Republic Algerian Democratic Popular
Ministry of High Education and Scientific Research

African University of Adrar
Faculty of Sciences and Technology
Department of Mathematics and Computer Science

A THESIS submitted
in partial fulfillment of
the requirements for the degree
MASTER IN COMPUTER SCIENCE
Option: Networks and Intelligent Systems

By:

ZITA Asma

Entitled:

Self Organization in Wireless Sensors Network Using Bio-inspired Mechanism

The Jury Members:

Chairman: Mr. Djilali CHOUGUEUR	MAA	Adrar University
Examiner: Mr. Mohammed OMRI	MCA	Adrar University
Supervisor: Mr. Mohammed KOUHILI	MAA	Adrar University
Co-Supervisor1: Mr. Mohammed DEMRI	MAB	Adrar University
Co-Supervisor2: Mr. El mamoun MAMOUNI	MAB	Adrar University

2012 / 2013

Abstract

Wireless Sensors Network (WSN) become in the last years a searching field, were the most researchers direct their efforts. Since this kind of networks suffers from its low battery criterion which presents a big challenge in face of new application development that can in a way enhance the performance of the network and in the same time affect its lifetime. The good systems should make a tradeoff between improving the network needs and conserving the overall energy. One of this systems is the self-organizing network, which benefits from its self-capabilities where exploring local information in its behaviors to reach a global objective. In our thesis, we propose a self-organizing Ant Colony Algorithm (ACA) in order to find an optimal path in an event monitoring application. The results of simulation show its performance in term of energy conservation and algorithm convergence.

Keywords: self-organizing network, energy, optimal path, Ant Colony Algorithm.

Résumé

Le réseau Capteurs Sans fil (RCSF) devient dans les dernières années un domaine de recherche, où la plupart des chercheurs lui orientent ses efforts, puisque ces réseaux souffrent de son critère de batterie qui présente un grand défi en face de développement de nouvelles applications qui peuvent améliorer la performance du réseau et en même temps, influencent sur sa durée de vie. Les bons systèmes doivent faire un compromis entre l'amélioration des besoins du réseau et la conservation de leur énergie. Un de ces systèmes est le réseau auto-organisé, qui bénéficie de ses auto-capacités où l'exploration d'information locale dans ses comportements pour atteindre un objectif global. Dans notre thèse, nous proposons un algorithme d'auto-organisation de colonie de fourmis dans le but est de trouver un chemin optimal dans une application de surveillance d'événements. Les résultats de la simulation montrent leur performance en terme de conservation d'énergie et de la convergence d'algorithme.

Mots Clés : réseau Auto-organisé, énergie, Chemin optimal, Algorithme de Colonie de fourmis.

Dedication

I dedicate this thesis to the light of my eyes, the most beautiful I see in this world my *mother, which give me the power to complete my work , and support me in my happy and bad time.*

To my father,

My sister and brothers .

To all my familly and friends.

Asma

ACKNOWLEDGEMENTS

بسم الله الرحمن الرحيم

First and foremost, my gratitude is to the one above all, the omnipresent Allah, for giving me the strength to overcome all obstacles and difficulties I hurdle in the completion of this research work.

This thesis would not have been possible to accomplish without the support of many people. I would like to thank my supervisors, Mr. DEMRI Mohammed and Mr. MAMOUNI El Mamoun, for their continuous support and help. I thank all of them for many insightful conversations during the development of the ideas in this thesis.

I don’t know if I can be thankful enough to Dr. OMARI Mohammed for his valuable professional, academic guidance over the last couple of years and who had the merit in any presented works.

I am grateful to my parents and my family , who encouraged and supported me since the beginning of this project, for their emotional support in both happy and unpleasant moments.

Thanks to all my friends.

Table of Contents

Chapter II ***State of the Art***

Chapter III ***Ant Colony Algorithms***

Chapter IV ***Simulation and Results Analysis***

List of figures

List of tables

General Introduction

Wireless Sensor Networks (WSNs) have attracted research interest in recent years due to the significance of the applications field and the advances in sensor technology. In areas where catastrophic events occur such as environmental disasters and battle fields, the network infrastructure is lost and there is an urgent need to build a network in order to monitor the area and to help in rescue operations . Sensors can be deployed manually for small areas, but for large areas other methods like dropping nodes from a plane is used.

The main task of a sensor network is to collect data from the surveillance area and report it to a base station. To achieve this, sensors can form a network with various architectures depending on the application, sensor types, and power constraints. Sensors can send the data directly to the base station or use a multi-hop path to deliver the data. Several routing mechanisms have been proposed to address the energy constraints as well as the network topology.

Self-organization is seen increasingly as an attractive alternative to design for engineering large-scale complex systems such as sensor networks, As systems grow in size, self-organization provides an inherently scalable, flexible and robust way to obtain effective functionality without the need for global communication or control. Where sensors base in its traffic decision on the local information leading to affect the overall network goal.

The objective of our thesis is to design a self-organization sensors network using a bio-inspired mechanism such as natural systems, which they incorporate the features of self-capabilities networks. We direct the implementation of our self-organizing network to routing algorithm in finding the optimal path for traffic management and in the same time getting a trade-off between connectivity maintenance and network lifetime.

We organize our thesis following the construction of four chapters, starting by a general introduction and finalizing by a conclusion and future work.

In the first chapter we give the main concepts about wireless sensor networks, like the application domains, evaluation metrics, routing protocols and categories with examples and their challenges, the design issues of WSNs.

The second chapter starts by defining the concept of self-organization, their meaning in natural systems with basics principals, then its implementation in WSN. We move to the bio- inspired networking and techniques, in the last we present some works that use the bio-inspired techniques to design a self-organizing networks.

We take on consideration in the third chapter the details of the ant colony algorithms, its main variants and we describe the proposed ant colony algorithm.

The fourth chapter of the implementation includes a brief presentation about the C++ builder programming environment, where we apply our Algorithm and both the steps of simulation and the result was discussed. Finally, since the challenges in the wireless sensors field are vary depending on the application need, we suggest some future works surround the concept of self-organization in WSNs using bio-inspired mechanisms.

Chapter 1

Wireless Sensors Network

1.1 Introduction

Wireless sensor networks (WSNs) consist of a large number of autonomous nodes equipped with sensing capabilities, wireless communication interfaces, and limited processing and energy resources [1]. This senor nodes are capable to operate, eventually to collect, process and transmit data of the environment over its radio to the base station situate in the center or the edge of the network called the “Sink” . Usually, the nodes are statically deployed over vast areas.

However, they can also be mobile and capable of interacting with the environment. In these cases, the network is more appropriately referred to as a robotic network and/or as a sensor-actor network. WSNs can be employed in a wide spectrum of applications in both civilian and military scenarios, including environmental monitoring, surveillance for safety and security, automated health care, intelligent building control, traffic control, object tracking, etc [1].

In this Chapter, we will introduce a description of the main sensor node components that can complete the intended goal from sensing the environment , processing to transmission of data. We will present then the applications field of a WSNs such as environmental and event monitoring using in the most research domains following by the architecture of a WSN topologies, and for measurement some evaluation metrics are mentioned . Finally, we will present the well known routing protocol categories depending on the structure of the network or the routing operations, and for a best network conception the design issues for a WSN will be discussed.

1.2 The components of a sensor node

A sensor node constitutes of the principals components as follow:

❖ **Sensing Unit:** is composed of two subunits, a capture device which removes the physical information of the local environment and a converter analog / digital called ADC ("Analog to Digital Converter "). Sensor is responsible for providing the analog / digital signals. Latter, converts these signals into a signal understandable by the digital processing unit.

❖ **Processing Unit:** execute the communication protocols that organize the collaboration between sensor nodes to complete the query. It must, at minimum, serve as

an effective interface to the sensor module and regulate data flow from the sensing unit to the radio transceiver.

❖ **Transceiver Unit:** responsible for transmission and reception of data over a wireless medium.

❖ **Power Unit:** responsible for control the remaining energy and the lifetime of the sensor node, since the energy resources of the sensor are limited, it will affect the node lifetime and the whole network as well.

❖ **Location Finding System:** supply the information about the location required for the routing techniques.

❖ **Mobilizer:** it is called if the node should be move to complete the query to be processed **[2]**.

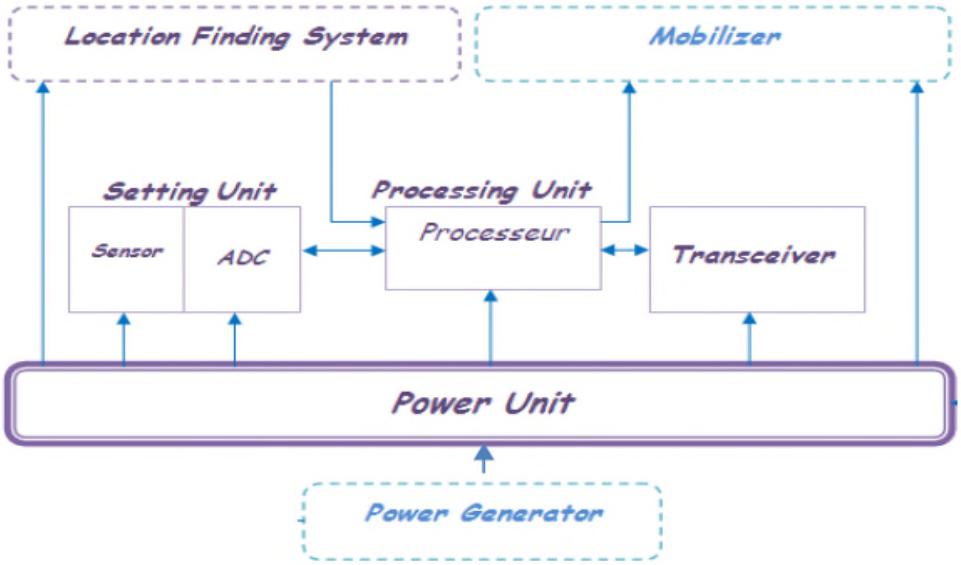

Figure 1.1 The components of a sensor node

In term of functionally, each sensor has a communication radius (Rc) and a radius of sensation (Rs). Figure 1.2 shows the areas defined by these two radiuses for the sensor A. The communication area is the area where the sensor A can communicate with other sensors (sensor B in Figure 1.2). On the other hand, the area of sensation is the area where the sensor can capture the event [3].

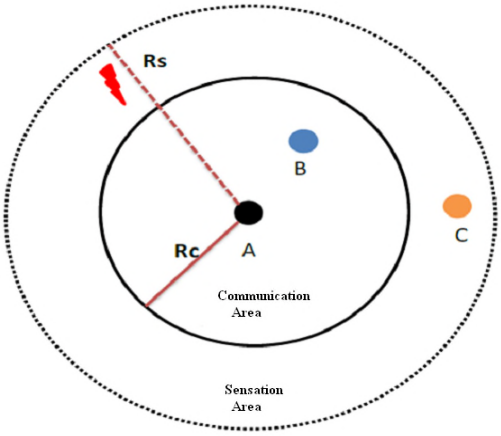

Figure 1.2 Communication and sensation radius of sensor node.

1.3 Wireless sensor network applications

The majority of wireless sensor network applications can be categorized into three classes: environmental data collection, event-based monitoring and object tracking.

1.3.1 Environmental data collection

Environmental data collection represents a class of sensor network applications with enormous potential benefits for scientific research communities and the society.

The intimate connection with the physical environment allows sensor networks to provide localized measurement and details information. Researchers collect sensor measurement from an environment during a long period in order to detect trends and interdependencies. This measurement would gather from hundreds of sensor nodes spread over several miles in field environment.

Figure 1.3 Example of environment data collection

The environmental data collection application is characterized by having a large number of nodes continually sensing and transmitting data back to a set of base stations.
These networks generally require very low data rates and extremely long lifetimes. Depending on application's requirements for sensing coverage and resolution, the network size may vary from hundreds to hundreds of thousands of nodes [4].

1.3.2 Event-based monitoring

Event-based monitoring is another class where nodes are responsible for event detection. Usually the event does not happen most of the time, so sensor nodes need periodically check neighbor information to ensure nodes functionality. When an event discovered in the environment, the nodes collaborate with each other, track the changes and report messages to the base station. The network can suffer from nodes failure; they cannot monitor sensing the areas which affect to some security problems. The topology control of an event detection monitoring will differ from that of data collection network; the large part of the energy will spent to confirming the functionality of neighboring nodes and performing event detection compared to data message transmission[4].

Figure 1.4 Examples of event-based monitoring

1.3.3 Object tracking

The third class of WSN applications is to track an object status in real time; the most visible and publicly discussed example is the usage of RFID (radio frequency identification) tags in supply chain management. In traditional inventory system, when merchandise passes a checkpoint, its barcode is scanned by a reader, so that the system can track the data flow through different checkpoints. However, it is time consuming inefficient to track the object because of the inflexible barcode scanning processing.

With wireless sensor networks, object can be identified and tracking by attaching RFID tags. The tags can be active, which are able to announce the presence of the devices [4].

Sensors can also be used in the health care and medicine domain, where could keep the medical records and data analysis of patient health, e.g. by being attached to a patient body in order to measure their vital signs (like blood pressure and heartbeats) and send alerts to their doctors if needed, hence identifying and early resolving various life threatening and health problems. Sensors may also monitor the behaviors of an elderly and remind him for example about where he left his pill box. At home also, WSN could be used to monitor a patient's weight or daily blood sugar levels for diabetics [5].

