which it receives and forwards all packets that must transit through it. In our approach, the path is seen as a tree where the *CH* is the root. A simple case is given in Illustration 5.2.

**Illustration 5.2.** For instance, we schematize a cluster  $c_1$  with the set of followers  $f(c_1) = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$  in Figure 5.3. So, depending on the number of hops, the size of each slot may differ. An illustration of the *TDMA* frame of the cluster  $c_1$  is given in Figure 5.4.

Concretely, a fixed slot time is first assigned to all leaf nodes, then progressively, the length of the slot time of the next hop is set to one unit of slot (its own slot) plus the number of its sensor child, i.e., the depth of the subpath, times the unit of slot. The length  $S_t$  of the slot time of an intermediary node is given by Equation 5.8,  $h$  is the number of child sensors.

Figure 5.3: Cluster presented like a k-way tree where the *CH* is the root and at each level of the tree, each sensor  $s_i \in c_1$  has no more than k children.

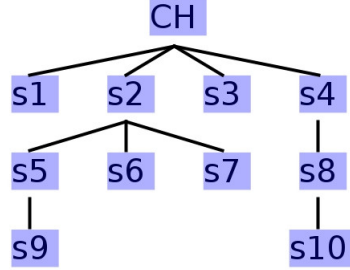
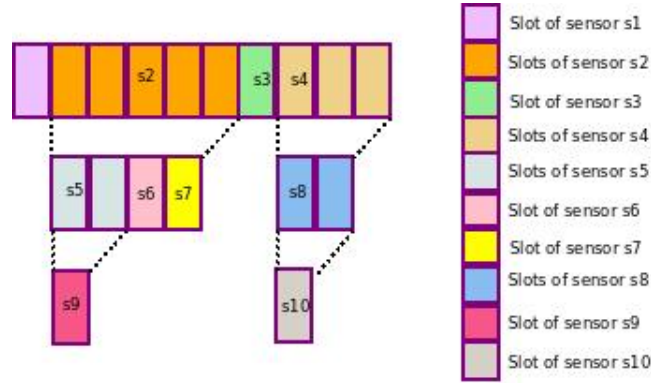


Figure 5.4: An illustration for the construction of the *TDMA* frame for the cluster  $c_1$ .



$$S_t = (h + 1) \times s_t \quad (5.8)$$

By assuming that, each sensor has to send its gathered data each  $u$  unit of time (a defined number of minutes or hours) and there is a small additional time  $\epsilon$  in  $\mu s$  for data acknowledgement messages, each sensor will switch off its radio at time  $w$  (see Equation 5.9):

$$w = u + S_t + \epsilon \quad (5.9)$$

This way of dividing the time periods for cluster scheduling (*TDMA*) will allow that, a sensor goes to sleep when it sends its packet to its parent node (next hop) and when it receives and forwards packets of its child nodes. The same mechanism, which also allows *CHs* to sleep in order to save their energy is implemented for inter-cluster communication. In fact, since each *CH* knows its next hop towards the *BS*, which is either another *CH* or the *BS* itself, we implemented also simplified *TDMA* mechanism for data transfer by *CHs* to the *BS*. Once it receives and aggregates all its followers' gathered data, the *CH* sends the aggregated data in a unique *D-AGG* message to the next hop toward the *BS*.



After that, it switches off its radio and wakes up at the start of the next  $u$  unit of time in order to receive a new sequence of gathered data from its followers. Indeed, each  $CH$  receives its slots length and the frequency at which it will wake up for the inter-cluster communication, from the  $BS$  at the start of each round.

Besides that, in order to avoid interference of signals that usually exist in the case of large scale sensor networks, in addition to the clear channel assessment feature, we use the effective collision avoidance mechanism that uses the signal strength ( $RSSI$ ) implemented in the B-MAC protocol. At the end of the creation of the  $TDMA$  frame, each  $CH$  sends the slot length of each of its followers again by using a controlled flooding mechanism. Then, the steady-state phase is launched.

#### 5.3.2.4 Steady-state

For each sensor, the steady-state phase consists of the use of its  $TDMA$  slots time to transmit the gathered data to its corresponding  $CH$ , either in a single-hop or in a multi-hop manner, depending on the density of the cluster and the location of the  $CH$ . This is well explained in Illustration 5.4 with an  $UML$  sequence diagram (see Section 5.3.3.2). When receiving all gathered data from its followers, the  $CH$  aggregates the received data and sends it to the  $BS$  by using the inter-cluster communication, i.e., by using intermediary  $CH$  or directly to the  $BS$  if the next hop of the considered  $CH$  is the  $BS$ .

#### 5.3.2.5 Fitness function derivation

The previously described clustering process in Section 5.3.2.2, aims at maximizing the network lifetime by well selecting  $CHs$ . In the selection of  $CHs$ , we take into account two criteria: the energy efficiency and the quality of service (QoS). The first constraint that has to be respected is the energy saving criterion while choosing a sensor as a  $CH_c$ . Indeed, since the  $CH$  is responsible for data aggregation and transferring all the  $Cl_m$  gathered data, more energy is used and these sensors' energy levels can decrease quickly. So, since in our network design there is no sensor with a particular power resource,  $CHs$  must be selected in an optimal way. Therefore, the evaluation of fitness considers the energy level of the  $CH_c$ , which should be greater than the fixed threshold value  $\varepsilon_t$ . Moreover, the proposed clustering algorithm aims at reducing the energy depletion that can be introduced during  $CHs$ ' operations, by minimizing the number of  $CHs$  since sensors' intra-cluster communication is achieved in a multi-hop manner. So, we define the energy criterion of the fitness function  $f^{energy}$  in Equation 5.10.

$$f^{energy} = p \times \frac{k}{m} \quad (5.10)$$

Where the constant  $p$  is a proportionality constant and  $k$  is the number of distinct  $CHs$

in the position vector of the ABC algorithm. Like adopted by Heinzelman et al. [25], the optimal number of CHs should be approximately set to 5% of all nodes. Since in our proposed approach the fitness is used only for a comparison purpose, without loss of generality, we assume that  $p = 1$ .

The TDMA based multi-hop approach adopted for the intra-cluster communication could slightly prolong the cluster scheduling time for a cluster with more  $Cl_m$  than others. This might introduce a supplementary delay. So, especially for the communication link quality, the QoS is considered as the second criterion, which aims at maximizing the packet delivery to the BS while reducing the delay. Thereby, in order to maximize the cluster communication link quality, let us define a quality parameter indicator  $QP_{Cl_{m_j}}(x)$ ,  $j \in [1, m]$ ,  $x \in [1, N(\mathcal{A}_j)]$  of a member of the cluster with the head  $c_j \in \mathfrak{C}(ch)$  in Equation 5.13. This quality indicator includes  $\mathfrak{R}_m(Cl_{m_j}(x))$  (see Equation 5.11), which is the best RSSI value received by the  $Cl_{m_j}(x)$ , and the worst value is  $\mathfrak{R}_w(Cl_{m_j}(x))$  (see Equation 5.12).

$$\begin{aligned} \mathfrak{R}_m(Cl_{m_j}(x)) &= \max_{j \in [1, m]} \left( \mathfrak{R}(Cl_{m_j}(x), i) \right) \\ i &\in [1, N(\mathcal{A}_j)] \text{ and } i \neq x \end{aligned} \quad (5.11)$$

$$\begin{aligned} \mathfrak{R}_w(Cl_{m_j}(x)) &= \min_{j \in [1, m]} \left( \mathfrak{R}(Cl_{m_j}(x), i) \right) \\ i &\in [1, N(\mathcal{A}_j)] \text{ and } i \neq x \end{aligned} \quad (5.12)$$

$$QP_{Cl_{m_j}(x)} = \left[ Re_m(Cl_{m_j}(x)) \times \frac{1}{Re_m(Cl_{m_j}(x)) - Re_w(Cl_{m_j}(x))} \right] \quad (5.13)$$

Then, we define the QoS criterion  $f^{QoS}$  in Equation 5.14 in such a way that the quality of the communication link within the cluster becomes maximal by minimizing the worst link qualities.

$$f^{QoS} = \sum_{x=1}^{N(\mathcal{A}_j)} QP_{Cl_{m_j}(x)} \quad (5.14)$$

We bring back the fitness function to a linear programming problem. In fact, we need to minimize the energy loss and the worst link quality. To achieve this, the two proposed criteria in Equation 5.10 and Equation 5.14 are used to build a multi-objective fitness function. For simplicity reasons, we use the weighting parameter  $\alpha$  and  $\beta$  in order to transform these objective functions into a single fitness function. Thus, let us give the fitness function that includes the energy level criterion, the QoS criterion and the weighting parameters  $\alpha$  and  $\beta$ , as a linear programming problem in Equation 5.15.

$$\text{Minimize } fit = \alpha \times f^{energy} + \beta \times f^{QoS} \quad (5.15)$$

Indeed, since the objective of the fitness function is to minimize the energy consumption and to minimize the use of worst link, i.e., maximize the QoS, we use  $\beta$  to control the QoS and  $\alpha$  to control the energy consumption. For a good weighting,  $\alpha$  and  $\beta$  are subjected to the constraints given in Equation 5.16 and Equation 5.17.

$$\alpha = 1 - \beta \quad (5.16)$$

$$0 < \beta < 1 \quad (5.17)$$

### 5.3.3 ABC-SD routing mechanism

Our proposed routing algorithm adopts an oblivious routing in which the routing path is discovered by each  $Cl_m$  before starting the steady-state phase. These paths cannot change throughout the round. The design of the routing algorithm is described before presenting the route discovery process in Section 5.3.3.2.

#### 5.3.3.1 The packet format

In the design of the proposed routing protocol, we adopt a unique packet format. The packet is structured into eight fields: packet  $id$ , packet type, source sensor  $id$ ,  $CH$   $id$ , the number of hops in the path, sum of the remaining energy of the path, minimum energy of the path and data. The packet format is given in Figure 5.5 and the complete description of the packet fields is given in Table 5.2. The fields  $hopC$  and  $sumE$  are initially set to zero.  $minE$  initially contains the remaining energy of the sender, i.e.,  $E_r(srcId)$ .

Figure 5.5: The adopted unique packet format with its eight fields for the data transfer over the network.

$id$	$type$	$srcId$	$destId$	$hopC$	$sumE$	$minE$	$data$
------	--------	---------	----------	--------	--------	--------	--------

Each packet in the whole network must have a unique identifier. Permitting a unique identification of a packet is necessary for the used controlled flooding mechanism. So, each packet generated by a sensor has an  $id$  compound of the sensor  $id$  that is concatenated with a sequence number. The field  $data$  is structured differently depending on the type of the packet. The used messages are described in Table 5.3.

The  $H\_MSG$  message is executed in the same mind as the  $hello$  message of the B-MAC protocol. For a  $P\_DISC$  message, the field  $data$  initially contains the  $srcId$ . In the  $P\_ACK$  message, fields  $id$ ,  $type$ ,  $srcId$  and  $destId$  are modified as explained in Section

Table 5.2: Descriptions of the fields in the packet format.

Field	Description
<i>id</i>	is the unique identifier assigned by sensor to a packet.
<i>type</i>	is the type of the message.
<i>srcId</i>	is <i>id</i> of the source sensor.
<i>destId</i>	is the <i>id</i> of the destination sensor. <i>destId</i> always contains the <i>id</i> of the <i>CH</i> .
<i>hopC</i>	is the number of hops crossed by the packet before reaching the destination.
<i>sumE</i>	is the sum of the remaining energy of all sensors in the path.
<i>minE</i>	is the energy of the sensor that has the minimum remaining energy in the path.
<i>data</i>	is the data field.

Table 5.3: The used messages in the ABC-SD routing algorithm.

Message	Description
<i>H_MSG</i>	is a <i>hello</i> neighborhood discover message.
<i>P_DISC</i>	is a path discovery message.
<i>P_ACK</i>	is a path acknowledgement message sent by the <i>CH</i> to a source sensor.
<i>R_MSG</i>	is a route message that contains the chosen path from a <i>Cl<sub>m</sub></i> and its <i>CH</i> .
<i>D_MSG</i>	is a data message sent by each <i>Cl<sub>m</sub></i> to its <i>CH</i> .
<i>D_ACK</i>	is a data acknowledgement message sent by the <i>CH</i> to the sender sensor.
<i>D_AGG</i>	is an aggregated data by a <i>CH</i> for sending to the <i>BS</i> .

**5.3.3.2.** In addition to the information to transfer, the path toward the *CH* is concatenated with the information in the *data* field in *D\_MSG* messages. The *D\_ACK* message is transformed into a simple *ACK* message like in the B-MAC implementation.

### 5.3.3.2 Route discovery mechanism

When the network is initialized and the clustering process is completed, each *Cl<sub>m</sub>* starts a route discovery in order to construct a pre-determined path toward its *CH*. To start the route discovery process, each *Cl<sub>m</sub>* broadcasts a *P\_DISC* message in the network with its *CH id* as the destination sensor. When a given *Cl<sub>m</sub>* receives a *P\_DISC* message, it'll perform the following actions:

1. increments the number *hopC* of hops.
2. updates the sum of the remaining energy in the path *sumE*.
3. if the *Cl<sub>m</sub>* remaining energy is lower than that in the current *minE*, replace the value of *minE* by the actual *Cl<sub>m</sub>* remaining energy.
4. adds the *Cl<sub>m</sub> id* in the *data* field.
5. rebroadcasts the *P\_DISC*.