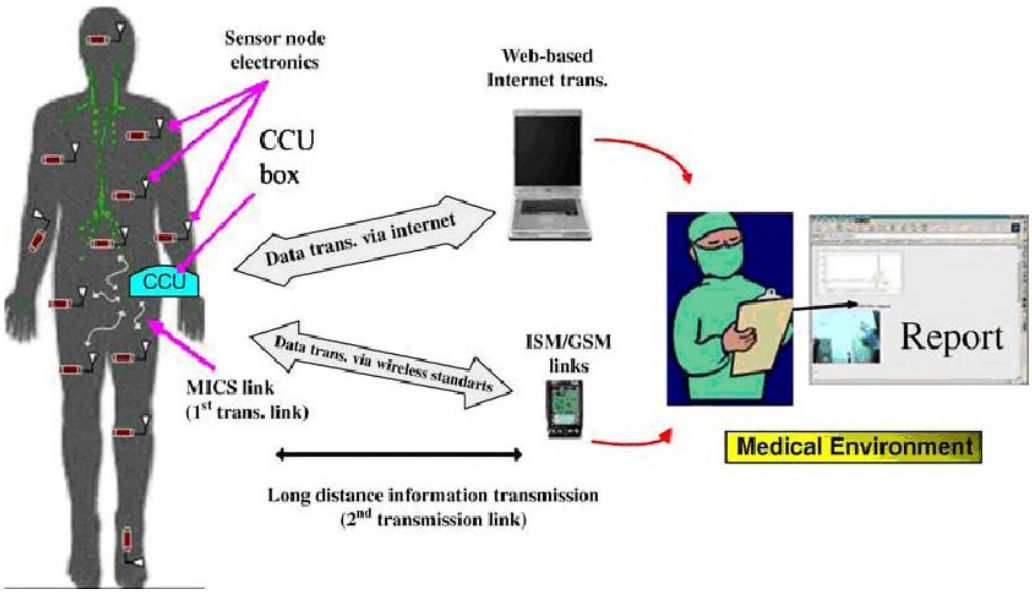

Figure 1.5 Object tracking: medical application.

One of the major applications for WSN in the security domain is intrusion detection. An intrusion is defined as an action that attempts to compromise the confidentiality, integrity, or availability of a resource. Intrusion detection systems (IDS) have been long investigated for wired and wireless networks to detect, identify, and eject an intruder before any damage is done [5]. An IDS for traditional networks function under the assumption that normal activity and intrusion activity have distinct behavior. Therefore, when an intrusion happens, a deviation in the activity of the system should be recognized. On the other hand, designing an IDS for WSN is somewhat a challenge, since WSN have some special characteristics over traditional wireless networks , like topology knowledge, failure tolerance , reliability…etc [5].

1.4 Wireless Sensor Network Architecture

Wireless sensor networks can be organized in flat or hierarchical topologies. While flat topologies are more suited for small networks, hierarchical topologies allow the deployment of bigger networks, more scalable and easier to manage. One popular subclass of hierarchical networks is tree-based clustered networks [5].

1.4.1 Flat sensor networks

A flat wireless sensor network is a homogeneous network where all nodes are identical in terms of battery and functions, except the "Sink". The latter acts as a gateway and is responsible for the transmission the collected information to the end user [2].

When the network becomes crowded, the big number of neighbors for each node will affect its allocated bandwidth and its experienced delay. Moreover, the number of possible routes will increase, and many nodes may try to talk to distant nodes directly, using a large transmission power [5].

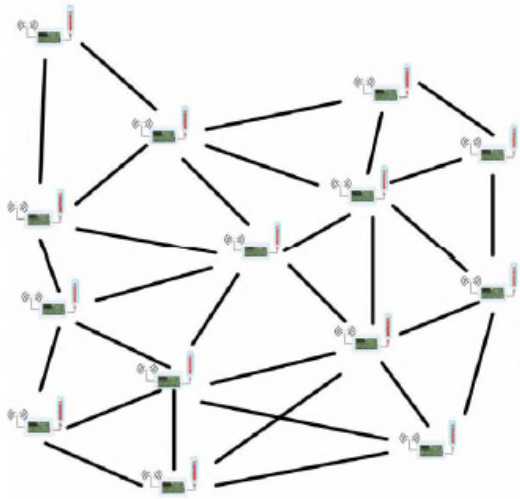

Figure 1.6 Flat WSN Architecture.

1.4.2 Hierarchical sensor networks

A hierarchical architecture has been proposed to reduce the cost and complexity of most of the sensor nodes. It consists of introducing a set of nodes more expensive and more powerful; creating an infrastructure that relieves the majority of simple nodes at low cost many network functions [2]. The network is tree-shaped and structured in layers, and nodes could have different roles [5].

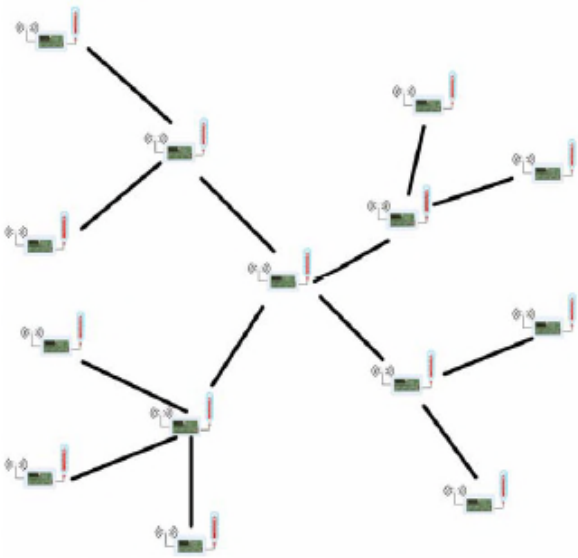

Figure 1.7 Hierarchical WSN Architecture.

Wireless sensor networks that contain typically hundreds of nodes and need to be easy scalable find this model very interesting, especially that it has the same data flow model as WSN where data is collected at the leaf nodes (i.e. the sensors) and relayed to the root (i.e. the sink) [5].

1.5 Evaluation Metrics

The key evaluation metrics for wireless sensor networks are lifetime, coverage, cost and ease of deployment, response time and security. Their importance is discussed below. One result is that many of these evaluation metrics are interrelated. Often it may be necessary to decrease performance in one metric, such as sample rate, in order to increase another, such as lifetime. Taken together, this set of metrics form a multidimensional space that can be used to describe the capabilities of a wireless sensor network.

1.5.1 Lifetime

Critical to any wireless sensor network deployment is the expected lifetime. The goal of both the environmental monitoring and security application scenarios is to have nodes placed out in the field, unattended, for months or years.

The primary limiting factor for the lifetime of a sensor network is the energy supply. Each node must be designed to manage its local supply of energy in order to maximize total network lifetime. In many deployments it is not the average node lifetime that is important, but rather the minimum node lifetime. In the case of wireless security systems, every node must last for multiple years. A single node failure would create a vulnerability in the security systems.

In some situations it may be possible to exploit external power, perhaps by tapping into building power with some or all nodes. However, one of the major benefits to wireless systems is the ease of installation.

In most application scenarios, a majority of the nodes will have to be self powered. They will either have to contain enough stored energy to last for years, or they will have to be able to scavenge energy from the environment through devices, such as solar cells. Both of these options demand that that the average energy consumption of the nodes be as low as possible.

The most significant factor in determining lifetime of a given energy supply is radio power consumption. In a wireless sensor node the radio consumes a vast majority of the

system energy. This power consumption can be reduced through decreasing the transmission output power or through decreasing the radio duty cycle [6].

1.5.2 Coverage

Coverage is the primary evaluation metric for a wireless network. It is usually defined as a measure of how well and for how long the sensors are able to observe the physical space. The first step in deploying a wireless sensor network is determining what it is exactly that you are attempting to monitor. Typically you would monitor an entire area, watch a set of targets, or look for a breach among a barrier.

Coverage of an entire area otherwise known as full or *blanket coverage* means that every single point within the field of interest is within the sensing range of at least one sensor node. Ideally you would like to deploy the minimum number of sensor nodes within a field in order to achieve blanket coverage [7].

A *barrier coverage* achieves a static arrangement of sensor nodes that minimizes the probability of undetected penetration through the barrier.

While *Sweep coverage* concerned moving a number of sensor nodes across a sensing field, such that it addresses a specified balance between maximizing the detection rate and minimizing the number of missed detections per unit area [8].

1.5.2.1 K-coverage

We consider a square region D of unit area where n sensors are deployed independently and uniformly.

A point in the region D is said to be *k-covered* if it is within the sensing radius of at least k active sensors. The region D is said to be *k-covered* if every point in D is *k-covered* [9].

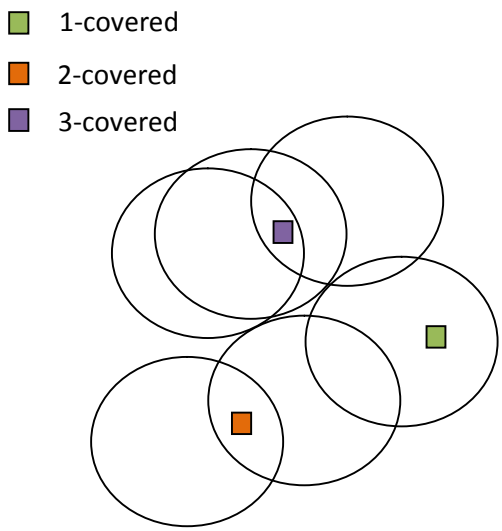

Figure 1.8 points k-covered [10].

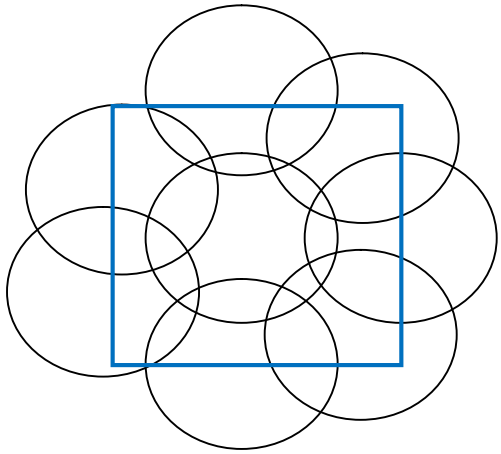

Figure 1.9 k-covered region [10].

1.5.2.2 Connectivity with coverage

In addition to coverage it is important for a sensor network to maintain connectivity. *Connectivity* can be defined as the ability of the sensor nodes to reach the data sink. If there is no available route from a sensor node to the data sink then the data collected by that node cannot be processed [7].

The areas of coverage and connectivity are closely related. Each are necessary conditions for a functional wireless sensor network. An important principal to consider is that if the communication range of the sensors is at least twice that of the sensing range then coverage of an area implies connectivity [7].

1.5.3 Cost and ease of deployment

A key advantage of wireless sensor networks is their ease of deployment. For system deployments to be successful, the wireless sensor network must configure itself automatically for any possible physical node placement [6].

The initial deployment and configuration is only the first step in the network lifecycle. In the long term, the total cost of ownership for a system may have more to do with the maintenance cost than the initial deployment cost. The security application scenario in particular requires that the system be extremely robust. In addition to extensive hardware and software testing prior to deployment, the sensor system must be constructed so that it

is capable of performing continual self-maintenance. When necessary, it should also be able to generate requests when external maintenance is required [6].

In addition to an initial configuration phase, the system must also adapt to changing environmental conditions. Throughout the lifetime of a deployment, nodes may be relocated or large physical objects may be placed so that they interfere with the communication between two nodes. The network should be able to automatically reconfigure on demand in order to tolerate these occurrences [6].

1.5.4 Response Time

Particularly in an alarm application scenario, system response time is a critical performance metric. An alarm must be signaled immediately when an intrusion is detected. Despite low power operation, nodes must be capable of having immediate, high-priority messages communicated across the network as quickly as possible.

Response time is also critical when environmental monitoring is used to control factory machines and equipment. Many users envision wireless sensor networks as useful tools for industrial process control. These systems would only be practical if response time guarantees could be met.

The ability to have low response time conflicts with many of the techniques used to increase network lifetime. Network lifetime can be increased by having nodes only operate their radios for brief periods of time. If a node only turns on its radio once per minute to transmit and receive data, it would be impossible to meet the application requirements for response time of a security system [6].

Response time can be improved by including nodes that are powered all the time. These nodes can listen for the alarm messages and forward them down a routing backbone when necessary.

1.5.5 Security

As we consider security oriented applications, data security becomes even more significant. Wireless sensor networks must be capable of keeping the information they are collecting private from eavesdropping. Not only must the system maintain privacy, it must also be able to authenticate data communication. It should not be possible to introduce a false alarm message or to replay an old alarm message as a current one.

A combination of privacy and authentication is required to address the needs of all three scenarios. Additionally, it should not be possible to prevent proper operation by interfering with transmitted signals [6].

Use of encryption and cryptographic authentication costs both power and network bandwidth. Extra computation must be performed to encrypt and decrypt data and extra authentication bits must be transmitted with each packet. This impacts application performance by decreasing the number of samples than can be extracted from a given network and the expected network lifetime [6].

1.6 Routing protocols in WSN

Routing is a process of determining a path between source and destination for data transmission. In WSNs the network layer is mostly used to implement the routing of the incoming data and Routing protocol is an important factor in design of a communication Stack. Routing protocols, designed for sensor networks, must accomplish high reliability.

Sensor nodes are constrained in energy supply and recharging sensor nodes is normally impractical due to their nature of deployment. Therefore, energy saving is an important design issue in Wireless sensor networks.

While the objective of traditional networks is to achieve high quality of service, sensor network protocols must focus additionally on power conservation also to maximize the network lifetime [11].

1.6.1 Design challenges for wsn routing protocol

The design of routing protocol in WSNs is influenced by many challenging factors as summarized below.

Node deployment

Node deployment in WSNs is application dependent and affects the performance of the routing protocol. The deployment is either deterministic (manual) or self-organizing (random). If the resultant distribution of nodes is not uniform, optimal clustering becomes necessary to allow connectivity and enable energy efficient network operation. The position of the sink or the cluster-head is very crucial in terms of energy efficiency and performance [11, 12].

Sensor locations

Another challenge that faces the design of routing protocols is to manage the locations of the sensors. Most of the proposed protocols assume that the sensors either are equipped with global positioning system (GPS) receivers or use some localization technique to learn about their locations.

Data delivery models

One of the features that make WSNs clearly different from other types of networks is the structure of the traffic patterns. The most common traffic patterns present in WSNs include: continuous monitoring, event-driven, query-driven and some hybrid combination of these**.** In the continuous delivery model, data packets are sent back to the monitoring node(s) at regular intervals. In event-driven and query-driven models, the transmission of data is triggered when an event occurs or the sink generates a query.

The routing protocol is highly influenced by the data delivery model, especially with regard to the minimization of energy consumption and route stability [1, 11].