In this way, the packet will reach the *CH*. Certainly, because clusters are not really disjoint in practice, a *Cl<sub>m</sub>* of a given cluster may receive a *P\_DISC* packet from a *Cl<sub>m</sub>* of a close

cluster, if these sensors are in the same communication range. In such a case, by using some features of the B-MAC protocol, the receiver will consider the packet only if the *destId* field contains its *CH id*, i.e., if they have the same *CH*. When receiving the *P\_DISC* packet, the *CH* will launch a path acknowledgement packet *P\_ACK* to the sender *srcId*. The *P\_ACK* packet is built from the received *P\_DISC* packet by altering *id*, *type*, *srcId* and *destId* fields. The value of the *id* field in the *P\_ACK* packet structure is assigned by using the *id* of the received *P\_DISC* packet in such a way that a  $Cl_m$  (i.e., the sender) can recognize that, it is the path acknowledgment message of its sent path discovery message. Of course, since the *CH* may receive many versions of a same initial *P\_DISC* message and launch path acknowledgement messages for them, the source sensor may therefore receive more than one *P\_ACK* packet at the end of the route discovery time. Well, in such a situation, the source sensor has to choose the best path.

The choice of the best path is achieved by making a tradeoff between the number of hops in the path and the path's available energy. In order to realize that, we introduced a non-linear choice parameter  $\zeta$  in the design of the proposed routing algorithm that aims at choosing the path with at least hops and better remaining energy. So, let us define the cost-based function  $\zeta(p(l))$  that evaluates the energy strength of a path  $p(l)$  in Equation 5.18.

$$\zeta(p(l)) = \left[ \left( n(p(l)) \right)^{-1} \times \frac{\text{sum}E - \left( n(p(l)) \right)^2}{\text{min}E - n(p(l))} \right] \quad (5.18)$$

Where  $l \in [1, N(\mathcal{A}_j) \cap \mathbb{N}] : j \in [1, m]$ .  $l$  is the  $l^{\text{th}}$  *P\_ACK* packet received by the *P\_DISC* sender, i.e., the  $Cl_m$  and  $n(p(l))$  is the number of hops in the path  $p(l)$ . Finally, when the best paths are chosen, each  $Cl_m$  sends a *R\_MSG* to its *CH* by using the controlled flooding mechanism implemented in the proposed protocol. While receiving all *R\_MSG* from its followers, the *CH* can now proceed to the construction of *TDMA* frames as explained in Section 5.3.2.3. We give in the following, the algorithm (see Algorithm 5.2) for route discovery, which is executed by each  $Cl_m$ .

In this way, unlike other bio-inspired routing algorithms for WSNs like those proposed in [1, 16, 21], each sensor  $s_i$ ,  $i \in [1, N]$  keeps just one route towards its *CH*. Of course, while clustering and routing processes are executed at the start of each round, the  $Cl_m$  established route may not be the same. An illustration is given below.

**Illustration 5.3.** For instance, in Figure 5.6, a sequence of performed actions for establishing routes is described. In this illustration, we take a simple case in which a cluster with the head  $s_2$  and its followers  $f(s_2) = \{s_1, s_3, s_4\}$ . During the time  $t_1$ , each follower will perform a number of actions (send/receive messages) in order to receive one or more paths toward  $s_2$  and during the time  $t_2$ , it'll choose the best path and informs  $s_2$  the chosen

**Algorithm 5.2** The ABC-SD route discovery algorithm

---

```

Begin
1. broadcasts( $P\_DISC$ )
2. while ( $t_1$ )
3.   if (receive( $P\_DISC$ )) then
4.     hopC = hopC + 1
5.     sumE = sumE +  $E_r(Cl_m)$ 
6.     if ( $E_r(Cl_m) < \text{minE}$ ) then
7.       minE =  $E_r(Cl_m)$ 
8.       data = data  $\odot Cl_m$ 
9.       rebroadcasts( $P\_DISC$ )
10.    else if (receive( $P\_ACK$ )) then
11.      Compute the path strength according to Equation 5.18
12.  while ( $t_2$ )
13.    if (receive( $P\_ACK$ )) then
14.      Compute the path strength according to Equation 5.18
15.    else if (receive( $P\_DISC$ )) then
16.      destroy( $P\_DISC$ )
17. Choose the best path among all computed path strength
18. Launch a  $R\_MSG$  message.
End

```

---

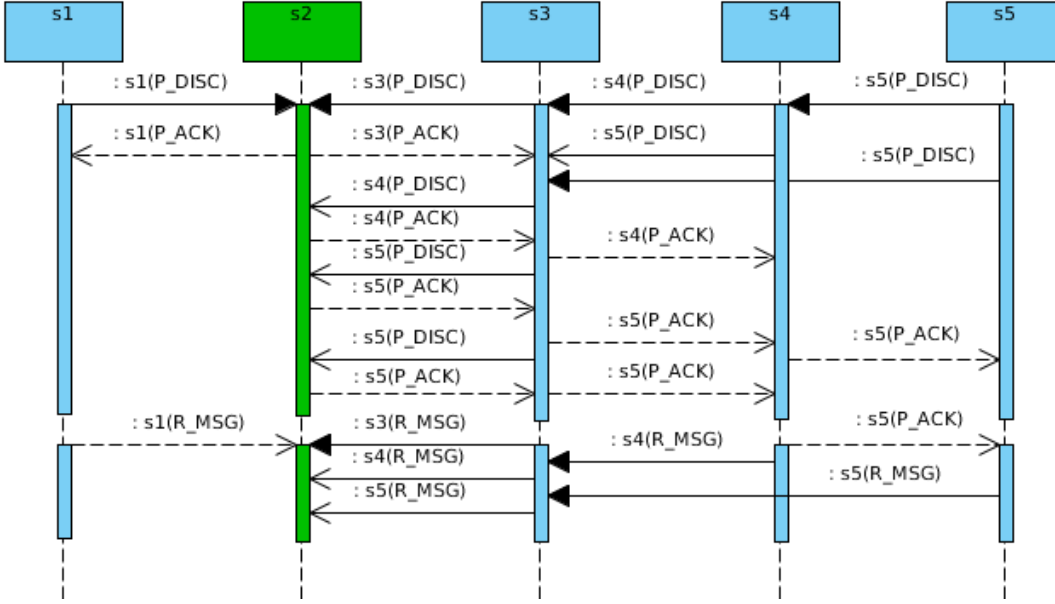
route among the received paths. In the figure, the filled arrow  $\blacktriangleright$  represents the send of a new message. The open arrow  $\longrightarrow$  represents the forwarding message. The dashed arrow  $--\rightarrow$  represents the path acknowledgement message both in sending and forwarding.

In this instance, let's take the case of sensor  $s_5$ . At the start of  $t_1$ , the sensor  $s_5$  sends the message  $s_5(P\_DISC)$ . Sensors  $s_4$  and  $s_5$  will then receive its sent discovery message since they are in its neighbor's table. So, the sensor  $s_4$  will forward the message  $s_5(P\_DISC)$  to its next hop, i.e.,  $s_3$  and the sensor  $s_3$  will then forward it to their head  $s_2$ . In the same way, since the sensor  $s_3$  also directly received the message  $s_5(P\_DISC)$ , it'll also forward again that message to their head  $s_2$ . Thus, when receiving these discovery messages the head  $s_2$  will launch two distinct acknowledgement messages  $s_5(P\_ACK)$  to  $s_5$  since it received two distinct discovery messages from  $s_5$ , from two different paths. To conduct well the steps in Algorithm 5.2, we took a case that one of these acknowledgement messages reaches the sensor  $s_5$  during the time  $t_2$ . At last, the sensor  $s_5$  will inform its head  $s_2$  the chosen path by launching the route message  $s_5(R\_MSG)$ .

### 5.3.3.3 Data transfer and acknowledgement mechanism

At the end of paths discovery process and the establishment of routes between each  $Cl_m$  and their  $CH$ , then comes the time of data gathering and transferring, which is explained in Section 5.3.2.4. In fact, during its slots time (see Equation 5.9), a sensor will send its gathered data to its  $CH$ . In the simulation, we stored the total number of received data from each  $Cl_m$  in each round, in order to analyse the energy efficiency and throughput metrics that is explained in Section 5.4.2. An illustration of this process is given hereinafter.

Figure 5.6: Route discovery in the ABC-SD.



**Illustration 5.4.** To illustrate the data transmission step, i.e., the steady-state, we provide an UML sequence diagram in Figure 5.7. In this instance, let's consider a case of the BS and two clusters represented by heads  $s_2$  and  $s_6$ . In the diagram, the filled arrow  $\rightarrow$  represents the send of a new message. The open arrow  $\rightarrow$  represents the forwarding message. The dashed arrow  $--\rightarrow$  represents the data acknowledgement message both in sending and forwarding.

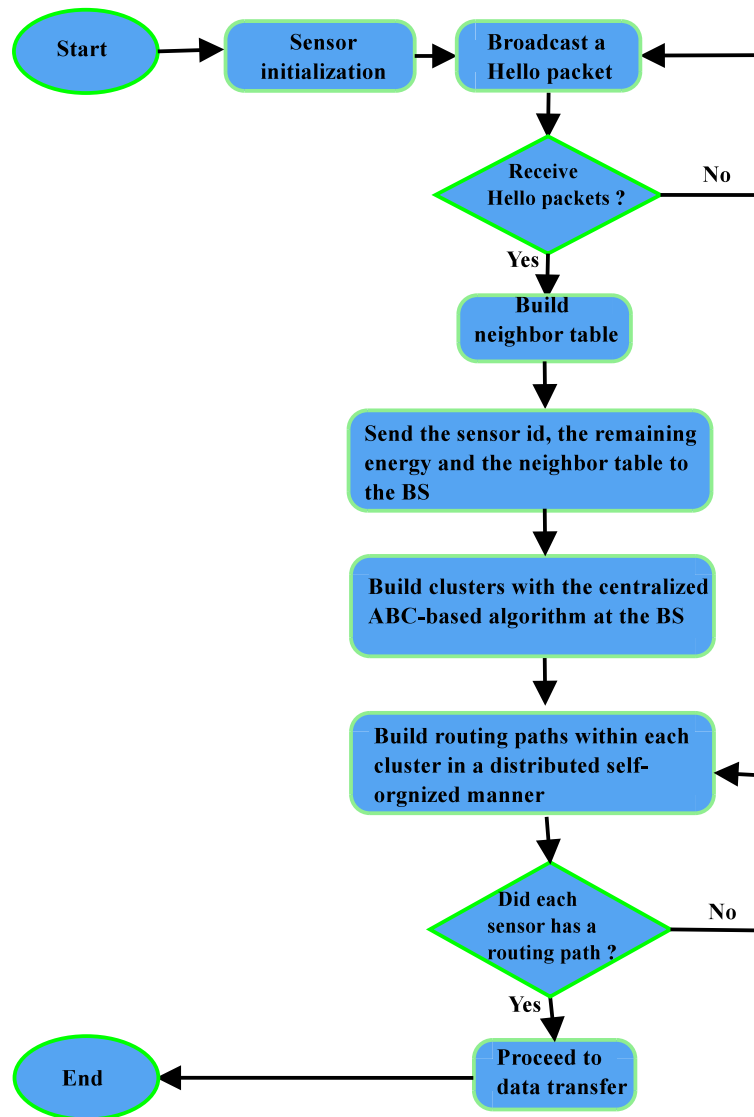
For simplicity reasons, we illustrate the inter-cluster communication just in the cluster with the head  $s_2$ . So, let's define  $f(s_2) = \{s_3, s_4, s_5\}$ . In the sequence diagram, let's take the case of  $s_4$ . The sensor  $s_4$  sends its gathered data to the head  $s_2$  by launching the data message  $s_4(D\_MSG)$ . Then its next hop, i.e.,  $s_3$  will receive and forward that message to their head  $s_2$ . When receiving the message,  $s_2$  will launch a data acknowledgement message to  $s_4$ . When receiving all data messages from their followers during their cluster's TDMA frames, the heads  $s_2$  and  $s_6$  have to switch off their radios and wake it up at the appropriate time for the inter-cluster communication.

Since data packets are exclusively sent towards the discovered and registered path and because our design aims at providing a WSN with less rate of data loss, we implement a simplified acknowledgement message of a sent data packet. Indeed, during data transfer on a given path, when a cluster member  $Cl_{m_j}$  receives a  $D\_MSG$  from a  $Cl_{m_{j-1}}$  in the same path  $p(l)$ , the  $Cl_{m_j}$  sends back a  $D\_ACK$  message to  $Cl_{m_{j-1}}$  during the additional time  $\epsilon$  (see Equation 5.9). When not receiving the  $D\_ACK$  message after the first part of  $\epsilon$  (in our implementation,  $\epsilon$  is divided in 7 different parts),  $Cl_{m_{j-1}}$  will retransmit the





Figure 5.9: Flow chart of the ABC-SD cluster-based routing protocol.



to some proposed evaluation metrics.