Node capabilities

In a sensor network, different functionalities can be associated with the sensor nodes. For some reasons a node can be dedicated to a particular special function such as relaying, sensing and aggregation, hence the energy of the node that has all this functionalities at the same time might quickly drain.

Data aggregation

Since sensor nodes may generate significant redundant data, similar packets from multiple nodes can be aggregated so that the number of transmissions is reduced, saving on the limited available hardware resources and reducing the negative effects due to radio interference. Data aggregation technique has been used to achieve energy efficiency and data transfer optimization in a number of routing protocols [1, 12].

Scalability

The number of sensor node in the target area may be on the order of hundreds or thousands or more so protocols should be able to scale to such high degree and take advantage of the high density of such networks [11].

1.6.2 Sensor routing categories

WSN routing protocols can be classified in two ways, according to the network structure and according to the protocol operations, the Figure 1.10 shows the classification of WSN routing protocols.

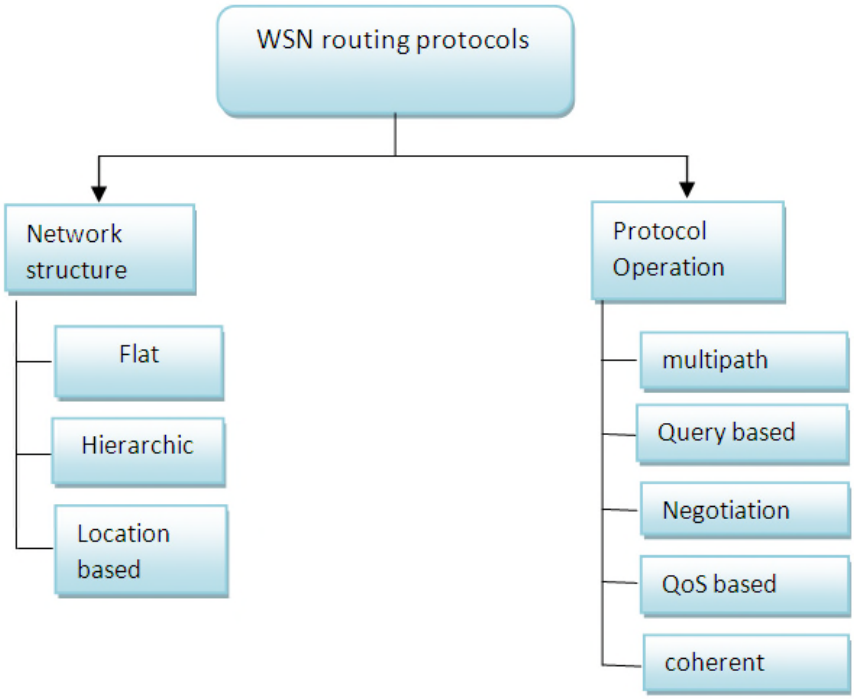

Figure 1.10 Classification of routing protocols in WSN.

1.6.2.1 Network structure

According to network flow model, the routing protocols are divided into flat routing, hierarchical-based and location-based routing. In flat-based routing, all nodes play the same role. In hierarchical-based routing, however, nodes will play different roles in the network. In location-based routing, sensor nodes positions are exploited to route data in the network [11].

a) Flat based routing

In these networks, all nodes play the same role and there is absolutely no hierarchy. Flat routing protocols distribute information as needed to any reachable sensor node within the sensor cloud . There is no effort to organize the network and its traffic. The

effort is made only to discover the best hop by hop route source to a destination by any path [13].

- **Directed diffusion (DD):**

Directed diffusion is a query driven protocol in which a request for a precise kind of data is interpreted as an interest with a certain data rate [14]. Directed diffusion was designed for collecting and publishing the information in WSNs. It has been developed to address the requirement of data flowing from the sink toward the sensors, i.e., when the sink requests particular information from these sensors [15].

In order to construct the route between the sink (inquirer) and the sensors that interest to the sink's request, there are four stages as the figure 1.11 show.

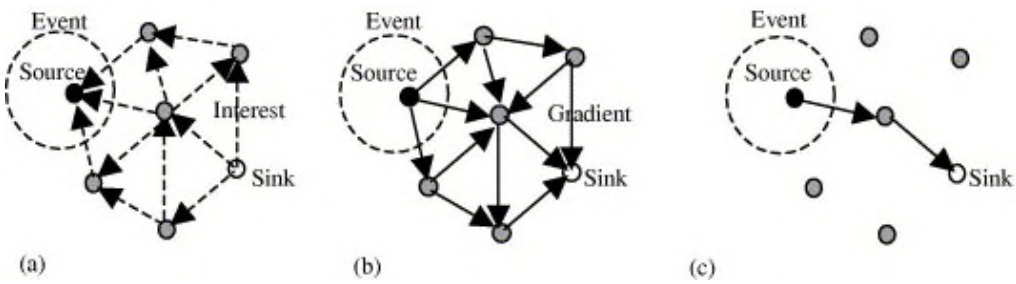

Figure 1.11 Directed Diffusion protocol phases. (a) Interest propagation, (b) initial gradients setup, (c) data delivery along reinforcement.

b) Hierarchical routing

This class of routing protocols sets out to attempt to conserve energy by arranging the nodes into clusters as shown in Figure 1.12. Nodes in a cluster transmit to a head node within close proximity which aggregates the collected information and forward this it to the base station [16].

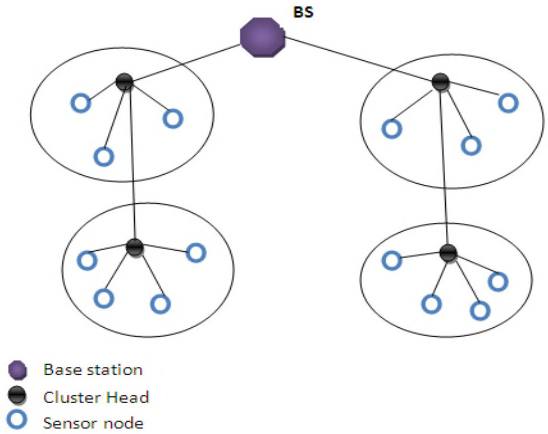

Figure 1.12 Hierarchical routing in WSN.

- **Low-energy adaptive clustering hierarchy (LEACH):**

LEACH is the first and most popular energy-efficient hierarchical clustering algorithm for WSNs that was proposed for reducing power consumption. In LEACH, the clustering task is rotated among the nodes, based on duration. Direct communication is used by each cluster head (CH) to forward the aggregated data to the base station (BS). It uses clusters to prolong the life of the wireless sensor network [11].

The operation of LEACH protocol has been divided into two phases, the *setup phase* and the *steady state phase*. In the *setup phase*, the clusters are organized and CHs are selected this decision is made by the node *n* choosing a random number between 0 and 1 [11].

If the number is less than a threshold *T(n)*, the node becomes a cluster-head for the current round. The threshold is set as:

$$T(n) = \begin{cases} \frac{P}{1-P*(r \bmod 1/P)} & if\ n \in G \\ 0 & otherwise \end{cases} \quad (1.1)$$

Where

p : the desired percentage of cluster heads over all nodes in the network.

r: the number of rounds of selection (the current round).

G: the set of nodes that are not selected in round 1/p.

c) Location based routing

Most of the routing protocols for sensor networks require location information for sensor nodes [16]. In most cases location information is needed in order to calculate the distance between two particular nodes so that energy consumption can be estimated. Generally two techniques are used to find location, one is to find the coordinate of the neighboring node and other is to use GPS (Global Positioning System). Since, there is no addressing scheme for sensor networks like IP addresses and they are spatially deployed on a region, location information can be utilized in routing data in an energy efficient way [11].

- **Geographic adaptive fidelity (GAF):**

Is an energy-aware location-based routing algorithm that conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity. Each node uses its GPS-indicated location to associate itself with a point in the virtual grid. Nodes associated with the same point on the grid are considered equivalent in terms of the cost of packet routing [17].

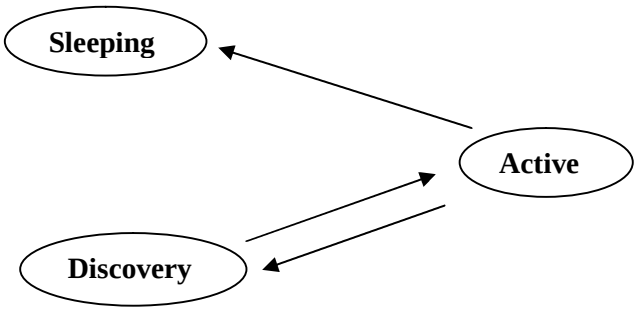

Figure 1.13 State transition diagram of GAF.

The state transition diagram of GAF has three states, namely, discovery, active, and sleeping. When a sensor enters the sleeping state, it turns off its radio for energy savings. In the discovery state, a sensor exchanges discovery messages to learn about other sensors in the same grid. Even in the active state, a sensor periodically broadcasts its discovery message to inform equivalent sensors about its state.

1.6.2.2 Protocol operation

According to protocol operation, routing protocols can also be classified into five routing techniques. The Table 1.1 illustrate the principle of each category and some examples.

Protocol operations	Description	Examples
Multipath Routing	• Use multiple paths in order to enhance network performance	Sensor-Disjoint Multipath, Braided Multipath, N-to-1 Multipath Discovery
Query based Routing	• Based on propagation of queries throughout the network by the nodes which require some data	SPIN, DD, SPEED, SAR
Negotiation based Routing	• Use high-level data descriptors in order to eliminate redundant data transmissions through negotiation	SPIN, DD, SAR
QoS based Routing	• Has to balance between energy consumption and data quality	SAR, SPEED, Energy-aware routing
Coherent /no Coherent based Routing	• The data is forwarded to aggregators after minimum processing/ Local data processing	Protocols using coherent /no coherent processing algorithms

Table 1.1 Description of routing operation based protocols.

1.7 WSN design issues

The wireless sensor networks are application specifics, so the design should take on consideration the main following objectives:

- **Low power consumption:** Since sensor nodes are powered by battery and it is often very difficult or even impossible to charge or recharge their batteries, it is crucial to reduce the power consumption of sensor nodes so that the lifetime of the sensor nodes, as well as the whole network is prolonged.

- **Scalability*:*** Since the number sensor nodes in sensor networks are in the order of tens, hundreds, or thousands, network protocols designed for sensor networks should be scalable to different network sizes.

- **Reliability**: Since nodes are battery-powered and communications are radio-based, nodes are more susceptible to failures. The information collected by individual node should be aggregated to give more accurate and reliable results. Sensor network should be reliable and be able to provide relevant data through information gathering techniques.

- **Self-configurability*:*** In sensor networks, once deployed, sensor nodes should be able to autonomously organize themselves into a communication network and reconfigure their connectivity in the event of topology changes and node failures.

As Network grow in size, self-configuration provides an inherently scalable, flexible and robust way to obtain effective functionality without the need for global communication or control.

A self-organizing networks are robust: they can withstand a variety of errors, perturbations, or even partial destruction. They will repair or correct most damage themselves, getting back to their initial state. When the damage becomes too great, their function will start to deteriorate, but "gracefully", without sudden breakdown. They will adapt their organization to any changes in the environment, learning new "tricks" to cope with unforeseen problems. Out of chaos, they will generate order. Seemingly random perturbations will help rather than hinder them in achieving an ever better organization.

- **Adaptability*:*** In sensor networks, a node may fail, join, or move, which would result in changes in node density and network topology. Thus, network protocols designed for sensor networks should be adaptive and sensitive to the dynamic environment where they are deployed.

- **Channel utilization:** Since sensor networks have limited bandwidth resources, communication protocols designed for sensor networks should efficiently make use of the bandwidth to improve channel utilization.

- **Fault tolerance:** Sensor nodes are prone to failures due to harsh deployment environments and unattended operations. Thus, sensor nodes should be fault tolerant and have the abilities of self testing, self-calibrating, self-repairing, and self-recovering.
- **Security*:*** A sensor network should introduce effective security mechanisms to prevent the data information in the network or a sensor node from unauthorized access or malicious attacks.
- **QoS support*:*** In sensor networks, different applications may have different quality of service (QoS) requirements in terms of delivery latency and packet loss. Thus, network protocol design should consider the QoS requirements of specific applications [12, 18].

1.8 Conclusion and Summary

In this first Chapter, we have presented some concepts of wireless sensor networks from the physical components to the high levels notions , We have briefly introduced some generalities about the applications that a WSN can be designed for and the essence of the flat and hierarchical architectures of a network as a basic manners for organizing sensors deployment. We have defined the evaluation metrics like coverage, lifetime , ..etc. while in the routing protocol section , we have described the design challenges of a routing mechanism and the main classification. Finally, the design issues for a WSN such as scalability, Self-configuration, Fault tolerance , and more was discussed as a highlight for designing a robust and scalable sensor networks.

Since the main goal of this study is the conception of a self-organization wireless sensor network using bio-inspired mechanisms, the next chapter will present some state-of-the-art techniques inspired from the behaviors of a natural organisms as a self-organizing systems. Autonomy, and simplicity are their properties to deal with the environmental changes depending on local information and the feedback to keep their system at a high performance.

Chapter II

State of the Art

2.1 Introduction

Since Sensors network requirements such as efficiency , adaptability and scalability become the main addressing goals for a robust network. Therefore , applying a self-organizing methods give the network the possibility of gains this features . Due to simplicity of the bio-inspiring mechanisms to handle a global tasks based on a local behaviors, there are suitable for the autonomous WSN applications for performing their main goal with a minimum energy consumption.

We will introduce in the next sections the concept of self-organization as a natural phenomena and his application in WSN, the classification of self-organization methods. then, we present a brief definition of bio-inspired networking, techniques and specification of special algorithms. Finally, some works based on self-organization using bio-inspired mechanisms will be described.

2.2 Self-Organization

Self-organization is not an invention nor it was developed by an engineer. The principles of self-organization have been evolved in nature and we finally managed to study and apply these ideas to technical systems. First articles about self-organization date back in the early 1960ies. *Ashby* and *von Foerster* analyzed self-organizing mechanisms. *Eigen* finally made the term self-organization popular in natural and engineering sciences [19].