#### 5.4.1 Theoretical analysis of energy consumption

##### 5.4.1.1 Energy consumption derivation

According to the proposed design, the consumed energy at each round includes the energy of bootstrapping, the energy of paths discovery and the energy of data transfer. At the beginning of each round, the bootstrapping process in which each sensor  $s_i$ ,  $i \in [1, N] \cap \mathbb{N}$  discovers its neighbors takes place. Each sensor has to broadcast a  $H\_MSG$  packet that will be received by a number of sensors in its communication range. A priori, we cannot know the number of neighbors of a given sensor since the deployment is done randomly.

Despite that, we suppose that, in the adopted sensors' deployment strategy, the maximum number of neighbors  $\beta$  is given in Equation 5.19.

$$\beta = \left\lceil N \times \frac{r}{\lambda_2(\mathcal{D})} \right\rceil \quad (5.19)$$

By taking that into account, the expended energy  $E_{boot}(s_j)$  by a sensor  $s_j$ ,  $j \in [1, N] \cap \mathbb{N}$  in the bootstrapping process is equal to the energy consumed  $E_i$  for sending the  $H\_MSG$  plus the energy consumed in the reception of  $\beta$   $H\_MSG$  packets from its neighbors (see Equation 5.20). Hence, the overall network energy consumed  $E_{boot}$ , i.e., the energy consumed by all sensors, in the bootstrapping process is given in Equation 5.21.

$$E_{boot}(s_j) = E_i \times (1 + \beta) \quad (5.20)$$

$$\begin{aligned} E_{boot} &= \sum_{j=1}^N E_{boot}(s_j) \\ &= N \cdot E_i \times (1 + \beta) \end{aligned} \quad (5.21)$$

Then, the energy consumption in paths discovery is computed according to the energy expended while sending and receiving  $P\_DISC$  and  $P\_ACK$  packets. Let  $E_{pdisc}(Cl_{m_j})$  and  $E_{pack}(Cl_{m_j})$  be the expended energies respectively in sending and receiving a  $P\_DISC$  packet (respectively, a  $P\_ACK$ ) by the  $Cl_{m_j}$ , which is a follower of the head  $c_j \in \mathfrak{C}(ch)$ ,  $j \in [1, m] \cap \mathbb{N}$ . In a worst case, i.e., by assuming that a  $P\_DISC$  packet sent by  $Cl_{m_j}$  will be received by all followers of the head  $c_j$ , the expended energy  $E_{pdisc}(Cl_{m_j})$  (see Equation 5.22) is equal to the energy consumed  $E_i$  by the  $Cl_{m_j}$ , in the broadcast of the  $P\_DISC$  packet and the energy consumed in the receiving and rebroadcasting (i.e.,  $2 \cdot E_i$ ) of the  $P\_DISC$  by followers, i.e., by  $Cl_{m_j} \in f(c_j)$ .

$$E_{pdisc}(Cl_{m_j}) = E_i \times (1 + 2 \cdot N(\mathcal{A}_j)) \quad (5.22)$$

For all members of the cluster with the head  $c_j$ , the expended energy by  $P\_DISC$  packets is computed as shown in Equation 5.23 by multiplying the energy expended by one follower with the number of followers of the head  $c_j$ . Then let us give in Equation 5.24,  $E_{pdisc}$ , which is the sum of the expended energy by each cluster, i.e., the energy expended by each  $s_i$ ,  $i \in [1, N] \cap \mathbb{N}$ .

$$\begin{aligned} E_{pdisc}(c_j) &= N(\mathcal{A}_j) \times E_{pdisc}(Cl_{m_j}) \\ &= E_i \cdot N(\mathcal{A}_j) \times (1 + 2 \cdot N(\mathcal{A}_j)) \end{aligned} \quad (5.23)$$

$$\begin{aligned}
E_{pdisc} &= \sum_{j=1}^m E_{pdisc}(c_j) \\
&= E_i \cdot \sum_{j=1}^m N(\mathcal{A}_j) \left(1 + 2 \cdot N(\mathcal{A}_j)\right)
\end{aligned} \tag{5.24}$$

Unlike the energy expended by the broadcast and reception of  $P\_DISC$  packets, we compute the exact value of the energy expended for the  $P\_ACK$  packets, since the acknowledgement paths are known. So, by using the number of hops  $n(p(l))$  of each path  $p(l)$  taken by  $P\_ACK$  packets, the energy expended in an acknowledgement of a  $P\_DISC$  packet initiated by the head  $c_j$  to a follower  $Cl_{m_j}$  is given in Equation 5.25.

$$E_{pack}(Cl_{m_j}) = E_i \times \left(1 + 2 \cdot (n(p(l)) - 2)\right) \tag{5.25}$$

Where  $l \in [1, N(\mathcal{A}_j) \cap \mathbb{N}]$ . By considering all followers of the head  $c_j$ , i.e., each  $Cl_{m_j} \in f(c_j)$ , the expended energy for  $P\_ACK$  packets by each cluster is given in Equation 5.26. In addition, the expended energy by the network in  $P\_ACK$  packets, which corresponds to the sum of each cluster's expended energy in  $P\_ACK$  packets is given in Equation 5.27. So, the energy expended in the routes establishment ( $E_{re}$ ) is given in Equation 5.28.

$$\begin{aligned}
E_{pack}(c_j) &= N(\mathcal{A}_j) \times E_{pack}(Cl_{m_j}) \\
&= E_i \cdot N(\mathcal{A}_j) \left(1 + 2 \cdot (n(p(l)) - 2)\right)
\end{aligned} \tag{5.26}$$

$$\begin{aligned}
E_{pack} &= \sum_{j=1}^m E_{pack}(c_j) \\
&= E_i \cdot \sum_{j=1}^m N(\mathcal{A}_j) \left(1 + 2 \cdot (n(p(l)) - 2)\right)
\end{aligned} \tag{5.27}$$

$$\begin{aligned}
E_{re} &= E_{pdisc} + E_{pack} \\
&= 2E_i \cdot \sum_{j=1}^m N(\mathcal{A}_j) \cdot \left(N(\mathcal{A}_j) + n(p(l)) - 1\right)
\end{aligned} \tag{5.28}$$

Furthermore, we can compute the expended energy in the routing of data packets since the number of hops is known in advance when a  $Cl_m$  tries to send its gathered data to its  $CH$ . Thereby, the resulting energy consumption in a data packet transfer toward a path  $p(l)$  is equal to the energy expended by the  $Cl_m$  in the sending and the energy expended by the  $CH$  at reception plus the energy of reception and forwarding at the intermediary sensors in the path, plus the energy of sending and receiving the  $D\_ACK$  messages. (see Equation 5.29). So, by assuming that each  $Cl_m$  has to send  $n_{data-t}$  data packets to its  $CH$  in each round, the energy consumption  $E_{data-t}$  in a best situation for data packets transfer for the whole network is given in Equation 5.30.

$$\begin{aligned}
E_{data-t}(Cl_{m_j}) &= 2 \cdot E_i \left( (n(p(l)) - 2) + 1 \right) + 2 \cdot E_i (n(p(l)) - 1) \\
&= 4 \cdot E_i (n(p(l)) - 1)
\end{aligned} \tag{5.29}$$

$$\begin{aligned}
E_{data-t} &= n_{data-t} \sum_{j=1}^m \sum_{l=1}^{N(\mathcal{A}_j)} E_{data-t}(Cl_{m_j}) \\
&= 4 \cdot E_i \cdot n_{data-t} \sum_{j=1}^m \sum_{l=1}^{N(\mathcal{A}_j)} (n(p(l)) - 1)
\end{aligned} \tag{5.30}$$

Finally, according to the aforementioned energy derivations, the energy expenditure ( $E_{exp}$ ) in each round, which includes the expended energy in the bootstrapping, routes establishment and data packets transfer, in a good network condition is given in Equation 5.31.

$$\begin{aligned}
E_{exp} &= E_{boot} + E_{re} + E_{data-t} \\
&= E_i \cdot \left( N \times (1 + \beta) + 2 \cdot \sum_{j=1}^m N(\mathcal{A}_j) \cdot (N(\mathcal{A}_j) + n(p(l)) - 1) \right. \\
&\quad \left. + 4 \cdot n_{data-t} \sum_{j=1}^m \sum_{l=1}^{N(\mathcal{A}_j)} (n(p(l)) - 1) \right)
\end{aligned} \tag{5.31}$$

#### 5.4.1.2 Packets retransmission derivation

The previous energy derivation for data packets transmission is achieved by assuming that all data packets are normally routed towards clusters. In practice, due to the hostile environment in which WSNs are generally deployed, for instance, in a forest or in an environment with many obstacles like buildings, loss of packets during the data transfer may occur. Therefore, according to the adopted packet retransmissions mechanism previously described in Section 5.3.3.3, let's define the probability  $P(p(l), k)$  (see Equation 5.32) of success on each data packet transmission on the path  $p(l)$ .

$$P(p(l), k) = e^{-rn(p(l))} \tag{5.32}$$

Where  $r > 0$  is the density parameter of the used Poisson distribution in Section 5.3.2.2 for the distribution of clusters on the sensing environment. Then in the case of a data packet retransmission, the modeling of the energy consumed in a certain number of retransmissions depends on the probability (see Equation 5.33) that the  $k^{th}$  retransmission leads to the first successful transmission. That probability  $P(X = k)$  is given such that the number of data packet retransmissions follows the geometric distribution.

$$P(X = k) = \left( 1 - P(p(l), k) \right)^{k-1} \cdot P(p(l), k) \tag{5.33}$$

Where  $X$  is the random variable that represents the number of successes in a data packet delivery. Therefore, the energy consumption  $E_{data-r}(Cl_{m_j})$  for  $n$  retransmissions of one data packet sent by  $Cl_{m_j}$  can be expressed as done in Equation 5.34. Hence, in each round, the energy consumption  $E_{data-r}$  for the retransmission of  $n_{data-t}$  data packets for the whole network is given in Equation 5.35. In the routing algorithm, the maximum number of retransmissions is set to 3.

$$E_{data-r}(Cl_{m_j}) = \sum_{k=1}^n \sum_{l=1}^{N(A_j)} \left\{ \left( 1 - P(p(l), k) \right)^{k-1} \cdot P(p(l), k) \right. \\ \left. \times \left( E_{data-t}(Cl_{m_j}) + (k-1)E_i \right) \right\} \quad (5.34)$$

$$E_{data-r} = n_{data-t} \sum_{j=1}^m E_{data-r}(Cl_{m_j}) \quad (5.35)$$

This energy derivation provides a theoretical overview of the overall network energy consumption in each round. Since, we aim at providing a WSN with high throughput, i.e., with a maximum number of delivered packets, the introduced packets acknowledgement and the retransmission mechanism have added a supplementary energy consumption. But, as we observed in simulation, the proposed clustering routing algorithm significantly improves the overall performance of the considered WSNs.

#### 5.4.2 Metrics of the performance evaluation

To evaluate the performance of the proposed algorithm, we analyzed the following metrics:

##### 5.4.2.1 Network lifetime

Since the lifetime is an important characteristic commonly analyzed in sensor networks performance evaluation, in the simulation we observed 3 metrics to evaluate the network lifetime of our algorithm:

- *Energy consumption.* It is a metric that allows us to analyse the average energy expended by each  $s_i$ ,  $i \in [1, N]$  during the simulation times.
- *Energy efficiency.* This metric is measured by evaluating the ratio of the total number of delivered packets to the *BS* and the overall network's consumed energy during the simulation times. At a given simulation time, that ratio is calculated according to the formula given in Equation 5.36. We use this metric to analyze the network lifetime as well as the throughput.

$$EE = \frac{\sum \text{data packets delivered to the } BS}{\sum \text{sensor's consumed energy}} \quad (5.36)$$

- *First sensor dead.* This metric allows us to know the amount of time that a sensor of our WSN design will be fully operative by evaluating the time until the first dead of a sensor during the simulation times.

### 5.4.2.2 Throughput

Given the fact that a lot of applications of WSNs need routing protocols with a high rate of successful packets delivered to the *BS*, we use two metrics in order to analyse the throughput:

- *Amount of delivered packets.* This metric is used to evaluate the amount of packets received by the *BS* at the end of the simulation times.
- *Packet loss rate.* This metric is calculated according to the formula given in Equation 5.37 in order to know the rate of packet losses caused by the designed routing algorithm.

$$PL = 1 - \frac{\sum \text{received data packets by the } BS}{\sum \text{sent data packets to the } BS} \quad (5.37)$$

### 5.4.2.3 Network coverage

The network coverage is an important characteristic in sensor networks, especially in WSNs requiring a high availability of gathered information. Even if sensors are well distributed on the sensing environment, unclustered sensor nodes cause the loss of information on their coverage area. To evaluate the performance in terms of network coverage, we reported the number of unclustered sensors in each round during the simulation time.

## 5.4.3 Simulation and discussion

We evaluated the proposed ABC-SD protocol by conducting series of intensive experiments in various scenarios. To be closer to the reality, we exploited the behaviors of the Chipcon CC2420 that is briefly described in Section 5.2.1 and its complete data sheet can be found in [88]. In this simulation, we used the realistic sensors' characteristics, the wireless channel and radio models included in the *Castalia 3.2 simulator*. *Castalia* is a simulator for WSNs and generally, for networks of low-power embedded devices, which is based on the *OMNet++ platform*. Since, the B-MAC is not found by default in the node communication module of the *Castalia simulator*, for this simulation, we have integrated a separated implementation in C++ of the B-MAC.

Unlike the bio-inspired clustering and routing approaches proposed in [16, 23], sensors of the proposed WSN model are homogeneous and can perform any task in the network. We have not used some special sensors with particular behaviors that act as pre-deployed gateways. It is assumed that all sensors are locations unaware. Indeed, all sensors of our network model are not equipped with a self-locating hardware such as the global positioning system. In the proposed clustering algorithm, we use an initial population of 50 bees and

we run the positions update of the ABC algorithm under a maximum of 500 cycles. The ABC swarm parameters are summarized in Table 5.4.

Table 5.4: The used ABC parameters for simulating the ABC-SD.

Parameter	Value
Number of cycles	500
Population size	50
Limit for neighborhood search	20

Experiments were carried out with randomly and uniformly deployed stationary sensors on a square field of  $100 \times 100 m^2$  that represents the sensing area. The *BS* is positioned at the coordinate (0;0) of the field. The initial energy of each sensor is equal to the energy of two AA batteries, which is set to  $18720 J$ . To evaluate the powerful and the full adaptability of the ABC-SD protocol, we consider four scenarios of networks in which each sensor network has a size of  $100 \times n$  :  $n = 1, 2, 3, 4$ . In each considered network scenario, the proposed algorithm is executed during  $5000 seconds$ . Since the number of sensors increases, in order to minimize the effect of communication overhead, the length of each round is set to  $500 seconds$  and the *TDMA* slot length is taken  $2/4$  as of a second. The obtained results in both clustering and routing are presented in a combined output. The complete simulation parameters are given in Table 5.5.

Table 5.5: Adopted Simulation parameters in the ABC-SD.

Parameter	Value
Square ( $m^2$ )	$100 \times 100$
<i>BS</i> location	(0;0)
Number of sensors	100, 200, 300, 400
Maximum number of CHs	5%
Initial Energy (J)	18720
$\epsilon_t$ (J)	4680
Radio frequency (GHz)	2.4
Packets length (Bytes)	128
Packets rate (packet/s)	1
Transmission speed (Kbps)	250
Transmission range (m)	75
TDMA Slot time (s)	0.4
CDMA MAC protocol	B-MAC
Times (s)	5000
Round length (s)	500

In order to highlight the value added introduced by the proposed clustering and routing approaches, the performance of the ABC-SD protocol is compared with some well known cluster-based protocols for WSNs:

- The LEACH [25] protocol that is the widely used in comparison purpose, the most popular and completely distributed clustering algorithm for WSNs;

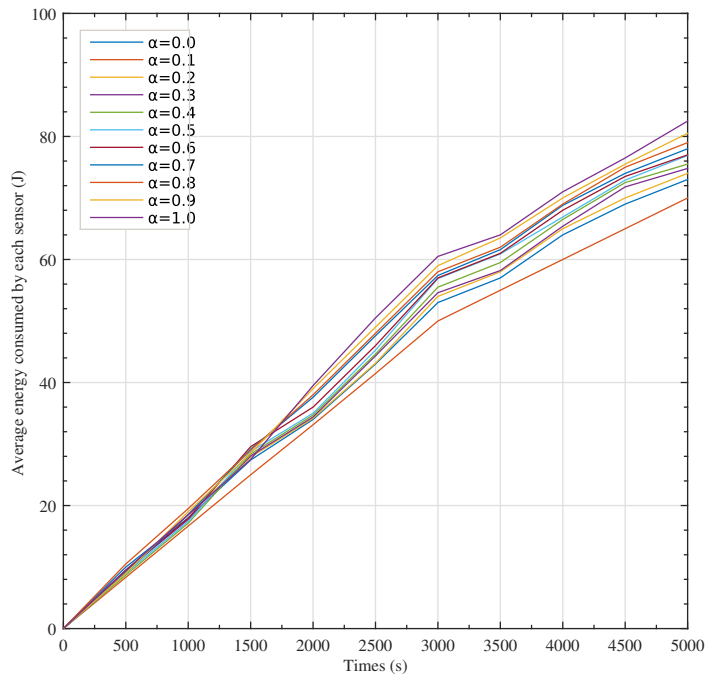
- The LEACH-C [27] protocol, which is the centralized version of the LEACH protocol;
- The PSO-C [22] protocol that is a completely centralized swarm based approach, which is inspired by social behaviors of bird flocking and fish schooling;
- The PSO-HC [29] protocol that is a hierarchical bio-inspired clustering approach for WSNs.
- The ABC-C [23] protocol that is a centralized clustering protocol which is inspired by the foraging behaviors of honeybees.

In each network scenario, we proceeded to 10 runs of these aforementioned protocols, including the ABC-SD. In each simulation run, 4 different network topologies are generated, totaling 40 network topologies. Each result presented hereafter is the average of outputs obtained in these runs. Also, all results are given at 99% confidence interval. Before presenting the results, let us evaluate by simulation the best value of the weighting parameter  $\alpha$  used in the fitness function (*see Section 5.3.2.5*), which maximizes the performance of the ABC-SD protocol. To do it, eleven distinct values of  $\alpha \in [0; 1]$  have been tested. In order to choose the best one, we measure the average energy consumed by each sensor during the simulation time. While considering the network scenario with 100 sensors, it can be observed in obtained curves (*see Figure 5.10*) that the best value of  $\alpha$  is 0.1. Also, with the 3 other network scenarios, we found the same value of  $\alpha$ .

Globally, we measured the three principal metrics described above in *section 5.4.2*: network lifetime, the throughput and the network coverage. The network lifetime is studied here by the measurement in each network scenario, of the energy consumption of each sensor, the energy efficiency and the time at which the first sensor dies. In their experiments, Latiff et al. [22] as well as Karaboga et al. [23] showed that the PSO-C protocol performs better than the LEACH and the LEACH-C. In the same mind, in their proposed protocol, Elhabyan and Yagoub [29] reported that, the PSO-HC protocol provides better results than the compared protocols. The results obtained by the conducted experiments are given in *Figures 5.11 - 5.19*. These graphics show the obtained results of the ABC-SD including the ABC-C, PSO-HC, PSO-C, LEACH-C and LEACH protocols.

The energy consumption is measured by the average energy expended by each sensor in each of the four network scenarios. It can be observed in the results' diagrams presented in *Figure 5.11* that the proposed cluster-based ABC routing protocol performs better than the compared one in terms of energy consumption. In each network scenario, the average energy expended by each sensor as well as their standard deviations in each case is computed and highlighted in *Table 5.6*. It can clearly be seen that, sensors in the ABD-SD protocol consume less energy than the sensors in the compared protocols. However, we must recognize that in terms of amount of variation of the expended energy by each sensor,



Figure 5.10: Energy consumed by each node in each value of  $\alpha$ .

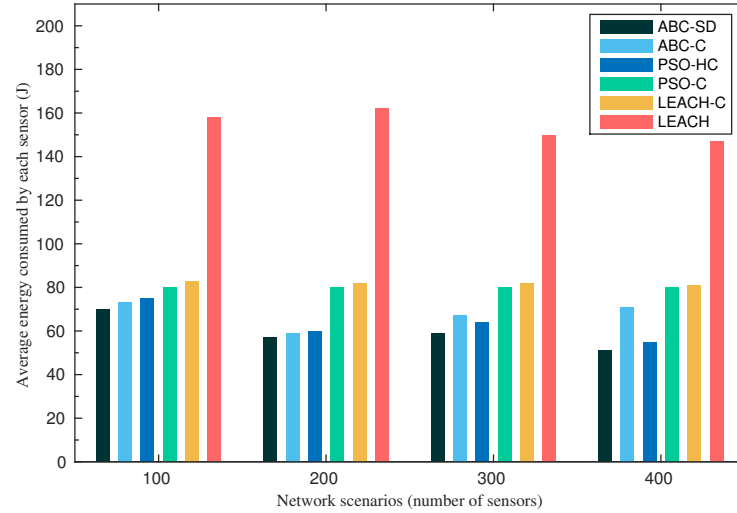
i.e., the standard deviation, the results obtained in PSO-C and LEACH-C protocols have the minimum dispersion.

Table 5.6: Average sensor's energy consumption ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ).

Protocol	Network scenario							
	#100		#200		#300		#400	
	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$
<b>ABC-SD</b>	<b>70.21</b>	<b>1.10</b>	<b>57.03</b>	<b>1.83</b>	<b>59.05</b>	<b>1.06</b>	<b>51.02</b>	<b>0.93</b>
<b>ABC-C</b>	73.01	1.63	59.13	1.93	67.01	0.95	71.12	1.01
<b>PSO-HC</b>	75.10	1.97	60.41	1.94	64.63	1.02	54.06	1.04
<b>PSO-C</b>	80.03	0.27	80.02	0.51	80.01	0.90	80.04	0.52
<b>LEACH-C</b>	83.13	0.81	82.61	1.02	82.01	0.23	81.31	0.41
<b>LEACH</b>	158.01	3.09	162.14	2.36	150.10	0.93	147.03	1.68

Although it is needed that the quantity of the expended energy by each sensor be minimal, the amount of data delivered at the *BS*, in other terms, the reliability of the network, is very important specially in the case of large scale WSNs. So, in *Figure 5.12*, we present the obtained energy efficiency that is equivalent to the amount of data received by the *BS* with 1 joule of power, during the simulation time in each network's scenario. Also,

Figure 5.11: Energy consumption

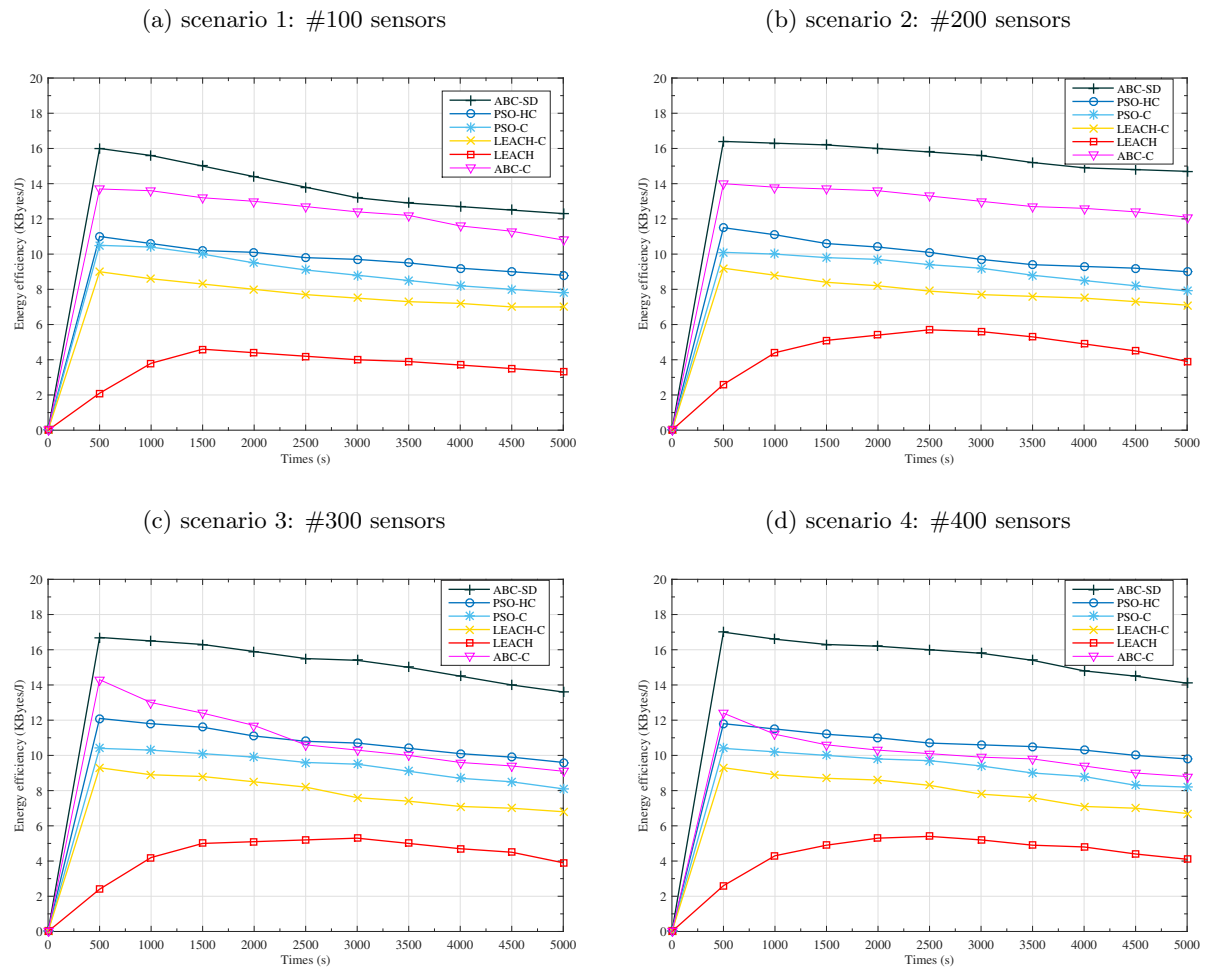


for evaluating the network lifetime, diagrams in *Figure 5.13* show the time at which the first sensor dies in each network's scenario. It can be seen that the first dead sensor in ABC-SD occurs long after the first dead in LEACH-C and LEACH. Moreover, according to the curves visible in *Figure 5.12a-d*, we conclude that the energy efficiency in the ABC-SD protocol is far better than the others since the proposed protocol offers a high data delivery to the *BS* while consuming less power during all the simulation time.