Self-organization can be summarized as the interaction of multiple components on a common global objective. This collaborative work is done without central or decentral control. Instead, the interaction is done using a local context, that can be changed and adapted by each individual and, therefore, affects the behavior of other individuals [19].

2.2.1 Definition

Self-organization is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of a system. Moreover, the rules specifying interactions among the systems 'components are executed using only local information, without reference to the global pattern [19].

One of the first definitions of self-organization by Yates *et al*. (1987) characterizes self-organization as follows: ‘ Technological systems becomes organized by commands from outside, as when human intentions lead to the building of structures and machines. But many natural systems become structured by their own internal processes: these are the self-

organizing systems, and the emergence of order within them is a complex phenomenon that intrigues scientists from all disciplines [20].

2.2.2 Self-Organization as Natural Phenomenon

Self-organization is a natural phenomenon of distributed systems, where components interact on a microscopic level leading to global behaviors that emerge on a macroscopic level. Such emergent behaviors are unintended and thus may be undesirable.
For example, unintended self-organizing phenomena have been observed in the Internet, cellular wireless networks, and computing grids.
As a design strategy, system components may be endowed with local rules intended to yield desired global behaviors [20].

There are many examples visible in our everyday life that represent a natural self-organization system , the most cited examples are perhaps the oscillating reaction of the *Belousov-Zhabotinskiy* reaction (Winfree 1972). The oscillation occurs due to two simultaneously conducted processes: *reaction* and *diffusion* . these processes cause a system in which the concentrations of reactants and products oscillate temporally and spatially. These oscillations lead to the creation of spectacular patterns, such as those shown in Figure 2.1 [20].

Figure 2.1 Pattern formation in the *Belousov-Zhabotinskiy* reaction [20].

Self-organizing systems have some essentials properties as described below [20]:

- No central control: there is no global control system or global information Available. Each subsystem must perform completely autonomously.
- Emerging structures: the global behavior or functioning of the system emerges in the form of observable patterns or structures.

- Resulting complexity: even if the individual subsystems can be simple and perform basic rules, the resulting overall system becomes complex and unpredictable.
- High scalability: there is no degradation if more subsystems are added to the system.

2.2.3 Basic Principles of Self-Organization

There are three major principles of self-organization mechanisms:

- *Feedback loops*;
- *Interaction between individuals*;
- *Probabilistic methods* .

This principles will be discussed in more details:

2.2.3.1 Feedback loops

One major component in understanding the interaction of components producing a complex pattern are *positive* and *negative feedback* loops [21].

Positive feedback acts as an amplifier for a given effect. In order to prevent overreactions and mis-regulations [21], it generally promotes changes in a system.

Self-enhancement, amplification, facilitation are all terms used to describe positive feedback. Actually, positive feedback is one of the major mechanisms that take self-organization possible. The amplifying nature of positive feedback that leads to the development of pattern means also that it has the potential to produce destructive explosions or implosions in any process in which it play a role. Positive feedback can be used to find a stable state or in additional feedback , the system can leave this stable state in order to coverage to another one [20].

Negative feedback is used to efficiently control the system behavior [21]. It can be created in the form of rules that have been developed to keep the system state within a given parameter range [20].

Multiple terms are used in literature for both positive feedback and negative feedback. In general, positive feedback is equal to reaction, amplification and promotion. Negative feedback is often described as diffusion, suppression and inhibition [20].

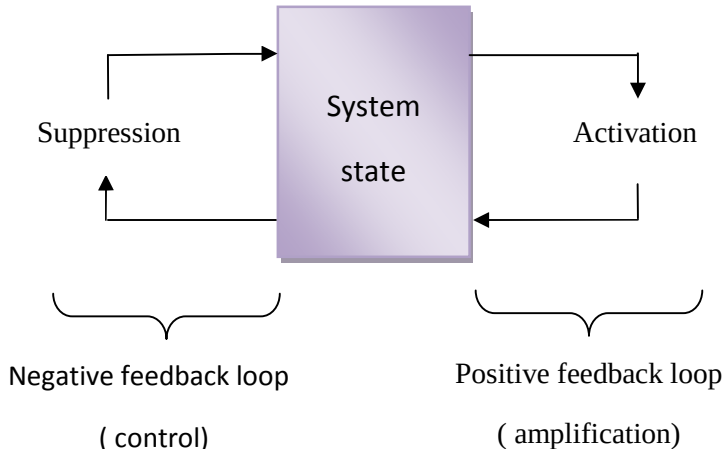

Figure 2.2 Positive and Negative feedback used for behavior control of autonomous systems [20].

2.2.3.2 Interaction between individuals

Information transfer between individuals is necessary to update the local state. There are two ways to conduct such interactions:

- Information transfer between individuals, i.e. direct interaction or communication between related subsystems (neighboring individuals) via signals;
- Interactions with the environment, i.e. indirect information flows via cues arising from work in progress. This process is also known as stigmergy (Di Caro and Dorgio 1998).

  The system influences the environment (it produces some effect). This effect can be measured and directly increases or decreases the activation capabilities to the system behavior [20,21].

2.2.3.3 Probabilistic methods

A third component for successfully building self-organizing systems is probability. Basically, all self-organizing systems include probabilistic techniques. Such mechanisms can be used either to entirely organize the local behavior of a single system or at least for parameter settings of other deterministic algorithms. The use of probabilistic techniques has several advantages , one of them is the break out of local optima [20].

2.3 Self-Organization in WSN

As we have discussed in first section, wireless sensor networks as a self-organizing systems lack any central command and control. The management and control is distributed in a sense that each sensor node only controls itself and nodes need to do tasks in coordination with each other to reach the objective of the network.

This causes the requirement for self-organization. Algorithms are required for sensor networks that can enable self-organization of individual components. At each level routing, discovery of resources, security etc. algorithms need to be developed which can support self-organization.

Self organization in systems also gives the benefit of adaptability. Keeping in view our discussion of self organization, it can be seen that wireless sensor networks can be designed as self organizing systems. The working unit is a sensor node. More often, sensor nodes take different roles in the overall system.

Moreover, sensor nodes work in coordination with each other, providing different support functionality to other nodes such as connectivity and routing of messages [22].

2.3.1 Applying Self-Organization in Wireless Networks

To achieve self organization in the network, the algorithms used in different modules of each node need to be designed in self-organizing fashion.

Deployment

A sensor network is composed of a large number of sensor nodes depending on the applications requirements. The nodes can be deployed in a predetermined position or arbitrarily. The arbitrary deployment allows random deployment in inaccessible areas. When sensors are deployed randomly, it requires the nodes to be self-organized without any central command and control (some nodes can be preconfigured to initiate the network establishment). Thus, self organizing protocols and algorithms are required. Each node organizes itself locally [22].

Network Maintenance

During network operations, network management and maintenance is required on different levels. Topology requirements and shape may change due to failures in nodes. Topology maintenance is required , so the nodes need to share their current state and resources available e.g., remaining energy and number of neighbors of the node.

To keep balance of energy among the nodes, the states and roles of nodes also need to be changed. In case of failures or security issues, maintenance may be required for reliability and resilience of network [22].

Discovery and Configuration

After the deployment of nodes, an autonomous setup of network is required. It requires the nodes to coordinate in discovering each other and making a network topology. Nodes take different states and roles such as active node and data aggregation node.

To be self organized, nodes must share some resources such as electromagnetic spectrum and bandwidth. The current context of the system need to be shared e.g., the data sink node and router nodes.

Routing

The usual addressing techniques which are used in wired networks are not used in wireless sensor networks due to random deployment of nodes.

Different naming techniques are used to identify the nodes. These naming techniques are required for self organized routing. For instance, nodes may take different tags such as sink node, heat sensor, data aggregator etc. And the data is routed using energy efficient routing, e.g., selecting the path which utilizes least energy consumption.

2.3.2 Classification of self-organization methods

The categorization of self-organization methods try to avoid global state information in order to increase the hole objective. Those categories are discussed in the following:

- **Location-based mechanisms**

Geographical positions or affiliation to a group of surrounding nodes, i.e. clustering mechanisms, are used to reduce necessary state information to perform routing decisions or synchronizations. Usually, similar methods as known for global state operations can be employed in this context. Depending on the size of active clusters or the complexity to perform localization methods, such location based mechanisms vary in communication and processing overhead [19].

- **Neighborhood information**

Further state reduction can be achieved by decreasing the size of previously mentioned clusters to a one-hop diameter. In this case, only neighborhood information is available to perform necessary decisions.

Usually, hello messages are exchanged in regular time periods. This keeps the neighborhood information up-to-date and allows the exchange of performance measures such as the current load of a system [19].

- **Probabilistic algorithms**

In some cases, it is useful to store no state information at all. For example, if messages are very infrequently exchanged or in case of high mobility, pure probabilistic methods can lead to optimal results. Statistical measures can be used to describe the behavior of the overall system. Obviously, no guarantee can be given that a desired goal will be reached [19].

2.4 Bio-inspired Networking

The term bio-inspired has been introduced to demonstrate the strong relation between a particular system ,which has been proposed to solve a specific problem and a biological system. There are three main area of bio-inspired research that can be distinguished :

- Bio-inspired computing : represents a class of algorithms focusing on efficient computing (e.g., for optimization processes and pattern recognition).
- Bio-inspired systems: constitute a class of system architectures for massively distributed and collaborative systems (e.g., for distributed sensing and exploration).
- Bio-inspired networking :is a class of strategies for efficient and scalable networking under uncertain conditions (e.g., for autonomic organization in largely distributed systems) [23**].**

Bio-inspired computing and system design have already become widely visible, due to the natural characteristics of biological systems and their solutions to problems. Technical systems and networking inspirit from biology the behaviors to solve some problems. This systems are:

- Adaptively to the varying environmental circumstances;
- Robustness and resilient to the failures caused by internal or external factors

- Ability to achieve complex behaviors on the basis of a usually limited set of basic rules;
- Ability to learn and evolve itself when new conditions are applied;
- Effectiveness management of constrained resources with an apparently global intelligence larger than the superposition of individuals ;
- Ability to self-organize in a fully distributed fashion, collaboratively achieving efficient equilibrium ;
- Survivability despite harsh environmental conditions due to its inherent and sufficient redundancy.

These characteristics constitute the basis for different levels of inspiration by biological systems towards the development of different approaches and algorithms at each of the networking layers for efficient, robust and resilient communication and information networks [23].

2.4.1 Bio-inspired Techniques

The development in the area of bio-inspired engineering is relying on various research fields including swarm intelligence, the artificial immune system, evolutionary and genetic algorithms, cell and molecular biology based approaches [21].we take the swarm intelligence, genetic algorithms and the artificial immune system as an examples as follow:

2.4.1.1 Swarm Intelligence

Swarm intelligence (SI), which is an artificial intelligence (AI) discipline based on collective behavior of self-organized systems, it is concerned with the design of intelligent multi-agent systems by taking inspiration from the collective behavior of social insects such as ants, termites, bees, and wasps, as well as from other animal societies such as flocks of birds or schools of fish. Colonies of social insects have fascinated researchers for many years, and the mechanisms that govern their behavior remained unknown for a long time. Even though the single member of these colonies are non-sophisticated individuals, they are able to achieve complex tasks in cooperation [24].

According to de Castro(2006), five basic principles of SI systems can be distinguished:

- Proximity ⁻ Individuals are able to form socials links.
- Quality ⁻ Individuals can evaluate their interactions among themselves and with the environment.
- Diversity ⁻ Diversity improves the capability of the system to react to unknown and unexpected situations.
- Stability ⁻ Individuals should not alter their local behavior in response to environmental fluctuations.
- Adaptability ⁻ the entire system should be able to adapt to environmental changes [20].

2.4.1.2 Genetic Algorithms

In the 1950s and the 1960s several computer scientists independently studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. The idea in all these systems was to evolve a population of candidate solutions to a given problem, using operators inspired by natural genetic variation and natural selection.

Genetic algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s. In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithms to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems.

GA is a method for moving from one population of "chromosomes" (e.g., strings of ones and zeros, or "bits") to a new population by using a kind of "natural selection" together with the genetics–inspired operators of crossover, mutation, and inversion. Each chromosome consists of "genes" (e.g., bits), each gene being an instance of a particular "allele" (e.g., 0 or 1) [25].

GA Operators

The simplest form of genetic algorithm involves three types of operators: selection, crossover and mutation.

Selection: This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce.

Crossover: This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. The crossover roughly mimics biological recombination between two single−chromosome organisms.

Mutation: This operator randomly flips some of the bits in a chromosome. Mutation can occur at each bit position in a string with some probability, usually very small.

2.4.1.3 Artificial immune system

The term Artificial Immune System (AIS) belongs to a terminology that refers to adaptive systems inspired by theoretical and experimental immunology with the goal of problem solving [23].

The primary goal of an artificial immune system (AIS) is to efficiently detect changes in the environment or deviations from the normal system behavior in complex problems domains. The role of the mammal immune system can be summarized as follows. It should protect the bodies from infections. The most interesting working behavior is the self-optimization and learning process. Two immune responses were identified. The primary one is to launch a response to invading pathogens leading to an unspecific response (using Leukocytes). In contrast, the secondary immune response remembers past encounters, i.e. it represents the immunologic memory. It allows a faster response the second time around showing a very specific response (using B-cells and T-cells) [26]. An AIS basically consists of three parts, which have to be worked out in the Immune engineering process [23] :

- Representations of the system components (i.e., the mapping of technical components to antigens and antibodies).
- Affinity measures (i.e., mechanisms to evaluate interactions [for example, stimulation pattern and fitness functions] and the matching of antigens and antibodies).

- Adaptation procedures to incorporate the system's dynamics (i.e., genetic election).

The scope of AIS is widespread. There are applications for fault and anomaly detection, data mining (machine learning, pattern recognition), agent based systems, control, and robotics.

One of the first AIS was presented in (Kephart 1994). Based on this work, misbehavior detection and attack or intrusion detection systems were developed based on the working principles of the natural immune system [26].

2.5 Swarm Intelligence Algorithms

With swarm intelligence, the developed algorithms need to be flexible to internal and external changes, to be robust when some individuals fail, to be decentralized and self-organized. we will address two of the most popular algorithms based on these concepts, including Particle Swarm Optimization (PSO) and Ant Colony System (ACS) algorithms.

2.5.1 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO)is an algorithm modeled on swarm intelligence, that finds a solution to an optimization problem in a search space or model and predict social behavior in the presence of objectives. The PSO is a stochastic, population-based computer algorithm modeled on swarm intelligence [24].

Particle swarm optimization (PSO) was first introduced by *James Kennedy* and *Russell C. Eberhart.* It is a relatively new stochastic optimization technique that can simulate the swarm behavior of birds flocking. In PSO, an individual in the swarm, called a particle, represents a potential solution. Each particle has a fitness value and a velocity, and it learns the experiences of the swarm to search for the global optima. Traditional PSO is shown in figure 2.3. These steps are described as follows [27].

(1) *Initialization*. We first decide how many particles used to solve the problem. Every particle has its own position, velocity and best solution. If we use M particles, their best solutions, and their velocities can be represented as:

$$\boldsymbol{X} = \{x_0, x_1, \dots, x_{M-1}\}, \qquad (2.1)$$

$$\boldsymbol{B} = \{b_0, b_1, \dots, b_{M-1}\}, \qquad (2.2)$$

$$\boldsymbol{V} = \{v_0, v_1, \dots, v_{M-1}\}. \qquad (2.3)$$

(2) *Velocity updating*. This step is shown in Eq. (4), where c_1 and c_2 are constants, r_1 and r_2 are random variables in the range from 0 to 1, $b_i(t)$ is the best solution of the i-th particle for the iteration number up to the t-th iteration.

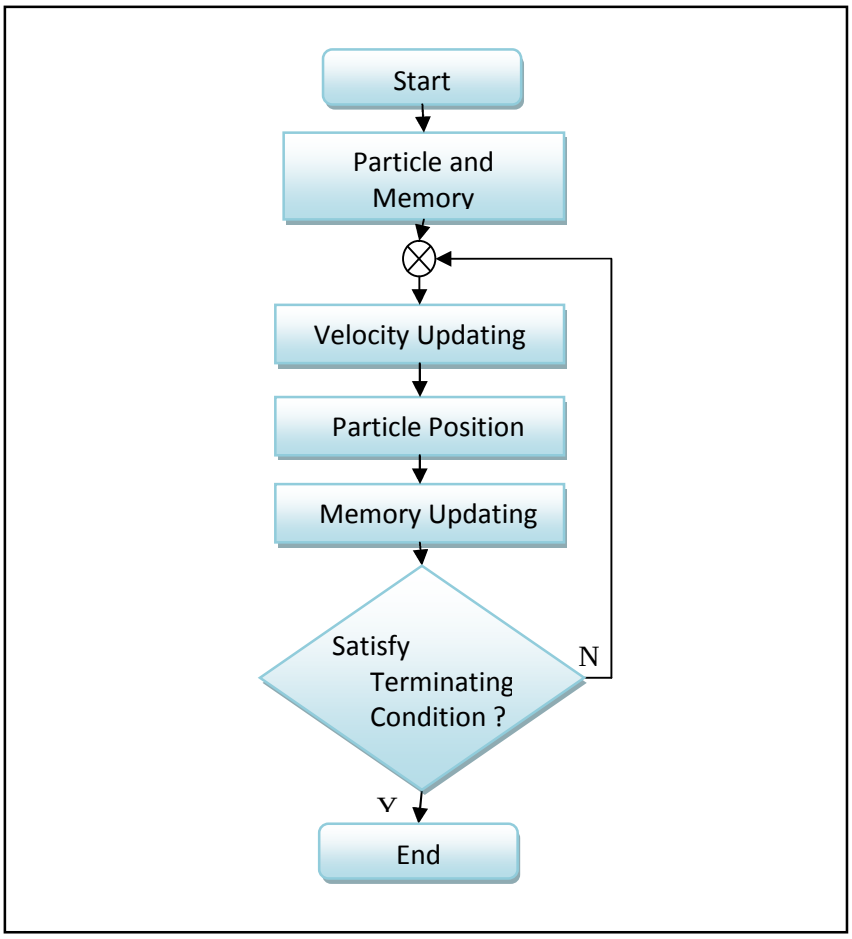

Figure 2.3 Procedures for particle swarm optimization

The G(t) is the best solution of all particles:

$$v_i(t+1) = v_i(t) + c_1.r_1.\big(b_i(t) - x_i(t)\big) + c_2.r_2.\big(G(t) - x_i(t)\big). \qquad (2.4)$$

To prevent the velocity from becoming too large, we set a maximum value to limit the range of velocity as $-V_{max} \leq V \leq V_{max}$.

(3) *Position updating*. which is processed by Eq. (5):

$$x_i(t+1) = x_i(t) + v_i(t), \quad i = 0, 1, \dots, M-1. \qquad (2.5)$$

(4) Memory updating. If we find a better solution than G (t) in G (t+1), G (t)

will be replaced by G (t+1). Otherwise, there will be no change for G (t).

(5) These recursive steps continue unless we reach the termination condition.

2.5.2 Ant Colony Optimization (ACO)

Ant colony optimization , whose first member, called Ant System (AS), was initially proposed by *Colorni*, *Dorigo* and *Maniezzo* [28] in the early 1990's. The development of these algorithms was inspired by the observation of ant colonies. Ants are social insects. They live in colonies and their behavior is governed by the goal of colony survival rather than being focused on the survival of individuals [29].

The behavior that provided the inspiration for ACO is the ants' foraging behavior, and in particular, how ants can find shortest paths between food sources and their nest. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants can smell pheromone. When choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source [29].

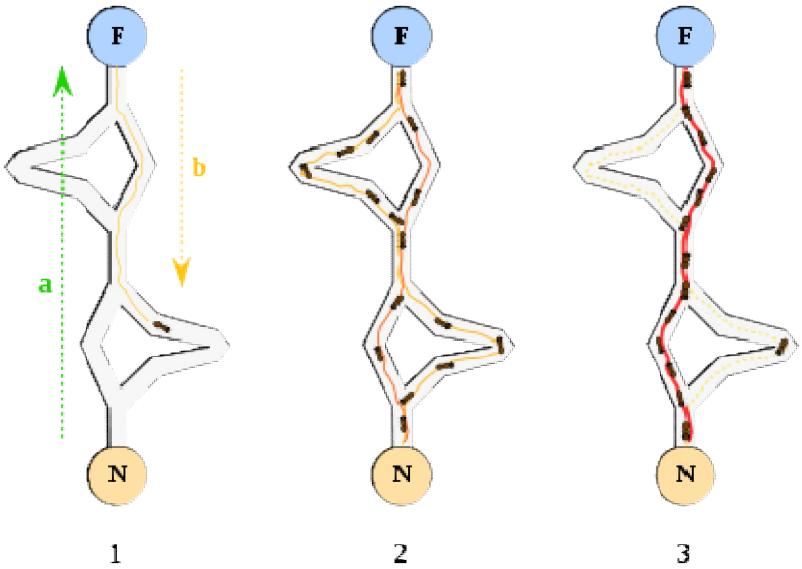

Figure 2.4 Illustrating the behavior of real ant movements.

In general the steps of an ant colony algorithm can be summarized as shown in figure 2.5.

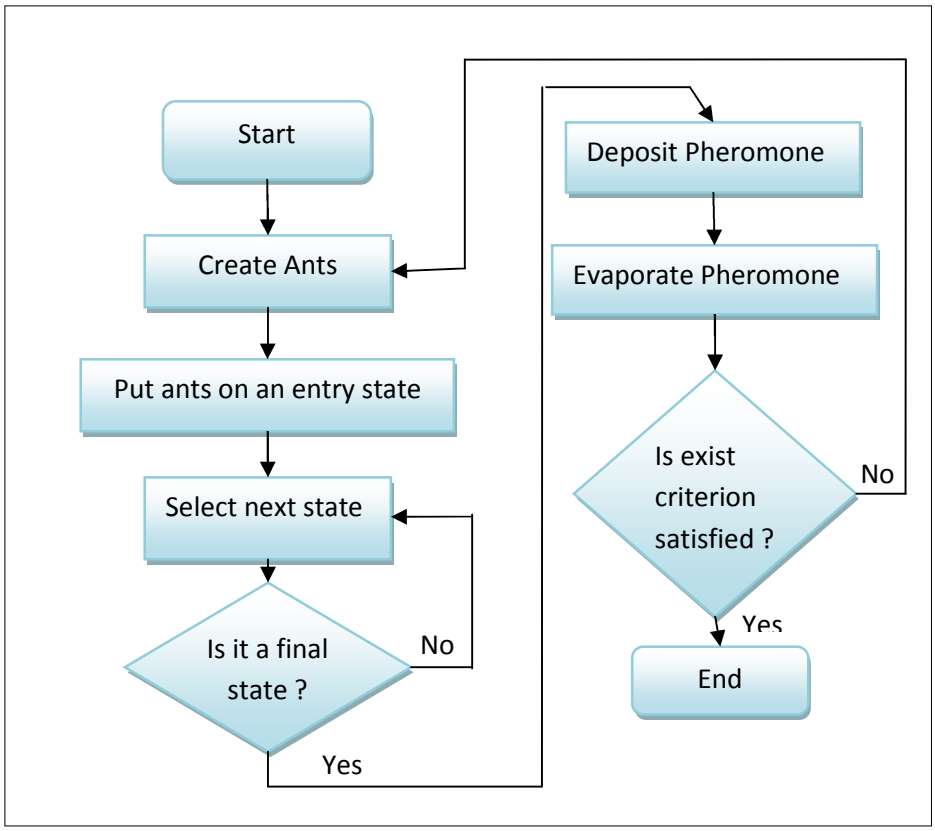

Figure 2.5 Ant Colony Algorithm Flowchart.

2.6 Self-Organization In Wsn Using Bio-Inspired Mechanism

Decades of research and vast implementation of wireless networks , has led it to grow tremendously, thereby creating performance issues and the need for manageability and scalability, …etc. Due to manageability, scaling and the need of better performance of network, it is important that the node is autonomous and work in a self-organizing manner. Researchers have proved, autonomous behavior not only helps in scalability and manageability but also helps in achieving global consensus using local information, cost efficient topology deployment and maintenance and the evolution of the network over time . Due to the self-organization behavior of the nodes these models were mostly decentralized and inspirations from nature were drawn.

In the next section we present the previews works that provide the self-organization of wireless sensor networks using bio-inspired mechanism. The researchers benefit from the advantages of biological systems to organize themselves and adapt to environmental changes depending on local state for assure a global behavior.

2.6.1 A Bio-Inspired Architecture for Division of Labour in SANETs (Sensor Ad-hoc Networks)

Division of labour is one of the possible strategies to efficiently exploit the resources of autonomous systems. It is also a phenomenon often observed in animal systems. In [30] an architecture that implements division of labour in Sensor/Actuator Networks was described. The architecture is based on probabilistic decisions. The architecture exploits the interactions between agents, but only within a limited range. The local interactions are however enough to induce division of labour at the global level, i.e. to provide a self-organizing behavior.

the application layer, which is in charge of selecting the tasks to perform. Agents have the capability to perceive whether their action is successful or not, either by directly sensing the environment, or by receiving a feedback from other nodes.

Before choosing a new task, the application layer adapts its parameter on the base of the outcome of the previous task. Robots and motes know a priori the list of possible tasks that they can perform. They have generally different sets of tasks T_{agent}. Each agent associates a task to a real number τ_i, with $i \in T_{agent}$. At the moment of selecting a task to perform, the agent chooses randomly between the tasks.

The probability to choose task *i* is:

$$P(i) = \frac{\tau_i^{\beta_{task}}}{\sum_{k \in T_{agent}} \tau_k^{\beta_{task}}}, \quad (2.6)$$

With $\beta_{task} \geq 1$. This equation is like the one used to model how ants choose one path among the several that bring to a food source. In the original formulation ,τ_i was the concentration of pheromone on path i. The parameter β was introduced to increase the exploitation of good paths.

The agent initializes $\tau_i = \tau_{init}$, $\forall i \in T_{agent}$. If the agent is successful in performing task *i*, then

$$\tau_i = \min\{\tau_{max}, \tau_i + \Delta\tau\} \quad (2.7)$$

And if it unsuccessful.

$$\tau_i = \max\{\tau_{min}, \tau_i - \Delta\tau\} \quad (2.8)$$

The agents in [30] have four tasks. For three of them, the behaviors of robots and sensors are the same. They suppose the agents are used to sense the environment and to report to a base host. The tasks are:

T1) measure the temperature locally and send it to the base;

T2) record the sound in the surroundings and send it to the base;

T3) record a video of the place and send it to the base.

Motes' and robots' behaviors are different in the case of the fourth task:

T4) motes broadcast help requests, robots answer to them and travel where they are needed.

When an agent want to rout a packet, it need to rout discovery process, the authors propose a probability P_{nd}^{i} for a node *i* to choose *n* as next hop to reach *d* .

In [30] the way of measuring the division of labour is to count how many nodes are involved in a task, and the spatial distribution of the tasks.

2.6.2 Ant based Self-organized Routing Protocol for Wireless Sensor Networks

multihop routing in WSN is affected by new nodes constantly entering/leaving the system. Therefore, biologically inspired algorithms are reviewed and enhanced to tackle problems arise in WSN. Ant routing has shown an excellent performance for sensor networks.

In [31] the design of an ant based autonomous routing algorithm for the sensor networks is presented. The proposed bio-inspired self-organized algorithm will also meet the enhanced sensor network requirements, including energy consumption, success rate and time.

The proposed self-organized system mainly based on routing section. The optimal route discovery is tackled by ant colony optimization. Routing decision will achieved through probabilistic decision rule that can be expressed mathematically by :

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha}.\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{\lambda \epsilon J_i^k}\left[\tau_{ij}(t)\right]^{\alpha}.\left[\eta_{ij}(t)\right]^{\beta}} \qquad (2.9)$$

Where

- $p_{ij}^{k}(t)$ overall desirability for ant k located in city i to choose to move to city j.
- T_{ij} is a value stored in a pheromone table.
- η_{ij} is an heuristic evaluation of edge (i,j).
- α and β control the relative weight.

The decision will depend on the used metrics as, packet receiving rate, velocity and remaining power mechanism as node *i* have (α^i, γ^i, β^i) respectively.