However, it's important to note that the ABC-C protocol enables good results in terms of first dead. However, concerning the results obtained in the evaluation of the energy efficiency in the ABC-C, the curves in *Figure 5.12* show good results for network scenarios with and 100 and 200 sensors, but in network scenarios with 300 and 400 sensors, the ABC-C enabled results close to those obtained in PSO-HC and PSO-C. This means that, the ABC-C protocol presents some shortcomings in terms of management of energy when the number of sensors grows.

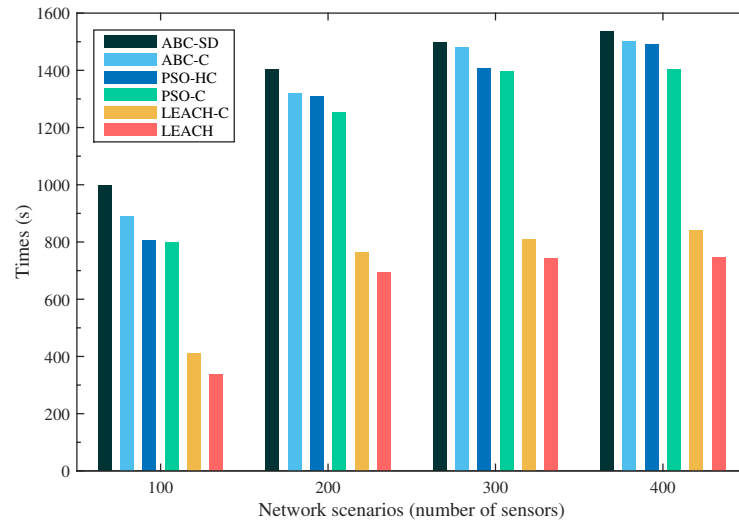
On the other hand, it is important to note that, results obtained in PSO-HC, PSO-C and LEACH-C protocols aren't much far away from each other. Unfortunately, the LEACH protocol presents poor results in all network scenarios. This is due to the adopted long radio transmission range and the single hop routing strategy in LEACH's design. In addition, in LEACH, there is a random and probabilistic based mechanism for the selection of *CHs* that doesn't consider the energy level of the sensor. Also, there is not a sleep/wakeup mechanism for energy conservation during the idle time. The adopted strategies in their design led to weakness results, in particular, it caused a considerable energy wastage and a low data delivery rate.

Figure 5.12: Energy efficiency



Furthermore, the adopted fitness function in the building of clusters favored this visible gain in performance obtained in the ABC-SD protocol. Indeed, the fitness function (*see Section 5.3.2.5*) is designed in such a way that it takes into account the energy as well as the QoS criteria. First, the energy criterion allows the choice of *CHs* with sufficient energy since they act as relay sensors that aggregate and forward gathered data to the *BS*. Moreover, since *CHs* and of course clusters change at each round, the power used by *CHs* is balanced among other sensors. The second criterion as to it, concerns the assignment of the cluster members to a given cluster according to the quality of the communication link. The position update mechanism existing in the ABC algorithm used here in the proposed clustering algorithm efficiently helps us in providing optimal assignment of cluster members. Moreover, the adopted mechanism in the route discovery process allows the choice of the path with fewer number of hops, and which is less power expensive (*see the proposed model in Equation 5.18*), in which gathered data will be forwarded to *CHs*. These

Figure 5.13: Time at which the first sensor dies.



mechanisms help the ABC-SD for the well balancing of the overall power consumption of sensors. The results obtained here by the proposed ABC-SD protocol, specially for network lifetime, confirm the results obtained by authors in [23]. This is probably favored by the use of the ABC optimization algorithm in our clustering algorithm.

Even if in the design of sensor networks, it is necessary to save sensor's energy as long as possible, keeping a good throughput is also essential since the main objective in WSNs is to collect the gathered data. Besides the energy consumption metrics, the overall network performance is also measured by the amount of packets delivered to the *BS* as well as the packets loss rate. Unlike the much adopted evaluation of the amount of data packets received by the *BS*, which considers the amount of aggregated data packets sent by *CHs* to the *BS*, we computed that quantity by considering all the data received by each *CH* before the aggregation. This allowed us to know the exact amount of data generated by the overall network.

Diagrams in Figure 5.14 present the amount of packets successfully received by the *BS* from *CHs* in each network's scenario. Table 5.7 shows the means and the standard deviations of the amount of packets received by the *BS* of the 10 simulation runs. By observing the results obtained for ABC-SD and those of the compared protocols, it can clearly be seen that the ABC-SD protocol produces a greater amount of packets received by the *BS*. In addition, given the standard deviation values, the ABC-SD presents minimum fluctuations in the mean while in PSO-C, LEACH-C and LEACH protocols, there is a lot of fluctuations around the mean. Moreover, as it is seen in the diagrams, the compared PSO based protocols, i.e., PSO-HC and PSO-C, also produce a good performance in this evaluation metric. However, results of this test show that the ABC-C protocol presents

some difficulties to handle a growing amount of sensors since the performance of the algorithm starts to decrease from 300 sensors.

Figure 5.14: Amount of data packets received by the *BS*.

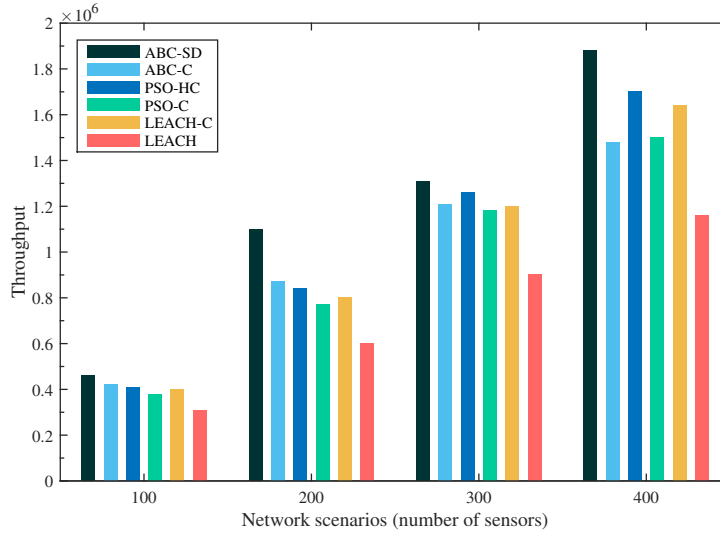


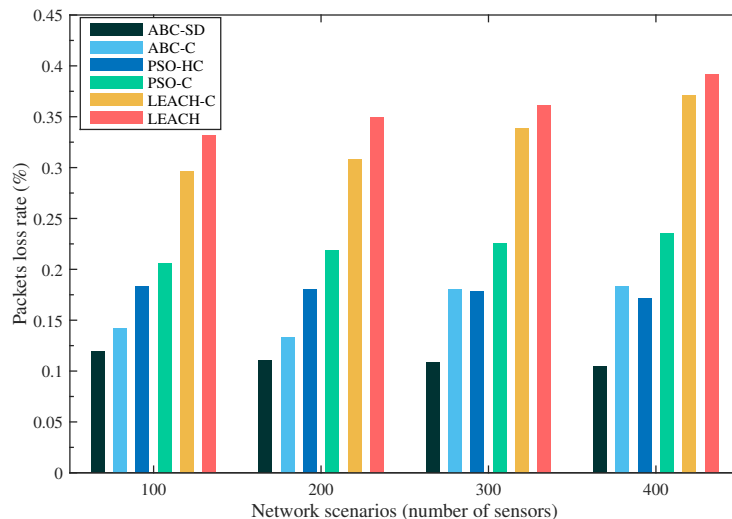
Table 5.7: Mean of throughput ( $\bar{\mu}(\times 10^6)$ ) and standard deviation ( $\sigma(\times 10^6)$ ).

Protocol	Network scenario							
	#100		#200		#300		#400	
	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$
<b>ABC-SD</b>	<b>0.46</b>	<i>0.001</i>	<b>1.10</b>	<i>0.003</i>	<b>1.31</b>	<i>0.003</i>	<b>1.88</b>	<i>0.005</i>
<b>ABC-C</b>	0.42	<i>0.003</i>	0.87	<i>0.005</i>	1.21	<i>0.006</i>	1.48	<i>0.009</i>
<b>PSO-HC</b>	0.41	<i>0.002</i>	0.84	<i>0.002</i>	1.26	<i>0.003</i>	1.70	<i>0.007</i>
<b>PSO-C</b>	0.38	<i>0.006</i>	0.77	<i>0.009</i>	1.18	<i>0.009</i>	1.50	<i>0.011</i>
<b>LEACH-C</b>	0.40	<i>0.009</i>	0.80	<i>0.017</i>	1.20	<i>0.023</i>	1.64	<i>0.096</i>
<b>LEACH</b>	0.31	<i>0.011</i>	0.60	<i>0.053</i>	0.90	<i>0.093</i>	1.16	<i>0.099</i>

Then, always for the evaluation of the throughput, we execute considered protocols in order to evaluate the packet loss rate. The means and their corresponding standard deviations of the packet loss rate is given in Table 5.8. It is clear from Figure 5.15 that, in all considered network scenarios, the packet loss rate for the ABC-SD protocol is far lesser than in LEACH-C and LEACH protocols. These protocols produce a high rate of data loss. This is owing to the adopted single hop communications.

Besides, since the proposal includes a centralized clustering algorithm installed at the *BS*, we evaluated the impact of the communication overhead in the whole execution of the

Figure 5.15: Packets loss.

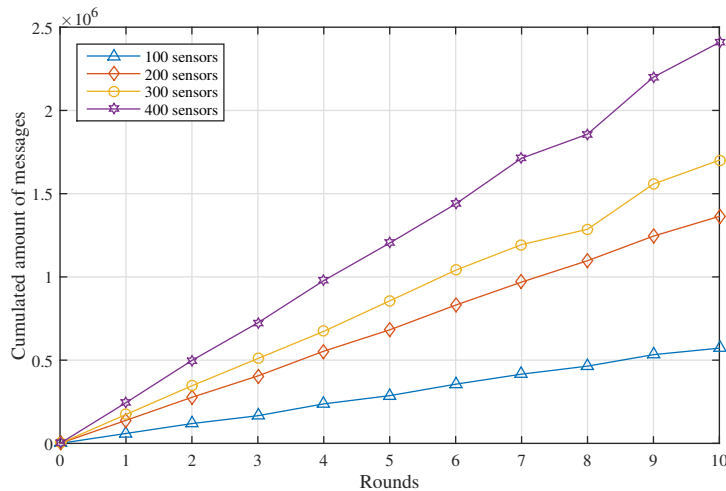
Table 5.8: Mean of packet loss rate ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ).

Protocol	Network scenario							
	#100		#200		#300		#400	
	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$
<b>ABC-SD</b>	<b>0.119</b>	<i>0.001</i>	<b>0.111</b>	<i>0.002</i>	<b>0.109</b>	<i>0.003</i>	<b>0.105</b>	<i>0.004</i>
<b>ABC-C</b>	0.142	<i>0.001</i>	0.133	<i>0.003</i>	0.180	<i>0.005</i>	0.183	<i>0.006</i>
<b>PSO-HC</b>	0.183	<i>0.001</i>	0.180	<i>0.002</i>	0.178	<i>0.005</i>	0.172	<i>0.006</i>
<b>PSO-C</b>	0.206	<i>0.001</i>	0.219	<i>0.003</i>	0.226	<i>0.004</i>	0.235	<i>0.005</i>
<b>LEACH-C</b>	0.296	<i>0.002</i>	0.308	<i>0.003</i>	0.339	<i>0.005</i>	0.371	<i>0.011</i>
<b>LEACH</b>	0.332	<i>0.005</i>	0.349	<i>0.008</i>	0.361	<i>0.008</i>	0.392	<i>0.019</i>

ABC-SD protocol during the simulation. To achieve that, we computed the cumulated amount of messages in each round, from the bootstrapping and setup step till the establishment of route paths and data transfer steps. Figure 5.16 shows the evolution of the communication overhead during each round in the four network scenarios. The obtained results show that the cumulated amount of generated messages increases proportionally to the number of sensors in each round. In addition, the adopted controlled flooding mechanism in the bootstrapping step contributed to contain the blast of the generated messages. This implies that, there is no significant communication overhead in the ABC-SD.

Now, in order to evaluate the performance of the proposal in terms of network coverage, we evaluated the number of unclustered sensors, i.e., the amount of inactive sensors, per round in each network scenario. To achieve this, we suppose that a sensor is considered inactive

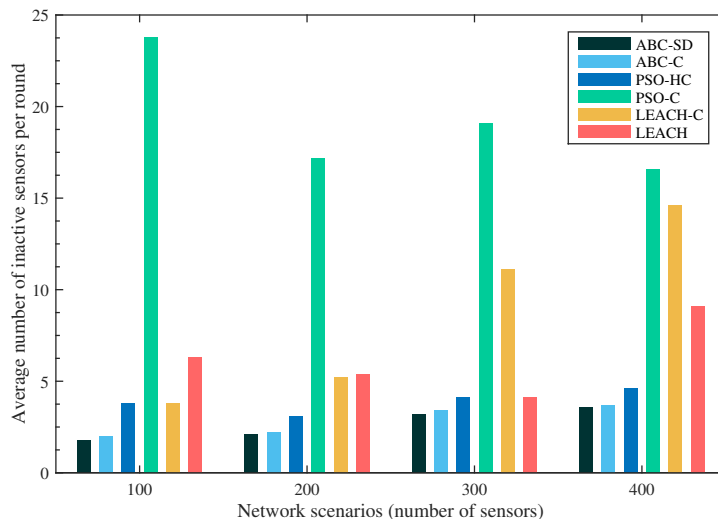
Figure 5.16: The communication overhead in ABC-SD.



if its energy level is not equal to zero and if it is not a follower of a given *CH*. Diagrams in Figure 5.17 show the comparison of our proposal with the compared protocols in terms of the average number of unclustered sensors per round. It can be observed that the amount of unclustered sensors in ABC-SD is significantly lesser than in its comparatives. This is allowed by the adopted fitness function in the clustering algorithm which takes care of energy consumption of sensors and the QoS of communication link within clusters. Moreover, it can be seen that the ABC-C and PSO-HC protocols yield good results. This is respectively allowed by the positions update of the ABC algorithm in the ABC-C and the adopted particle encoding scheme in the PSO-HC. However, the results of the PSO-C protocol for this test are very poor. This is due to the used cost-based function for the intra-cluster communication, which does not allow far sensors to be members of a cluster. Regarding the LEACH and LEACH-C protocols since their design's aim at balancing the load of CHs, sensors are assigned to CHs that are far away from them. This fact leads to a lot of death of CHs and this considerably increases the amount of unclustered sensors.

In the above presented results, we have used the  $100 \times 100 \text{ m}^2$  sensing area varying the number of sensors in the performance evaluation. However, in order to confirm the effectiveness of the ABC-SD, 10 simulation runs of the four network scenarios in an area of  $500 \times 500 \text{ m}^2$  with the *BS* positioned at the coordinate (0;0) of the field have been used in order to evaluate the potential impact of changing the sensing area on the average energy consumed by each sensor and the throughput. Figure 5.18 and Figure 5.19 respectively show the results of the average energy consumed by each sensor and the results obtained in the evaluation of the throughput. In addition, in each network scenario, the average energy expended by each sensor and the standard deviations is given in Table 5.9 and Table 5.10.

Figure 5.17: Amount of unclustered sensors.

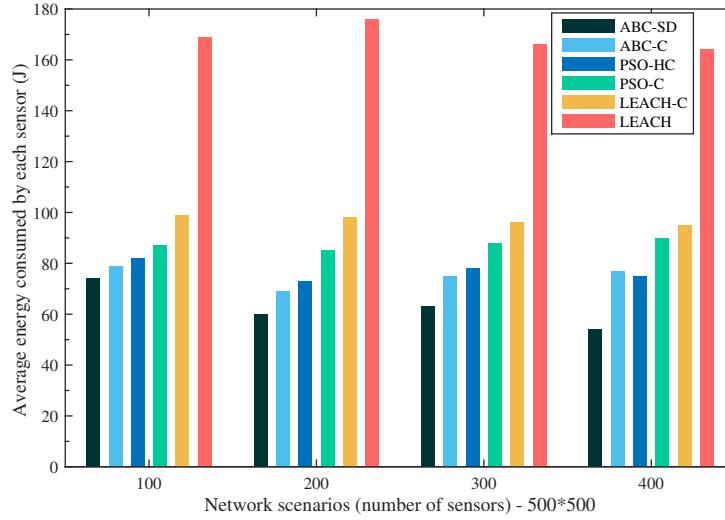
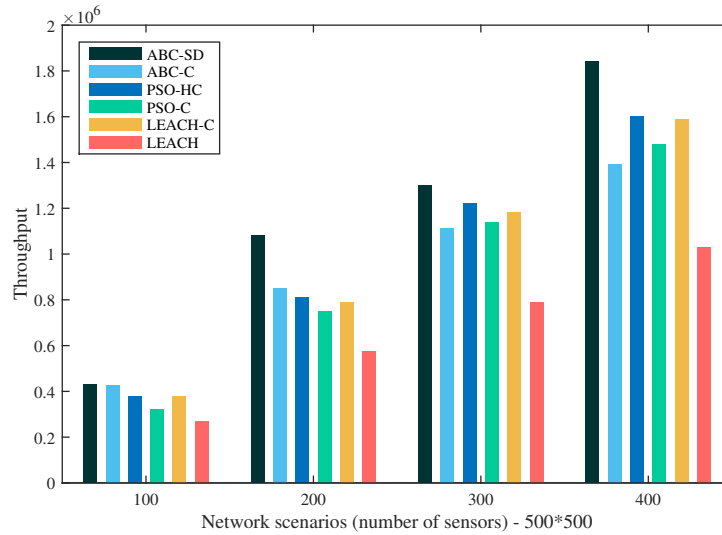


It can be observed from Figure 5.18 that the ABC-SD enables good results for the energy expenditure in each sensor than the compared protocol even if the sensing area becomes larger. Nevertheless, the standard deviations presented in Table 5.9 show that the PSO-C protocol has a minimum dispersion. In addition, it's important to note that the ABC-C and PSO-HC also outputted good results when changing the sensing area. Figure 5.19 reveals the effectiveness of the ABC-SD in terms of amount of data packets received by the BS with a minimum dispersion, in a sensing area of  $500 \times 500 m^2$ . Although the ABC-SD enables a better throughput than the compared, it can be found from Table 5.10 that the ABC-SD enables higher dispersion in the results.

Table 5.9: Average sensor's energy consumption ( $\bar{\mu}$ ) and standard deviation ( $\sigma$ ) - sensing area of  $500 \times 500 m^2$ .

Protocol	Network scenario							
	#100		#200		#300		#400	
	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$
<b>ABC-SD</b>	<b>74.81</b>	<b>1.03</b>	<b>59.98</b>	<b>1.63</b>	<b>61.82</b>	<b>1.01</b>	<b>52.72</b>	<b>1.03</b>
<b>ABC-C</b>	79.11	1.01	64.23	1.03	71.01	0.95	76.10	0.98
<b>PSO-HC</b>	81.41	2.12	71.44	2.14	76.43	1.69	74.96	1.53
<b>PSO-C</b>	87.83	0.29	82.11	0.41	85.12	0.63	86.40	0.62
<b>LEACH-C</b>	99.03	1.01	98.93	1.04	97.03	0.98	96.33	0.30
<b>LEACH</b>	169.01	3.54	178.41	2.97	165.32	1.97	164.09	2.19



Figure 5.18: Energy consumption in a sensing area of  $500 \times 500 m^2$ .Figure 5.19: Amount of data packets received by the *BS* in a sensing area of  $500 \times 500 m^2$ .

Finally, by returning to the design goals of the ABC-SD protocol (*see Section 5.3.1*), the results obtained in the evaluation of the network lifetime as well as the network coverage and the throughput, confirm that the ABC-SD protocol aims at maintaining a good link quality, as well as providing a great number of delivered packets while preserving the sensor's power consumption as best as possible.

Table 5.10: Mean of throughput ( $\bar{\mu}(\times 10^6)$ ) and standard deviation ( $\sigma(\times 10^6)$ ) - sensing area of  $500 \times 500 m^2$ .

Protocol	Network scenario							
	#100		#200		#300		#400	
	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$	$\bar{\mu}$	$\sigma$
<b>ABC-SD</b>	<b>0.42</b>	<i>0.024</i>	<b>1.09</b>	<i>0.014</i>	<b>1.29</b>	<i>0.099</i>	<b>1.83</b>	<i>0.101</i>
<b>ABC-C</b>	0.41	<i>0.004</i>	0.83	<i>0.005</i>	1.10	<i>0.008</i>	1.37	<i>0.009</i>
<b>PSO-HC</b>	0.39	<i>0.005</i>	0.80	<i>0.008</i>	1.21	<i>0.004</i>	1.60	<i>0.006</i>
<b>PSO-C</b>	0.31	<i>0.008</i>	0.74	<i>0.010</i>	1.14	<i>0.010</i>	1.46	<i>0.019</i>
<b>LEACH-C</b>	0.38	<i>0.012</i>	0.79	<i>0.014</i>	1.17	<i>0.029</i>	1.59	<i>0.101</i>
<b>LEACH</b>	0.27	<i>0.016</i>	0.59	<i>0.061</i>	0.79	<i>0.125</i>	1.01	<i>0.163</i>

## 5.5 Conclusion

In this chapter, we investigated problems of clustering and routing in WSNs and then, we proposed a centralized cluster-based routing protocol called ABC-SD. The clustering problem is formulated as a LP problem and the routing problem is solved with a cost-based function. Clusters are built in an energy efficient way by exploiting the efficient features and the fast convergence characteristics of the ABC metaheuristic. As for the routing problem, it is brought back to a problem of choosing a low-cost path among existing paths from a source to a destination, for the intra-cluster communication. The choice of CHs in the clustering algorithm also helps us in the construction of routes for the inter-cluster communication. The results of simulation show that, the proposed protocol delivers better performance than its comparatives in terms of the network lifetime, the network coverage and the amount of packets received by the BS.



## Chapter 6

# Conclusion and Future Research Directions

### 6.1 Conclusion

During the last few years, advances in miniaturization and low power design of electronic devices have allowed the development of small and smart sensors that are capable of detecting ambient conditions from a given environment such as temperature, movement, vibration and acoustic phenomenon.

Research in this field is very exciting because of a variety of offered services by sensor networks. Usually, sensors are organized into a massively deployed WSN, which cooperatively gather and transmit information to an end user or a BS. The technology of WSNs has enabled a number of opportunities for diverse applications. The sensing platforms and the existing applications are fastly growing. In the case of large scale networks, the management of sensors becomes a challenge. Energy efficiency, throughput and scalability are key requirements in WSNs.

To address these goals, sensors are often organized into disjoint and mostly non-overlapping groups named clusters. Clustering and routing that are well known optimization problems have been widely studied for extending the network lifetime, which is a critical issue in sensor networks.

Owing to the complexity introduced by large scale sensor networks and the limited resources enabled by individual sensors, the wonderful behaviors of biological systems and its different solutions for optimization problems are progressively applied in order to optimize a whole management of WSNs. In this thesis, we contributed in clustering and routing protocols for improving the management of WSNs by using biologically inspired solutions.

We started this thesis by presenting an overview of the domains of WSNs and applications. This is followed by a survey of different well known clustering algorithms for WSNs by highlighting their design, features, complexity and insufficiency.

Then, we first addressed the issue of distributed clustering in WSNs by paying attention to the CH election process in order to reduce the energy wastage. The proposed approach called NEST for CH election and clustering is inspired by the nest-sites selection process of honeybee swarms. CHs are elected in a distributed and a self-organized manner by sensors themselves and each proceeding CH election is done inside the cluster, if the CH election is necessary. Thus, the energy consumption of the network is significantly balanced within all cluster members. This avoids the rapid death of CHs. So, the network lifetime is well improved. For the validation of the NEST, extensive simulations were made with a great number of sensors, in order to be closer to the reality of large scale sensor networks. The results of experiments showed that, the NEST is more performant than the most existing clustering solutions in terms of the network lifetime, energy consumption and throughput.

In large scale networks, it is demonstrated that, the optimal clustering mechanism has a strong influence on the overall performance of the network. For this reason, in the second contribution, we investigated problems of centralized clustering and routing in WSNs and then, we proposed a cluster-based routing protocol called ABC-SD. In the proposal, the clustering problem is formulated as a Linear Programming problem and the routing problem is addressed with a Cost-based Function. We opted for a centralized clustering optimization carried out in the BS, which has a rich resource suitable for it. Clusters are built in a low costly way by exploiting the efficient features and the fast convergence characteristics of the Artificial Bee Colony metaheuristic. The routing problem is brought back to a problem of choosing a low-cost path among existing paths from a source to a destination, for the intra-cluster communication. The choice of CHs in the clustering algorithm also helps us in the construction of routes for the inter-cluster communication. Contrary to the existing protocols, the major contributions of our proposal are illustrated as follows. Firstly, in order to be closer to the real sensing environment and existing sensor's hardware, a realistic energy consumption model is used. Secondly, it defines a new cluster balancing mechanism that makes a trade-off between sensor's energy level and the quality of the communication link. Thirdly, we designed a pre-established routes approach in which the chosen routing path is less-costly in terms of power consumption and employs a least number of hops. Fourthly, unlike the most existing methods, which focus on one metric, the ABC-SD protocol performs well in three evaluation metrics. Intensive simulations were made with a number of network topologies in different network sizes and the obtained results show that, the proposed protocol delivers better performance than its comparatives in terms of the network lifetime, the network coverage and the throughput.

## 6.2 Future Research Directions

In our contributions, we obtained encouraging results but it could lead to several improvements at various levels.