Initial result through this implementation is the pheromone table on each node, based on the latter the routing table was built .

2.6.3 Biologically Inspired, Cooperative Target Tracking Framework for WSNs

In [32] an energy-efficient Cooperative target tracking framework for Wireless Sensor Networks (WSN) was introduced. A Biological Independent Task Allocation (BITA) algorithm is presented to execute an application using a group of nodes. BITA is inspired from biological behaviors of differentiation in zygote formation.

When a zygote is formed, it comprises a collection of similar stem cells. Over time, the zygote cells start to specialize with different functionalities. This behavior is called differentiation . The same principle is be applied in the proposed system; the network nodes start equally in a default state and then exhibit some kind of differentiation to perform certain tasks according to their resource availability and location with respect to the target and other nodes. The nodes behave as if the target is a virtual chemical emitter. Their proximity to the target is used to influence the differentiation process.

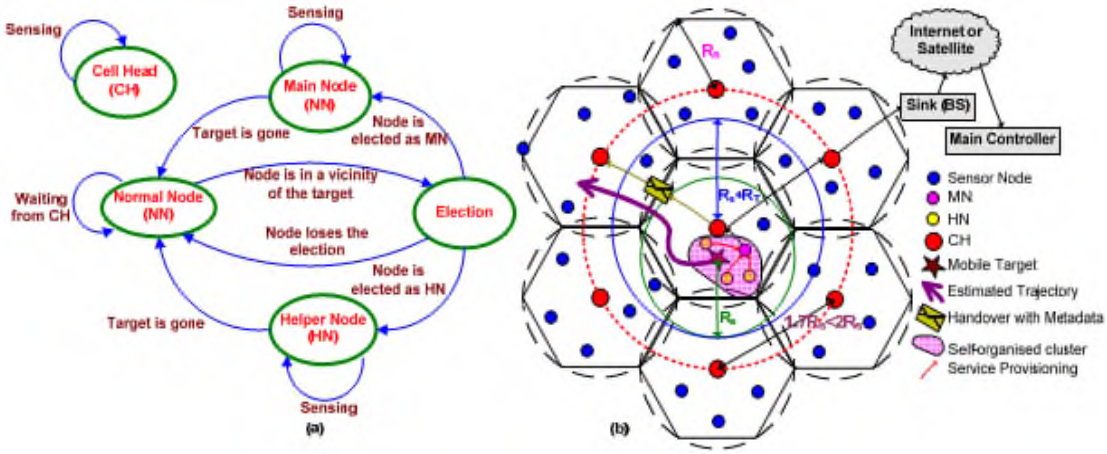

Figure 2.6 (a) Node State Transition Diagram, (b) System Architecture [32].

There are three system operational phases In [32], namely: group discovery, service provisioning and group management which base on the fitness functions proposed in each phase.

2.6.4 Ant Colony inspired Self-Optimized Routing Protocol based on Cross Layer Architecture for Wireless Sensor Networks

The authors in [33] propose a cross layer design based self-optimized (ACO) routing protocol for WSN ,link quality, energy level and velocity parameters are used to discover an optimal route. The signal strength, remaining power and timestamp metrics are trade in from physical layer to network layer.

Under BIOSARP, ACO is based on only two types of ant agents, which are, search ant (SA) and data ant (DA) agents. While data forwarding, the node first calls the DA. DA will select the optimal node based on the pheromone value stored in neighbour table. DA will move hop by hop on the base of pheromone values for neighbouring nodes until the destination. While selecting optimal node, if DA could not find the entry in the neighbour table, it will invoke SA as given in Figure 2.7. The SA will search for new nodes and calculates their pheromone value through the probabilistic rule.

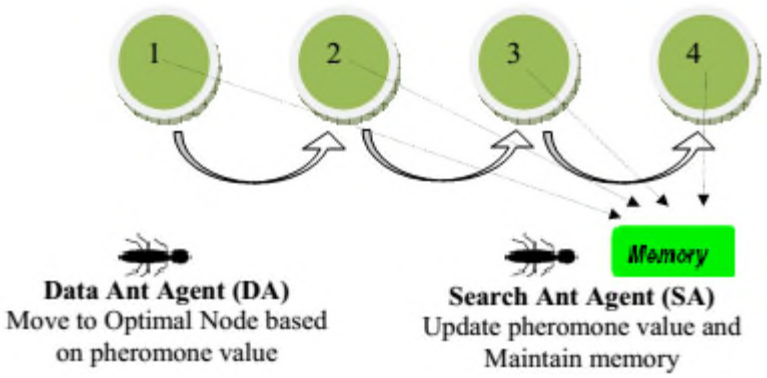

Figure 2.7 BIOSARP ACO mechanism based on Data Ant (DA) and Search Ant (SA) agents [33].

The maximum packet velocity (V) between a pair of nodes is calculated by :

$$V = \frac{d(S,N)}{delay(S,N)} \tag{2.10}$$

where d(S,N) is the one-hop distance between source node S and destination node N.
A second heuristic value ω_{ij} was added in the probability function to consider the link quality of the neighboring nodes while making decision.

The probabilistic decision rule is expressed mathematically via the following Equation:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}(t)]^{\beta}.[\omega_{ij}(t)]^{\vartheta}}{\sum_{h\epsilon J_i^k}[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}(t)]^{\beta}.[\omega_{ij}(t)]^{\vartheta}} \qquad (2.11)$$

- $p_{ij}^{k}(t)$ overall desirability for ant k located in city i to choose to move to city j.
- τ_{ij} is a value of pheromone depends on the delay parameter.
- η_{ij} is an heuristic evaluation of edge (i,j).
- ω_{ij} is the 2nd heuristic evaluation of edge (i,j).
- α, β and ϑ are three parameters that control the relative weight of pheromone trail and heuristic values.

V / V_m calculate the value of τ_{ij} , V_{batt}/ V_{mbatt} calculates the value of η_{ij} and value of ω_{ij} is obtained by Packet Receiving Rate. Where V_m is the maximum velocity of the Radio Frequency signal that is equal to the speed of light. V_{batt} is the battery voltage expressed previously in [33]. V_{mbatt} is the maximum battery voltage for sensor nodes and is equal to 3.6 volts.

The proposed system uses interaction between the physical layer and the network layer in order to select the best next node as shown in figure 2.8.

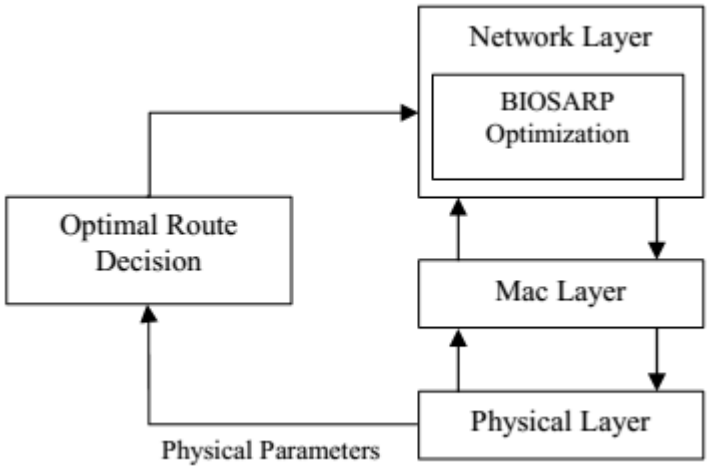

Figure 2.8 Cross layer architecture [33].

2.7 Conclusion

Applying Bio-inspired algorithms in WSN become the solution toward a self-organizing network. With this features such as positive feedback that enhance the local process leading to a global efficiency of the system, the interaction between individuals have a role of updating the local information therefore a global vision of the network will be take. The use of probability in this system organize the decision and more then that make possible of converging to the optimal.

As we shown , a variety of bio-inspired algorithms can be applying for gain some desired features depending on the global goal of the hole network. Swarm Intelligence algorithms Such as ACA(ant colony Algorithm) agree in its steps with the concept of self-organization.

The ACO algorithm mentioned before describe its application on the TSP which is the first problem designed for. The next chapter will describe in details the concept of the algorithm from real to artificial , its main variants in sensor network applications.

Chapter III

Ant Colony Algorithms

3.1 Introduction

The field of «ant algorithms» studies models derived from the observation of real ants' behavior, and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems. The main idea is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve computational problems. Several different aspects of the behavior of ant colonies have inspired different kinds of ant algorithms. Examples are foraging, division of labor and cooperative transport. In all these examples, ants coordinate their activities via stigmergy, a form of indirect communication mediated by modifications of the environment. For example, a foraging ant deposits a chemical on the ground which increases the probability that other ants will follow the same path. Biologists have shown that many colony-level behaviors observed in social insects can be explained via rather simple models in which only stigmergic communication is present. In other words, biologists have shown that it is often sufficient to consider stigmergic, indirect communication to explain how social insects can achieve self-organization.

3.2 Toward Artificial Ants

An important insight of real ants' behavior was that most of the communication among individuals, or between individuals and the environment, is based on the use of chemicals produced by the ants. These chemicals are called *pheromones*.

Trail pheromone is a specific type of pheromone that some ant species use for marking paths on the ground, for example, paths from food sources to the nest. By sensing pheromone trails foragers can follow the path to food discovered by other ants [34].

The behavior of real ants, may be used as an inspiration to design artificial ants that solve optimization problems a similar way. Stigmergic communication happens via the pheromone that ants deposit on the ground. Analogously, artificial ants may simulate pheromone laying by modifying appropriate pheromone variables associated with problem states they visit while building solutions to the optimization problem[35].

Another important aspect of real ants' foraging behavior that may be exploited by artificial ants is the coupling between the *autocatalytic* mechanism and the *implicit*

evaluation of solutions. By implicit solution evaluation, we mean the fact that shorter paths (which correspond to lower cost solutions in the case of artificial ants) are completed earlier than longer ones, and therefore they receive pheromone reinforcement quicker. Implicit solution evaluation coupled with autocatalysis can be very effective: the shorter the path, the sooner the pheromone is deposited, and the more ants use the shorter path.

Both real and artificial ant colonies are composed of a population of individuals that work together to achieve a certain goal. A colony is a population of simple, independent, asynchronous agents that cooperate to find a good solution to the problem at hand. In the case of real ants, the problem is to find the food, while in the case of artificial ants, it is to find a good solution to a given optimization problem.

In ACO, the artificial pheromone trails are the sole means of communication among the ants. A mechanism analogous to the evaporation of the physical pheromone in real ant colonies allows the artificial ants to forget the history and focus on new promising search directions.

There are however some important differences between real and artificial ants:

- Artificial ants live in a discrete world| they move sequentially through a finite set of problem states.
- The pheromone update (i.e., pheromone depositing and evaporation) is not accomplished in exactly the same way by artificial ants as by real ones. Sometimes the pheromone update is done only by some of the artificial ants, and often only after a solution has been constructed.
- Some implementations of artificial ants use additional mechanisms that do not exist in the case of real ants. Examples include look-ahead, local search, backtracking, etc [35].

3.3 Artificial Ants and Minimum Cost Paths

Consider a connected graph G=(N,A) where N is the set of nodes and A is the set of undirected arcs connecting them. The two points between which we want to establish a minimum cost path are called source and destination nodes, as typically done in the literature on minimum cost path problems (when the cost of arcs is given by their length, the minimum cost path problem is the same as the shortest-path problem); sometimes, in analogy to the shortest-path–finding behavior of real ants, it will also called nest and food source.

A simple-ACO algorithm is presented in[34], which adapts the real ants' behavior to the solution of minimum cost path problems on graphs. To each arc (i,j) of the graph G=(N,A) it associate a variable τ_{ij} called artificial pheromone trail, shortened to pheromone trail in the following. Pheromone trails are read and written by the ants. The amount (intensity) of a pheromone trail is proportional to the utility, as estimated by the ants, of using that arc to build good solutions.

3.3.1 Ants' Path-Searching Behavior

Each ant builds, starting from the source node, a solution to the problem by applying a step-by-step decision policy. At each node, local information stored on the node itself or on its outgoing arcs is read (sensed) by the ant and used in a stochastic way to decide which node to move to next. At the beginning of the search process, a constant amount of pheromone (e.g., $\tau_{ij} = 1, \forall (i,j) \in A$))is assigned to all the arcs. When located at a node *i* an ant *k* uses the pheromone trails τ_{ij} to compute the probability of choosing *j* as next node :

$$P_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha}}, & if\ j \in N_i^k; \\ 0, & if\ j \notin N_i^k \end{cases} \tag{3.1}$$

Where N_i^k is the neighborhood of ant k when in node *i*. In S-ACO the neighborhood of a node i contains all the nodes directly connected to node *i* in the graph G=(N,A) except for the predecessor of node *i* (i.e., the last node the ant visited before moving to *i*). In this way the ants avoid returning to the same node they visited immediately before node *i* [34].

3.3.2 Path Retracing and Pheromone Update

When reaching the destination node, the ant switches from the *forward* mode to the *backward* mode and then retraces step by step the same path backward to the source node. An additional feature is that, before starting the return trip, an ant eliminates the loops it has built while searching for its destination node. The problem of loops is that they may receive pheromone several times when an ant retraces its path backward to deposit pheromone trail, leading to the problem of self-reinforcing loops.

Loop elimination can be done by iteratively scanning the node identifiers position by position starting from the source node: for the node at the *i*-th position, the path is scanned starting from the destination node until the first occurrence of the node is encountered, say, at position *j* (it always holds that $i \leq j$ because the scanning process stops at position *i* at the latest). If we have $j > i$, the sub path from position $i + 1$ to position j corresponds to a loop and can be eliminated.

During its return travel to the source the *k*-th ant deposits an amount $\Delta\tau^k$ of pheromone on arcs it has visited. In particular, if ant *k* is in the backward mode and it traverses the arc (i, j), it changes the pheromone value τ_{ij} as follows:

$$\tau_{ij} = \tau_{ij} + \Delta\tau^k , \qquad (3.2)$$

By this rule an ant using the arc connecting node *i* to node *j* increases the probability that forthcoming ants will use the same arc in the future.

An important aspect is the choice of $\Delta\tau^k$. In the simplest case, this can be the same constant value for all the ants. In this case, only the *differential path length* works in favor of the detection of short paths: ants which have detected a shorter path can deposit pheromone earlier than ants traveling on a longer path. In addition to the deterministic backward pheromone trail update, the ants may also deposit an amount of pheromone trail which is a function of the path length, the shorter the path the more pheromone is deposited by an ant.

3.3.3 Pheromone Trail Evaporation

Pheromone trail evaporation can be seen as an exploration mechanism that avoids quick convergence of all the ants toward a suboptimal path. In fact, the decrease in pheromone intensity favors the exploration of different paths during the whole search process. In real ant colonies, pheromone trails also evaporate, but, as we have seen, evaporation does not

play an important role in real ants' shortest-path finding. The fact that, on the contrary, pheromone evaporation seems to be important in artificial ants is probably due to the fact that the optimization problems tackled by artificial ants are much more complex than those real ants can solve. A mechanism like evaporation that, by favoring the forgetting of errors or of poor choices done in the past, allows a continuous improvement of the ''learned'' problem structure seems therefore to be necessary for artificial ants. pheromone trails are evaporated by applying the following equation to all the arcs:

$$\tau_{ij} = (1-\rho)\tau_{ij}\,, \quad \forall\,(i,j) \in A, \tag{3.3}$$

Where $\rho \in [0,1]$ is a parameter. After pheromone evaporation has been applied to all arcs, the amount of pheromone $\Delta\tau^k$ is added to the arcs [34].

3.4 Main Variants of ACO

From the appearance of ant system many Ant colony optimization algorithms extend in their implementation to the modification of the pheromone update process, we will present the more successful ones.

3.4.1 Ant System (AS)

Ant System was the first ACO algorithm to be proposed in the literature. Its main characteristic is that the pheromone values are updated by all the ants that have completed the tour [35]. The two main phases of the AS algorithm constitute the ants' solution construction and the pheromone update will be described below.

Tour Construction

In AS, m (artificial) ants concurrently build a tour of the Travelling Salesman Problem (TSP). Initially, ants are put on randomly chosen cities. At each construction step, ant k applies a probabilistic action choice rule, called *random proportional rule*, to decide which city to visit next. In particular, the probability with which ant *k*, currently at city *i*, chooses to go to city *j* is [34]:

$$P_{ij}^k = \frac{[\tau_{ij}]^\alpha\,[\eta_{ij}]^\beta}{\sum_{l\in N_i^k}[\tau_{il}]^\alpha\,[\eta_{il}]^\beta}\,, if\ j \in N_i^k; \tag{3.4}$$

where $\eta_{ij} = 1/d_{ij}$ is a heuristic value that is available a priori, α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information, and N_i^k is the feasible neighborhood of ant k when being at city i, that is, the set of cities that ant k has not visited yet (the probability of choosing a city outside N_i^k is 0).

Each ant k maintains a memory M^k which contains the cities already visited, in the order they were visited. This memory is used to define the feasible neighborhood N_i^k in the construction rule given by equation (3.4). In addition, the memory M^k allows ant k both to compute the length of the tour T^k it generated and to retrace the path to deposit pheromone.

Concerning solution construction, there are two different ways of implementing it:
parallel and sequential solution construction. In the parallel implementation, at each construction step all the ants move from their current city to the next one, while in the sequential implementation an ant builds a complete tour before the next one starts to build another one. In the AS case, both choices for the implementation of the tour construction are equivalent in the sense that they do not significantly influence the algorithm's behavior.

Update of Pheromone Trails

After all the ants have constructed their tours, the pheromone trails are updated. This is done by first lowering the pheromone value on all arcs by a constant factor, and then adding pheromone on the arcs the ants have crossed in their tours. Pheromone evaporation is implemented by :

$$\tau_{ij} = (1-\rho)\tau_{ij}\,, \quad \forall\,(i,j) \in L\,; \tag{3.5}$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails and it enables the algorithm to ''forget'' bad decisions previously taken. In fact, if an arc is not chosen by the ants, its associated pheromone value decreases exponentially in the number of iterations. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta\,\tau_{ij}^{k}\,, \quad \forall\,(i,j) \in L\,; \tag{3.6}$$

Where $\Delta\ \tau_{ij}^{k}$ is the amount of pheromone ant *k* deposits on the arcs it has visited. It is defined as follows:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C^k}, & if\ (i,j) \in\ T^k\ ; \\ 0, & otherwise\ \ ; \end{cases} \tag{3.7}$$

Where C^k, the length of the tour T^k built by the *k*-th ant, is computed as the sum of the lengths of the arcs belonging to T^k [34].

3.4.2 Ant Colony System (ACS)

ACS (Dorigo & Gambardella, 1997a,b) differs from AS in three main points. First, it exploits the search experience accumulated by the ants more strongly than AS does through the use of a more aggressive action choice rule. Second, pheromone evaporation and pheromone deposit take place only on the arcs belonging to the best-so-far tour. Third, each time an ant uses an arc (*i*,j) to move from city *i* to city *j* it removes some pheromone from the arc to increase the exploration of alternative paths.

Tour Construction

In ACS, when located at city *i*, ant *k* moves to a city *j* chosen according to the so called *pseudorandom proportional rule*, given by

$$j = \begin{cases} argmax_{l\in N_i^k}\{\tau_{il}[\eta_{il}]^{\beta}\}, & if\ q \le q_0\ ; \\ J\ , & otherwise\ \ equation\ (3.4) is\ used\ ; \end{cases} \tag{3.8}$$

where q is a random variable uniformly distributed in [0,1], $q_0 (0 \le q_0 \le 1)$ is a parameter [34].

Local pheromone Update

The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the last edge traversed [35]:

$$\tau_{ij} = (1-\rho)\ \tau_{ij} + \rho\ \tau_0 \tag{3.9}$$

Global Pheromone Trail Update

In ACS only one ant (the best-so-far ant) is allowed to add pheromone after each iteration. Thus, the update in ACS is implemented by the following equation [34]:

$$\tau_{ij} = (1-\rho)\,\tau_{ij} + \rho\Delta\tau_{ij}^{bs}\,, \forall\,(i,j) \in T^{bs}\,; \tag{3.10}$$

It is important to note that in ACS the pheromone trail update, both evaporation and new pheromone deposit, only applies to the arcs of T^{bs}, not to all the arcs as in AS. This is important, because in this way the computational complexity of the pheromone update at each iteration is reduced from $O(n^2) to\ O(n)$; where n is the size of the instance being solved [34].

3.4.3 $\mathcal{MAX}$ -$\mathcal{MIN}$ Ant System

MAX-MIN Ant System is an improvement over the original AS idea. MMAS was proposed by StÄutzle and Hoos , who introduced a number of changes of which the most important are the following [35]:

- only the best ant can update the pheromone trails, and
- the minimum and maximum values of the pheromone are limited.

Eq. (3.6) takes the following new form:

$$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}^{best} \tag{3.11}$$

Where $\Delta\tau_{ij}^{best}$ is the pheromone update value defined by :

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}}\,, & if\ the\ best\ ant\ used\ edge\ (i,j)\ in\ its\ tour; \\ 0, & otherwise\,; \end{cases} \tag{3.12}$$

L_{best} is the length of the tour of the best ant.

Concerning the limits on the minimal and maximal pheromone values allowed, respectively $\tau_{min}$and τ_{max}, StÄutzle and Hoos suggest that they should be chosen experimentally based on the problem at hand.

The maximum value τ_{max} may be calculated analytically provided that the optimum ant tour length is known. In the case of the TSP, τ_{max} is given by :

$$\tau_{max} = \frac{1}{\rho} . \frac{1}{L^*} \tag{3.13}$$

where L^* is the length of the optimal tour. If L^* is not known, it can be approximated by L_{best}. The minimum pheromone value τ_{min} should be chosen with caution as it has a rather strong influence on the algorithm performance. StÄutzle and Hoos present an analytical approach to finding this value based on the probability p_{best} that an ant constructs the best tour found so far. This is done as follows. First, it is assumed that at each construction step an ant has a constant number *k* of options available. Therefore, the probability that an ant makes the right decision (i.e.,the decision that belongs to the sequence of decisions leading to the construction of the best tour found so far) at each of *n* steps is given by $p_{dec} = \sqrt[n-1]{p_{best}}$. The analytical formula they suggest for finding τ_{min} is [35]:

$$\tau_{min} = \frac{\tau_{max} . (1 - p_{dec})}{k . p_{dec}} \tag{3.14}$$

The process of pheromone update in MMAS is concluded by verifying that all pheromone values are within the imposed limits :

$$\tau_{ij} = \begin{cases} \tau_{min} , & if \ \tau_{ij} < \tau_{min} \\ \tau_{ij} \ , & if \ \tau_{min} \leq \tau_{ij} \leq \tau_{max} \\ \tau_{max} , & if \ \tau_{ij} > \tau_{max} \end{cases} \tag{3.15}$$

3.5 Our proposed Algorithm

Finding the optimal path in wireless sensor routing was one of the most interesting process in the research field, the idea in the new related works is how adapt the design of the path search algorithm to the requirement of WSN. The need of wireless sensor network routing that conserve in a way or another the lifetime, reliability, coverage, connectivity …etc, is the main goal behind any routing algorithm.

Some researchers have exploited the application of the ACA to meet the requirements of protocol design in sensor networks. The application of the ACA provides a better self-organizing approach in optimization of routing, Applying the ACA in routing is inspired by the intuitive matching of the ACA's solution searching process and the routing path searching process. With the attractiveness parameters of the Moving Decision set to the application's requirement, each agent in the ant-like routing explores the optimal path to the destination node, based on the pheromone trail.

3.5.1 The proposed ant routing algorithm

Our algorithm maintain the connectivity and the lifetime of th network ,Since the self-organization system are autonomous , the algorithm should be run at sensor nodes level not the sink, so the nodes that apply the algorithm which are represented by ants search the optimal path to the BS depending on the more connected path, with less energy consumption. The details of the algorithm will be present in the following:

The moving decision

The ant move from a node to another with a probability *p* as defined in the ACA:

$$P_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}\,[\eta_{ij}]^{\beta}}{\sum_{l\notin tabu^{k}}[\tau_{il}]^{\alpha}\,[\eta_{il}]^{\beta}} & ,if\ j \notin\ tabu^{k}; \\ 0, \qquad otherwise; \end{cases} \tag{3.16}$$

The ant choose with probability p the node to move to it providing that this node does not belong to the *tabu list* of this ant. The *tabu list* of an ant include the list of visited neighbors nodes. it means that an ant at node *i* does not choose to move to one of the neighbors nodes if it is visited before by the same ant.

Where:

τ_{ij}: is the trail level between node *i* and *j* , constant at the beginning of the algorithm.

η_{ij}: is the attractiveness of the move .

α , β: represent the impact of the trail level and the attractiveness respectively.

In our algorithm:

- The trail level τ between all existing edge is set initially to a constant as all the ACA.
- Where η the attractiveness of the move depend on three parameters (distance, residual energy of the next node and the connectivity level(number of neighbors)).

Trail update

The ant in its *forward state* discover the path from the source node until it reach the destination (the sink in our algorithm). After that , it turn to the *backward state* to retrace the path and updating pheromone of the edges forming that path. the ants update the pheromone with respect to ρ the evaporation parameter. Means the pheromone depend on evaporation rate and the deposited quantity.

Specification

a) *The attractiveness*:

$$\eta_{ij} = \frac{E_r(j).C(j)}{d_{ij}} \qquad (3.17)$$

Where

$E_r(j)$: the residual energy of node j.

$C(j)$: the connectivity level (number of neighbors) of node j.

d_{ij}: is the distance between node i and j.

As the initial trail level of the edges are constant, the first decision is based on the attractiveness of the move, the ant choose the node with the more residual energy , since we do not prefer the node with less energy for conserving its life. The node with high

connectivity level is also preferred because it provide as the multi-choice path and overload at nodes will be reduce, the nodes with less neighbors may be die quickly if they will be chosen.

b) *Trail level*:

$$\tau_{ij}(t) = (1-\rho).\tau_{ij}(t-1) + \Delta\,\tau_{ij} \qquad (1.18)$$

Where

ρ: is the evaporation parameter, $0 \leq \rho \leq 1$.

$\tau_{ij}(t-1)$: the trail level of edge(i,j) at time (t-1).

$\Delta\,\tau_{ij}$: the value of th deposited pheromone on edge (i,j).

We define The quantity of the pheromone deposited by ant k by:

$$\Delta\,\tau_{ij}^{k} = \frac{1}{L_k + (E_c(i) + E_c(j))} \qquad (1.19)$$

where

L_k : is the tour length founded by ant k.

$E_c(i)$, $E_c(j)$: is the energy consumed by node *i* and *j* respectively.

The amount of pheromone deposited by ant k is a tread off between the length of the discovered path and the energy consumed in this edge(*i*,*j*) , signify that if the ant was discover a long path and a lot of energy was consumed at node *i* and j (an overload at node *i* and j in the past) it deposit less pheromone contrary it deposit more ,so the ant reinforce the good edges and therefore the good paths.

Terminating condition

Since our algorithm is a sequential ACA, the ants discover its path from the source node to the destination one by one, forming by that an iteration. The algorithm will be run until satisfy condition , this condition may be the number of iteration, a parameter related to network simulation like the number of alive node, the time to first dead node, the best path length…etc.

3.6 Conclusion

The problem of applications in sensor networks such as routing, should be defined first to transform the objectives of protocol to the goals of the ACA. Then the entities of the protocol and their action need to be mapped to the agents of the ACA. The core in the design is implementing the controlling mechanisms: specifying the moving decision and trail update. The weight parameters should be adjusted deliberately in different applications. The trail update method should also be considered: to adopt the local, global or the hybrid of the two.

As in our ACA the main goal is to maintain connectivity and the life time of the network, the number of ants depend on the number of sensor nodes , each ant is associated to a node. Sequentially, the ants in its forward state move from node to node depending on the more connected path taking in to account the residual energy until they reach the sink. Updating in the backward state the path by depositing pheromone.

In the next chapter, we will present the simulation of our algorithm , the language and environment where it was implemented. The result of the simulation will be discussed in details.

Chapter IV

Simulation and Results Analysis

4.1 Introduction

In this chapter we try to simulate the conception of the proposed algorithm, we use the C++ Builder programming environment to realize our WSN simulator and implement the proposed ant colony algorithm , in order to prolong the network lifetime .