1. Always for reducing the energy consumption in the NEST, the envisaged amelioration concerns an optimal integration of a collective decision-making in a honeybee swarm, in order to preserve coverage during the CH election. Also, we are planning to study the effects of some network and physical layer issues in the NEST algorithm, such as channel contention, noise, retransmissions, dropped packets and errors.
2. For mobility purpose, we plan to study the effects of mobile sensors as well as mobile sink in the proposed ABC-SD algorithm. Furthermore, we would also focus on the design of a TDMA frame in the case of variable packet lengths.
3. To highlight strengths and weaknesses of proposed protocols, we plan to conduct some experiments on our contributions by using a real physical WSN in various sensing environments.

In the forthcoming future, we plan to design a new scheme of mobile sensing that aims at providing a good coverage and throughput while maintaining better energy efficiency and high network availability. Indeed, in a given WSN, determining the sensor coverage in a designated sensing environment is an important goal that allows the evaluation of the WSN effectiveness. Moreover, the sensor coverage problem depends on the application. For example, target tracking applications may require a higher degree of coverage while environmental or agricultural monitoring applications can digest a less degree of sensor coverage. In target tracking applications, in order to get more reliable gathered data, the adoption of a strategy in which multiple sensors monitor the same location is needed. It has been confirmed that mobility significantly improves the overall network lifetime, reliable data gathering, delay and latency problems [24, 91, 92, 93].

In WSNs, mobile sensing refers to the presence of one or more mobile sinks or mobile sensors, which have the main role of collecting the gathered data by sensor nodes. Mobile WSNs enable more dynamic topology since mobile sinks move continuously over the network in a more or less random fashion. However, the problem of designing a strategy that allows mobile sinks to move in a distributed self-organized manner is not evident to solve by a deterministic polynomial algorithm. Bacterial Foraging Optimization Algorithm (BFOA) [94] that is inspired by some features of the social foraging behaviors of the *Escherichia coli* bacteria, can be applied to model a trial solution for the autonomous movement of mobile sinks, which is an optimization problem. As future research direction, we plan to design models that allow mobile sinks to move over the network in a self-organized and self-adaptive way. We provided a start of solution in [95].



# Bibliography

- [1] Adamu Murtala Zungeru, Li-Minn Ang, and Kah Phooi Seng. Termite-hill: Performance optimized swarm intelligence based routing algorithm for wireless sensor networks. *Journal of Network and Computer Applications*, 35(6):1901–1917, 2012.
- [2] Ado Adamou Abba Ari, Abdelhak Gueroui, Nabila Labraoui, and Blaise Omer Yenke. Concepts and evolution of research in the field of wireless sensor networks. *International Journal of Computer Networks & Communications*, 7(1):81–98, 2015.
- [3] Chafiq Titouna, Makhoulf Aliouat, and Mourad Gueroui. FDS: fault detection scheme for wireless sensor networks. *Wireless Personal Communications*, 86(2):549–562, 2016.
- [4] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12):2292–2330, 2008.
- [5] Jian Li, Haiying Zhou, De-Cheng Zuo, Kun Mean Hou, Hepeng Xie, and Peng Zhou. Energy consumption evaluation for wireless sensor network nodes based on queuing petri net. *International Journal of Distributed Sensor Networks*, 2014, 2014.
- [6] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks (TOSN)*, 6(2):16, 2010.
- [7] Agustin Barberis, Leonardo Barboni, and Maurizio Valle. Evaluating energy consumption in wireless sensor networks applications. In *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*, pages 455–462. IEEE, 2007.
- [8] Joseph Kabara and Maria Calle. MAC protocols used by wireless sensor networks and a general method of performance evaluation. *International Journal of Distributed Sensor Networks*, 2012, 2012.



- [9] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.
- [10] Faranak Heidarian, Julien Schmaltz, and Frits Vaandrager. Analysis of a clock synchronization protocol for wireless sensor networks. *Theoretical Computer Science*, 413(1):87–105, 2012.
- [11] Nabila Labraoui, Mourad Gueroui, and Larbi Sekhri. On-off attacks mitigation against trust systems in wireless sensor networks. In *Computer Science and Its Applications*, pages 406–415. Springer, 2015.
- [12] Nabila Labraoui, Mourad Gueroui, Makhlouf Aliouat, and Jonathan Petit. Reactive and adaptive monitoring to secure aggregation in wireless sensor networks. *Telecommunication systems*, 54(1):3–17, 2013.
- [13] Ado Adamou Abba Ari, Blaise Omer Yenke, Nabila Labraoui, and Abdelhak Gueroui. Energy efficient clustering algorithm for wireless sensor networks using the abc meta-heuristic. In *Computer Communication and Informatics (ICCCI), 2016 International Conference on*, pages 1–6, Coimbatore, Tamilnadu, India, January 2016. IEEE.
- [14] Mohammad Hammoudeh and Robert Newman. Adaptive routing in wireless sensor networks: Qos optimisation for enhanced application performance. *Information Fusion*, 22:3–15, 2015.
- [15] Riham Elhabyan and Mustapha Yagoub. Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network. *Journal of Network and Computer Applications*, 52:116–128, 2015.
- [16] Pratyay Kuila and Prasanta K Jana. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Engineering Applications of Artificial Intelligence*, 33:127–140, 2014.
- [17] Falko Dressler. *Self-organization in sensor and actor networks*. John Wiley & Sons, 2008.
- [18] Xuxun Liu. A survey on clustering routing protocols in wireless sensor networks. *Sensors*, 12(8):11113–11153, 2012.
- [19] Adamu Murtala Zungeru, Li-Minn Ang, and Kah Phooi Seng. Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison. *Journal of Network and Computer Applications*, 35(5):1508–1536, 2012.

- [20] Ankit Sharma and Jawahar Thakur. An energy efficient network life time enhancement proposed clustering algorithm for wireless sensor networks. In *International Conference of Parallel and Distributed Computing Systems (PDCS 2014)*, pages 111–115. 2014.
- [21] Xuelian Cai, Yulong Duan, Ying He, Jin Yang, and Changle Li. Bee-Sensor-C: An energy-efficient and scalable multipath routing protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015(976127):0–14.
- [22] Nurul Muazzah Abdul Latiff, Charalampos Tsimenidis, and Bayan Sharif Sharif. Energy-aware clustering for wireless sensor networks using particle swarm optimization. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5. IEEE, 2007.
- [23] Dervis Karaboga, Selcuk Okdem, and Celal Ozturk. Cluster based wireless sensor network routing using artificial bee colony algorithm. *Wireless Networks*, 18(7):847–860, 2012.
- [24] Elie Tagne Fute and Emmanuel Tonye. Modelling and self-organizing in mobile wireless sensor networks: Application to fire detection. *International Journal of Applied Information Systems, IJAIS, New York, USA*, 5(3), 2013.
- [25] Wendi Beth Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10–20. IEEE, 2000.
- [26] Ossama Younis and Sonia Fahmy. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3(4):366–379, 2004.
- [27] Wendi Beth Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, 2002.
- [28] Muhammad Saleem, Gianni A Di Caro, and Muddassar Farooq. Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. *Information Sciences*, 181(20):4597–4624, 2011.
- [29] Riham Elhabyan and Mustapha Yagoub. PSO-HC: Particle swarm optimization protocol for hierarchical clustering in wireless sensor networks. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*, pages 417–424. IEEE, 2014.

- [30] Francois Avril, Thibault Bernard, Alain Bui, and Devan Sohier. Clustering and communications scheduling in WSNs using mixed integer linear programming. *Communications and Networks, Journal of*, 16(4):421–429, 2014.
- [31] Jing Yang, Mai Xu, Wei Zhao, and Baoguo Xu. A multipath routing protocol based on clustering and ant colony optimization for wireless sensor networks. *Sensors*, 10(5):4521–4540, 2010.
- [32] Falko Dressler and Ozgur Baris Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881–900, 2010.
- [33] Dervis Karaboga and Bahriye Akay. A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 31(1-4):61–85, 2009.
- [34] Sohail Jabbar, Rabia Iram, Abid Ali Minhas, Imran Shafi, Shehzad Khalid, and Muqet Ahmad. Intelligent optimization of wireless sensor networks through bio-inspired computing: survey and future directions. *International Journal of Distributed Sensor Networks*, 2013, 2013.
- [35] Saad Ahmed Munir, Biao Ren, Weiwei Jiao, Bin Wang, Dongliang Xie, and Jian Ma. Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops-Volume 02*, pages 113–120. IEEE, 2007.
- [36] Ruairí de Fréin, Radhika Loomba, and Brendan Jennings. Selecting energy efficient cluster-head trajectories for collaborative mobile sensing. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–7. IEEE, 2015.
- [37] Fayçal Ait Aoudia, Matthieu Gautier, and Olivier Berder. GRAPMAN: gradual power manager for consistent throughput of energy harvesting wireless sensor nodes. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*, pages 954–959. IEEE, 2015.
- [38] Th Arampatzis, John Lygeros, and Stamatias Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 719–724. IEEE, 2005.
- [39] Shivangi Katiyar and Devendra Prasad. A comprehensive survey of data processing approaches. *International Journal of Computer Applications*, 126(11), 2015.
- [40] Xi Xu, Rashid Ansari, Ashfaq Khokhar, and Athanasios Vasilakos. Hierarchical data aggregation using compressive sensing (HDACS) in WSNs. *ACM Transactions on Sensor Networks (TOSN)*, 11(3):45, 2015.

- [41] Mukesh Kumar and Kamlesh Dutta. A survey of security concerns in various data aggregation techniques in wireless sensor networks. In *Intelligent Computing, Communication and Devices*, pages 1–15. Springer, 2015.
- [42] Marco Conti and Stefano Giordano. Mobile ad hoc networking: milestones, challenges, and new research directions. *Communications Magazine, IEEE*, 52(1):85–96, 2014.
- [43] Jiguo Yu, Ying Chen, Liran Ma, Baogui Huang, and Xiuzhen Cheng. On connected target k-coverage in heterogeneous wireless sensor networks. *Sensors*, 16(1):104, 2016.
- [44] Asma Mesmoudi, Mohammed Feham, Nabila Labraoui, and Chakib Bekara. An efficient area-based localization algorithm for wireless sensor networks. *International Review on Computers and Software (IRECOS)*, 10(10):1062–1070, 2015.
- [45] Nabila Labraoui, Mourad Gueroui, and Makhoulf Aliouat. Secure DV-Hop localization scheme against wormhole attacks in wireless sensor networks. *Transactions on Emerging Telecommunications Technologies*, 23(4):303–316, 2012.
- [46] Achim Berger, Markus Pichler, Johannes Klinglmayr, Albert Potsch, and Andreas Springer. Low-complex synchronization algorithms for embedded wireless sensor networks. *Instrumentation and Measurement, IEEE Transactions on*, 64(4):1032–1042, 2015.
- [47] Davut Incebacak, Ruken Zilan, Bulent Tavli, Jose Maria Barcelo-Ordinas, and Jorge Garcia-Vidal. Optimal data compression for lifetime maximization in wireless sensor networks operating in stealth mode. *Ad Hoc Networks*, 24:134–147, 2015.
- [48] Nabila Labraoui, Mourad Gueroui, Makhoulf Aliouat, and Tanveer Zia. Data aggregation security challenge in wireless sensor networks: a survey. *Ad hoc & Sensor Networks, International Journal*, 12(3-4):295–324, 2011.
- [49] Nabila Labraoui, Mourad Gueroui, Makhoulf Aliouat, and Jonathan Petit. RAHIM: Robust adaptive approach based on hierarchical monitoring providing trust aggregation for wireless sensor networks. *Journal of Universal Computer Science*, 17(11):1550–1571, 2011.
- [50] Farah Khedim, Nabila Labraoui, and Mohamed Lehsaini. Dishonest recommendation attacks in wireless sensor networks: A survey. In *Programming and Systems (ISPS), 2015 12th International Symposium on*, pages 1–10. IEEE, 2015.
- [51] Nabila Labraoui. A reliable trust management scheme in wireless sensor networks. In *Programming and Systems (ISPS), 2015 12th International Symposium on*, pages 1–6. IEEE, 2015.

- [52] Gaurav Gupta and Mohamed Younis. Fault-tolerant clustering of wireless sensor networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1579–1584. IEEE, 2003.
- [53] Chafiq Titouna, Makhlouf Aliouat, and Mourad Gueroui. Outlier detection approach using bayes classifiers in wireless sensor networks. *Wireless Personal Communications*, 85(3):1009–1023, 2015.
- [54] Shazana Md Zin, Nor Badrul Anuar, Miss Laiha Mat Kiah, and Al-Sakib Khan Pathan. Routing protocol design for secure WSN: Review and open research issues. *Journal of Network and Computer Applications*, 41:517–530, 2014.
- [55] Nurul Muazzah Abdul Latiff, Charalampos Tsimenidis, and Bayan Sharif Sharif. Performance comparison of optimization algorithms for clustering in wireless sensor networks. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on*, pages 1–4. IEEE, 2007.
- [56] Buddha Singh and Daya Krishan Lobiyal. A novel energy-aware cluster head selection based on particle swarm optimization for wireless sensor networks. *Human-Centric Computing and Information Sciences*, 2(1):1–18, 2012.
- [57] Haidar Safa, Mathieu Moussa, and Hassan Artail. An energy efficient genetic algorithm based approach for sensor-to-sink binding in multi-sink wireless sensor networks. *Wireless networks*, 20(2):177–196, 2014.
- [58] Shahaboddin Shamshirband, Nor Badrul Anuar, Miss Laiha Mat Kiah, Vala Ali Rohani, Dalibor Petković, Sanjay Misra, and Abdul Nasir Khan. Co-FAIS: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks. *Journal of Network and Computer Applications*, 42:102–117, 2014.
- [59] Zhanshan Sam Ma and Axel Krings. Insect sensory systems inspired computing and communications. *Ad Hoc Networks*, 7(4):742–755, 2009.
- [60] Yanfei Liu and Kevin Passino. Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. *Journal of Optimization Theory and Applications*, 115(3):603–628, 2002.
- [61] Marco Dorigo, Mauro Birattari, and Thomas Stützle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.
- [62] James Kennedy. Particle swarm optimization. In *Encyclopedia of Machine Learning*, pages 760–766. Springer, 2010.

- [63] Arouna Ndam Njoya, Wahabou Abdou, Albert Dipanda, and Emmanuel Tonye. Evolutionary-based wireless sensor deployment for target coverage. In *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 739–745. IEEE, 2015.
- [64] Dervis Karaboga. Artificial bee colony algorithm. *scholarpedia*, 5(3):6915, 2010.
- [65] Ado Adamou Abba Ari, Blaise Omer Yenke, Nabila Labraoui, Irépran Damakoa, and Abdelhak Gueroui. A power efficient cluster-based routing algorithm for wireless sensor networks: Honeybees Swarm Intelligence based approach. *Journal of Network and Computer Applications*, 69:77–97, 2016.
- [66] Jean Louis Fendji Kedieng Ebongue, Mafai Nelson, and Jean Michel Nlong. Empirical path loss models for 802.11 n wireless networks at 2.4 Ghz in rural regions. In *e-Infrastructure and e-Services for Developing Countries*, pages 53–63. Springer, 2014.
- [67] K Sathish Kumar and T Jayabarathi. Power system reconfiguration and loss minimization for an distribution systems using bacterial foraging optimization algorithm. *International Journal of Electrical Power & Energy Systems*, 36(1):13–17, 2012.
- [68] N Rajasekar, Neeraja Krishna Kumar, and Rini Venugopalan. Bacterial foraging algorithm based solar PV parameter estimation. *Solar Energy*, 97:255–265, 2013.
- [69] Wilfredo Alfonso, José J Velásquez, Kevin Passino, and Eduardo Francisco Caicedo. A honeybee social foraging algorithm for feedback control of smart lights. *Engineering Applications of Artificial Intelligence*, 48:13–31, 2016.
- [70] Bai Li, Ligang Gong, and Yuan Yao. On the performance of internal feedback artificial bee colony algorithm (IF-ABC) for protein secondary structure prediction. In *Advanced Computational Intelligence (ICACI), 2013 Sixth International Conference on*, pages 33–38. IEEE, 2013.
- [71] Mark L Winston. *The biology of the honey bee*. Harvard University Press, 1991.
- [72] Fernand-Nestor Tchuenguem Fohouo, Sidonie Fameni Tope, and Dorothea Brückner. Foraging and pollination behaviour of *Xylocopa olivacea* (Hymenoptera: Apidae) on *Phaseolus coccineus* (Fabaceae) flowers at Ngaoundere (Cameroon). *International Journal of Tropical Insect Science*, 34(02):127–137, 2014.
- [73] Simon Garnier, Jacques Gautrais, and Guy Theraulaz. The biological principles of swarm intelligence. *Swarm Intelligence*, 1(1):3–31, 2007.
- [74] Kevin Passino and Thomas Dyer Seeley. Modeling and analysis of nest-site selection by honeybee swarms: the speed and accuracy trade-off. *Behavioral Ecology and Sociobiology*, 59(3):427–442, 2006.

- [75] Kirk Visscher. Group decision making in nest-site selection among social insects. *Annual Review of Entomology*, 52:255–275, 2007.
- [76] Scott Camazine, Kirk Visscher, Jennifer Finley, and Richard Vetter. House-hunting by honey bee swarms: collective decisions and individual behaviors. *Insectes Sociaux*, 46(4):348–360, 1999.
- [77] Omar Moussaoui, Adlen Ksentini, Mohamed Naimi, and Mourad Gueroui. A novel clustering algorithm for efficient energy saving in wireless sensor networks. In *Computer Networks, 2006 International Symposium on*, pages 66–72. IEEE, 2006.
- [78] Satyasai Jagannath Nanda and Ganapati Panda. A survey on nature inspired meta-heuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 16:1–18, 2014.
- [79] Seema Bandyopadhyay and Edward John Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1713–1723. IEEE, 2003.
- [80] Amit Bhattacharjee, Balagopal Bhallamudi, and Zahid Maqbool. Energy-efficient hierarchical cluster based routing algorithm in WSN: A survey. *International Journal of Engineering Research and Technology*, 2(5 (May-2013)), 2013.
- [81] Nauman Aslam, William Phillips, William Robertson, and Shyamala Sivakumar. A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks. *Information Fusion*, 12(3):202–212, 2011.
- [82] Khelifi Nesrine and Maher Ben Jemaa. HEERP: Hierarchical energy efficient routing protocol for wireless sensor networks. In *Communications and Information Technology (ICCIT), 2012 International Conference on*, pages 308–313. IEEE, 2012.
- [83] Jiguo Yu, Yingying Qi, Guanghui Wang, and Xin Gu. A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution. *AEU-International Journal of Electronics and Communications*, 66(1):54–61, 2012.
- [84] Antoniel da Silva Rego, Joaquim Celestino, Andre Dos Santos, Eduardo Coelho Cerqueira, Anup Patel, and Mehdi Taghavi. BEE-C: a bio-inspired energy efficient cluster-based algorithm for data continuous dissemination in wireless sensor networks. In *Networks (ICON), 2012 18th IEEE International Conference on*, pages 405–410. IEEE, 2012.
- [85] Omid Bozorg Haddad, Abbas Afshar, and Miguel Mariño. Honeybees mating optimization algorithm (HBMO); a new heuristic approach for engineering optimization.

- In *Proceeding of the First International Conference on Modeling, Simulation and Applied Optimization (ICMSA0/05)*, Sharjah, UAE, pages 1–3, 2005.
- [86] Horst Wedde, Muddassar Farooq, Thorsten Pannenbaecker, Bjoern Vogel, Christian Mueller, Johannes Meth, and Rene Jeruschkat. BeeAdHoc: an energy efficient routing algorithm for mobile ad hoc networks inspired by bee behavior. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 153–160. ACM, 2005.
- [87] Shih Tzay-Farn. Particle swarm optimization algorithm for energy-efficient cluster-based sensor networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(7):1950–1958, 2006.
- [88] Texas Instruments. Chipcon as smartrf® CC2420 preliminary datasheet (rev 1.2), 2004-06-09. Technical report, <http://www.ti.com/lit/ds/symlink/cc2420.pdf>, 2014 [accessed on 27.06.2015].
- [89] Norman Lloyd Johnson, Adrienne Winifred Kemp, and Samuel Kotz. *Univariate discrete distributions*, volume 444. John Wiley & Sons, 2005.
- [90] Umberto Malesci and Samuel Madden. A measurement-based analysis of the interaction between network layers in tinyos. In *Wireless Sensor Networks*, pages 292–309. Springer, 2006.
- [91] Md Zakirul Alam Bhuiyan, Guojun Wang, and Athanasios Vasilakos. Local area prediction-based mobile target tracking in wireless sensor networks. *Computers, IEEE Transactions on*, 64(7):1968–1982, 2015.
- [92] Miao Zhao, Yuanyuan Yang, and Cong Wang. Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 14(4):770–785, 2015.
- [93] Marwan Al-Jemeli and Fawnizu Azmadi Hussin. An energy efficient cross-layer network operation model for IEEE 802.15. 4-based mobile wireless sensor networks. *Sensors Journal, IEEE*, 15(2):684–692, 2015.
- [94] Kevin Passino. Bacterial foraging optimization. *Innovations and Developments of Swarm Intelligence Applications*, page 219, 2012.
- [95] Ado Adamou Abba Ari, Abdelhak Gueroui, Nabila Labraoui, Blaise Omer Yenke, Chafik Titouna, and Irépran Damakoa. Adaptive scheme for collaborative mobile sensing in wireless sensor networks: Bacterial foraging optimization approach. In *2016 IEEE 27th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): Mobile and Wireless Networks (IEEE*



*PIMRC2016 Mobile and Wireless*), pages 1–6, Valencia, Spain, September 2016. IEEE.



**Titre :** Intégration des Solutions Bio-inspirées pour une Gestion optimale dans les Réseaux de Capteur sans Fils.

**Mots clés :** RCSF, Routage, Clustering et Bio-inspirées

**Résumé :** Au cours de ces dernières années, les réseaux de capteurs sans fils ont connu un intérêt croissant à la fois au sein de la communauté scientifique et industrielle en raison du large potentiel en terme d'applications offertes. Toutefois, les capteurs sont conçus avec d'extrêmes contraintes en ressources, en particulier la limitation de l'énergie. Il est donc nécessaire de concevoir des protocoles efficaces, évolutifs et moins consommateur d'énergie afin de prolonger la durée de vie de ces réseaux. Le clustering est une approche très populaire, utilisée pour l'optimisation de la consommation d'énergie des capteurs. Cette technique permet d'influencer fortement la performance globale du réseau. En outre, dans de tels réseaux, le routage génère un nombre assez élevé d'opérations non négligeables qui affectent considérablement la durée de vie du réseau ainsi que le débit offert. Dans cette thèse, nous nous sommes intéressés d'une part aux problèmes de clustering et de routage en utilisant des méthodes d'optimisation inspirées de certaines sociétés biologiques fournissant des modèles puissants qui conduisent à l'établissement d'une intelligence globale en se basant sur des comportements individuels très simples. Nous avons proposé une approche de clustering distribuée basée sur le processus de sélection

des sites de nidification chez les colonies d'abeilles. Nous avons formulé le problème de clustering distribuée comme un processus social de prise de décision dans lequel les capteurs agissent d'une manière collective pour choisir des représentants au sein de leurs clusters respectifs. Le protocole proposé assure une distribution de l'équilibrage de charge entre les membres de chaque cluster afin de prolonger la durée de vie du réseau en faisant un compromis entre la consommation d'énergie et la qualité du canal de communication. D'autre part, nous avons proposé un protocole de routage basé sur des clusters en utilisant un algorithme inspiré du phénomène de butinage des abeilles. Nous avons formulé le problème de clustering comme un problème de programmation linéaire alors que le problème de routage est résolu par une fonction de coûts. L'algorithme de clustering permet la construction efficace des clusters en faisant un compromis entre la consommation d'énergie et la qualité du canal communication au sein des clusters tandis que le routage est réalisé de manière distribuée. Les protocoles proposés ont été intensivement expérimentés sur plusieurs topologies dans différents scénarios de réseaux et comparés avec des protocoles bien connus de clustering et routage. Les résultats obtenus démontrent l'efficacité des protocoles proposés.

**Title:** Bio-inspired Solutions for Optimal Management in Wireless Sensor Networks.

**Keywords:** WSNs, Routing, Clustering and Bio-inspired Computing

**Abstract:** During the past few years, wireless sensor networks witnessed an increased interest in both the industrial and the scientific community due to the potential wide area of applications. However, sensors' components are designed with extreme resource constraints, especially the power supply limitation. It is therefore necessary to design low power, scalable and energy efficient protocols in order to extend the lifetime of such networks. Cluster-based sensor networks are the most popular approach for optimizing the energy consumption of sensor nodes, in order to strongly influence the overall performance of the network. In addition, routing involves non negligible operations that considerably affect the network lifetime and the throughput. In this thesis, we addressed the clustering and routing problems by hiring intelligent optimization methods through biologically inspired computing, which provides the most powerful models that enabled a global intelligence through local and simple behaviors. We proposed a distributed clustering approach based on the nest-sites selection process of a honeybee swarm. We formulated the distributed clustering problem as a social decision-making process in which sensors act in a collective manner to choose their cluster heads.

To achieve this choice, we proposed a multi- objective cost-based fitness function. In the design of our proposed algorithm, we focused on the distribution of load balancing among each cluster member in order to extend network lifetime by making a tradeoff between the energy consumption and the quality of the communication link among sensors. Then, we proposed a centralized cluster-based routing protocol for wireless sensor networks by using the fast and efficient searching features of the artificial bee colony algorithm. We formulated the clustering as a linear programming problem and the routing problem is solved by proposing a cost-based function. We designed a multi-objective fitness function that uses the weighted sum approach, in the assignment of sensors to a cluster. The clustering algorithm allows the efficient building of clusters by making a tradeoff between the energy consumption and the quality of the communication link within clusters while the routing is realized in a distributed manner. The proposed protocols have been intensively experimented with a number of topologies in various network scenarios and the results are compared with the well-known cluster-based routing protocols. The results demonstrated the effectiveness of the proposed protocols.