The goal of our algorithm is to find the optimal path from any node in the network to the Sink, depending on the Energy of each node and its connectivity. The algorithm tries in each network state to find the optimal path without affecting the weak nodes, and try to reinforce the strongest paths according to the pheromone deposit.

We will introduce first a brief presentation about the C++ Builder, the goal of simulation, then the description of the environment, from the setting sensor network field to the ant colony algorithm parameters. Finally we discuss the results obtained by the application of the proposed algorithm in the performance of the network and suggest some future works concerning the ant colony algorithms applied in the field of WSNs.

4.2 About C++ Builder environment

We use in our application the Embarcadero C++ Builder 2010 which is one of the leading, integrated RAD C/C++ development environments for generating native applications under Microsoft Windows. This unique development environment combines the considerable flexibility of a Rapid Application Development (RAD) environment with the efficiency of the ISO-standardized programming languages C and C++.

The application area of C++Builder 2010 ranges from fast prototyping to large-scale applications across all economic sectors, regardless of client or server programs. The supported target platforms range from Microsoft Windows 2000 operating system through the current version of Microsoft Windows 7 operating system.

C++ is a programming language, which, with the aid of a compiler, is translated into native machine code. This makes it possible to generate highly efficient code. While C remains rather restricted to system programming, C++ has been developed to become a universal language with a stronger, more static typification based on C, and directly supports multiple programming styles. Today, C++ combines the object-orientated with the procedural, the abstract and the generic programming. It is particularly the generic programming that enables a high degree of flexibility. In doing so, the programmer has

the choice, and styles can be combined at will. As a conscious effort has been made to reject platform-specific properties, C++ is not just as fast as C, it is just as easy to port.

The Development Environment

The individual parts of the program are linked to each other within the development environment. When you launch C++Builder, it appears in the default layout. By contrast with the older versions 5 and 6, the individual parts are combined (docked) in one main window.

The following figure shows the default layout of the development environment.

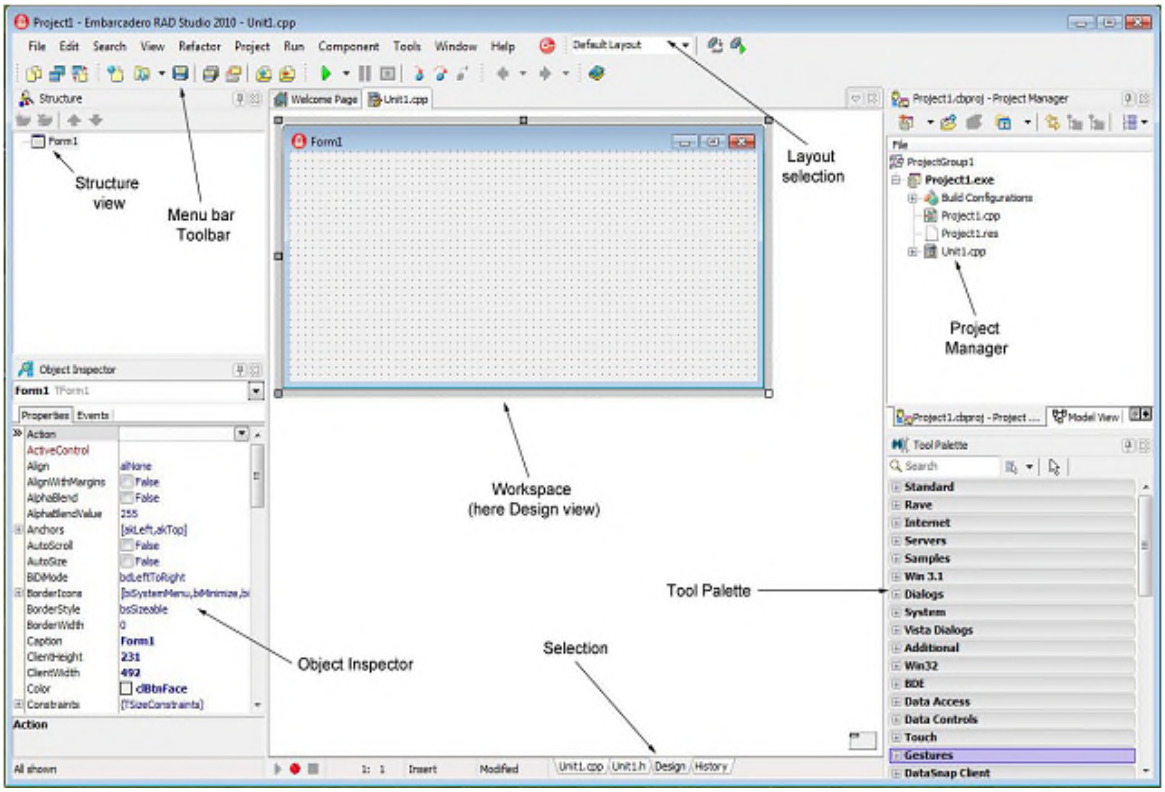

Figure 4.1 Development environment in the default layout

4.3 The goal of simulation

The goal of our application is to establish a WSN based on the ant colony Algorithm .For a reason to simulate the behavior of an ant system for searching an optimal path from the source node to the Sink , we choose the Sink as a final node since we simulate an event driven network where the detected event information is routed to the BS.

The ant colony routing algorithm proposed in the simulation take its decision based first on the more connected path for giving the packets the variety of taking one from different paths and therefore the chance of taking the optimal path , the second decision is based on the residual energy of the choosing nodes to pass through it, means the more Residual Energy the more the node is preferred.

The success rate of the algorithm is based on the Ant System Parameters : the pheromone rate, the attractiveness rate and the evaporation rate. The good adjustment of the two first rates make the algorithm trapped in the optimal path , the last (evaporation) rate has two advantages : first, it decrease the big accumulation values of the pheromone deposited in edges (paths), then its correct the past of the algorithm if it was made a wrong decision by reinforcing the bed paths.

4.4 Simulation Steps

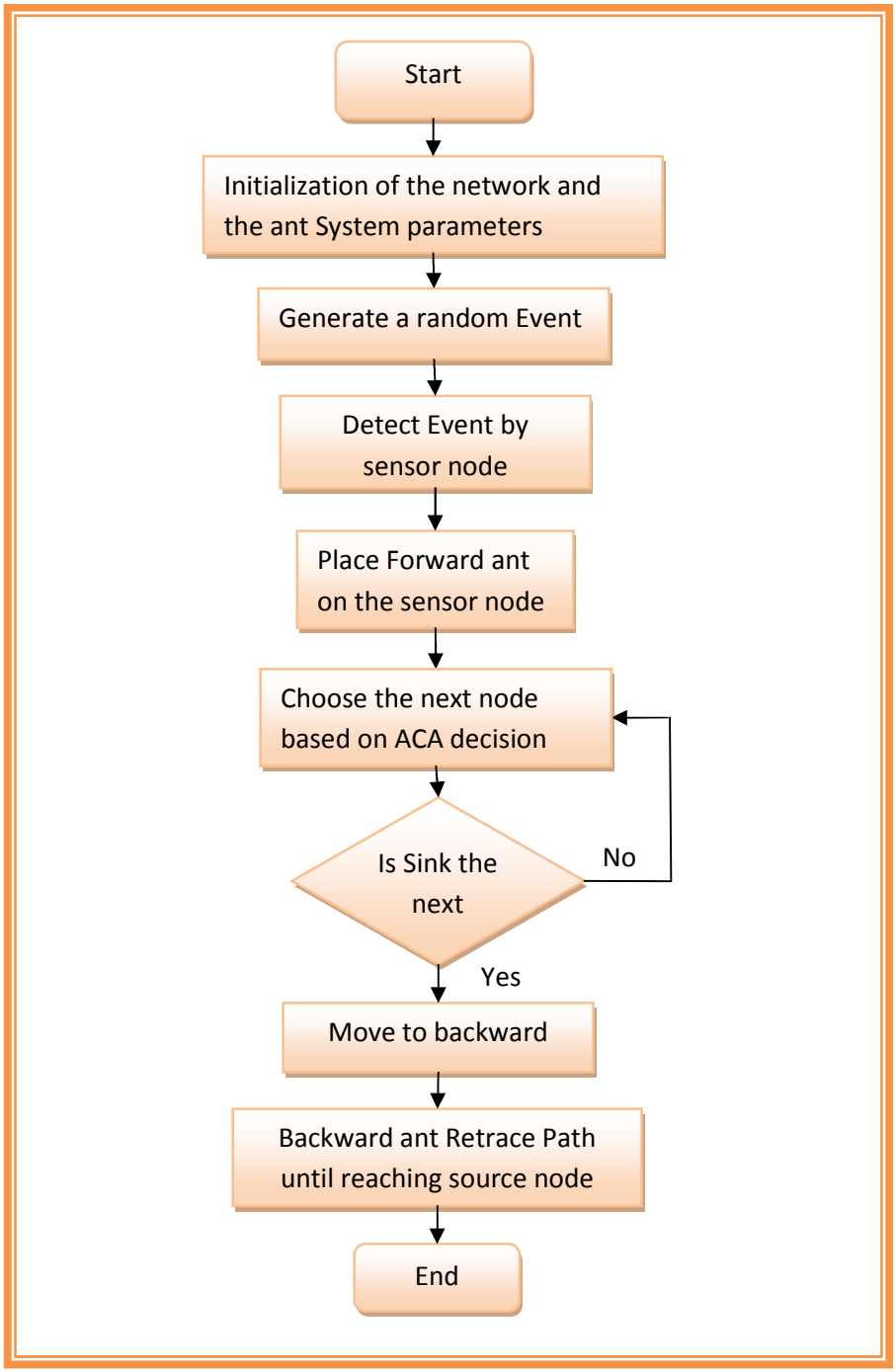

Figure 4.2 The simulated Algorithm Flowchart

4.5 Description of simulation

We run first the WSN_ACO application as shown in figure 4.3.

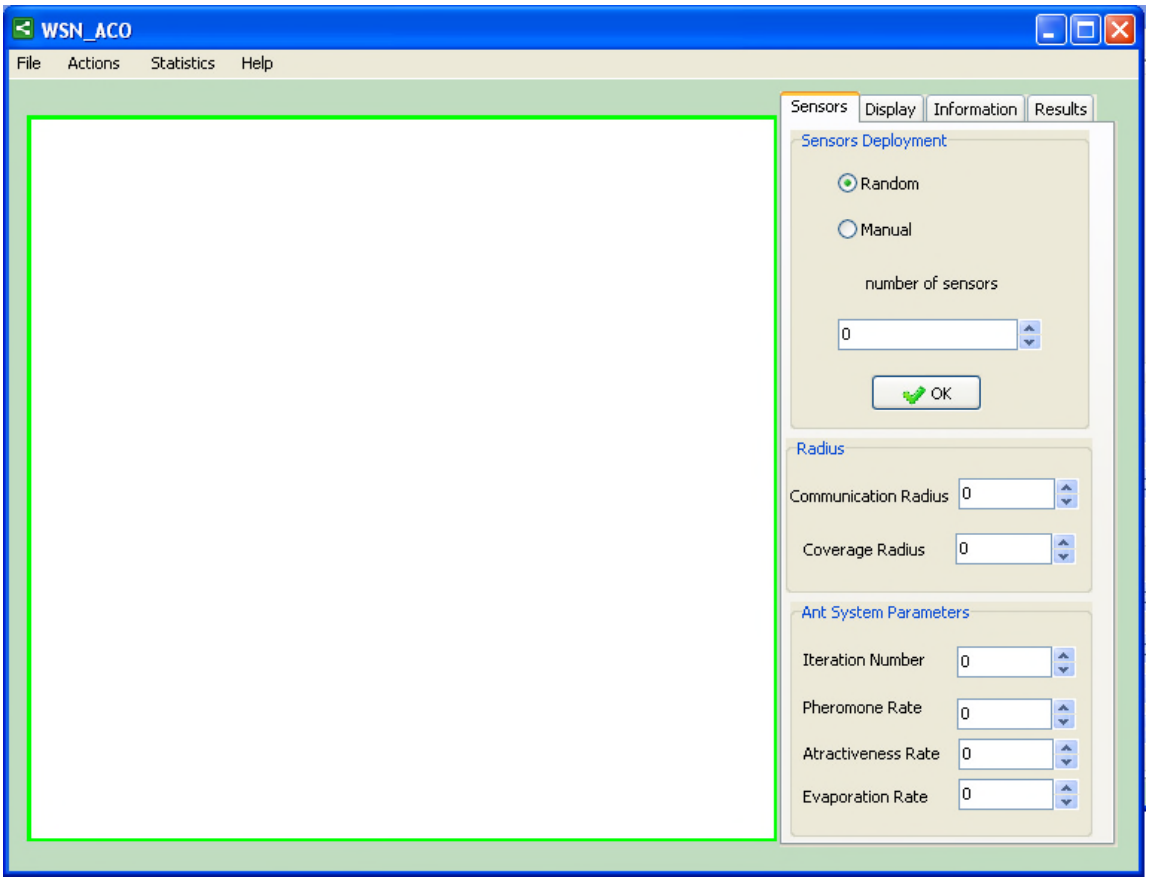

Figure 4.3 Simulation interface.

4.5.1 Network Setting parameters

In the tab sheet «Sensors», network setting is divided into two parts as follow:

Sensors Deployment

The deployment of the sensors in the network can be expressed in random or manual, respecting:

Number of sensors: determine the maximum number of sensors deployed in the field.

Communication Radius: represent the maximum range of communication between sensors nodes.

Coverage Radius: represent the maximum range of detecting an event by a sensor.

Ant System parameters

We use in the proposed ant colony algorithm the main significant parameters:

Number of iterations: define the number of times the algorithm will be run and as a terminating condition.

Pheromone Rate: the coefficient of pheromone defines the importance of the pheromone trail level in the probability calculation.

Attractiveness Rate: represent the attractiveness importance in the move decision.

Evaporation Rate: defines the percentage of the pheromone evaporation.

4.5.2 Displaying parameters

«Display» tab sheet contain the possible displaying information about the network topology as shown in figure 4.4.

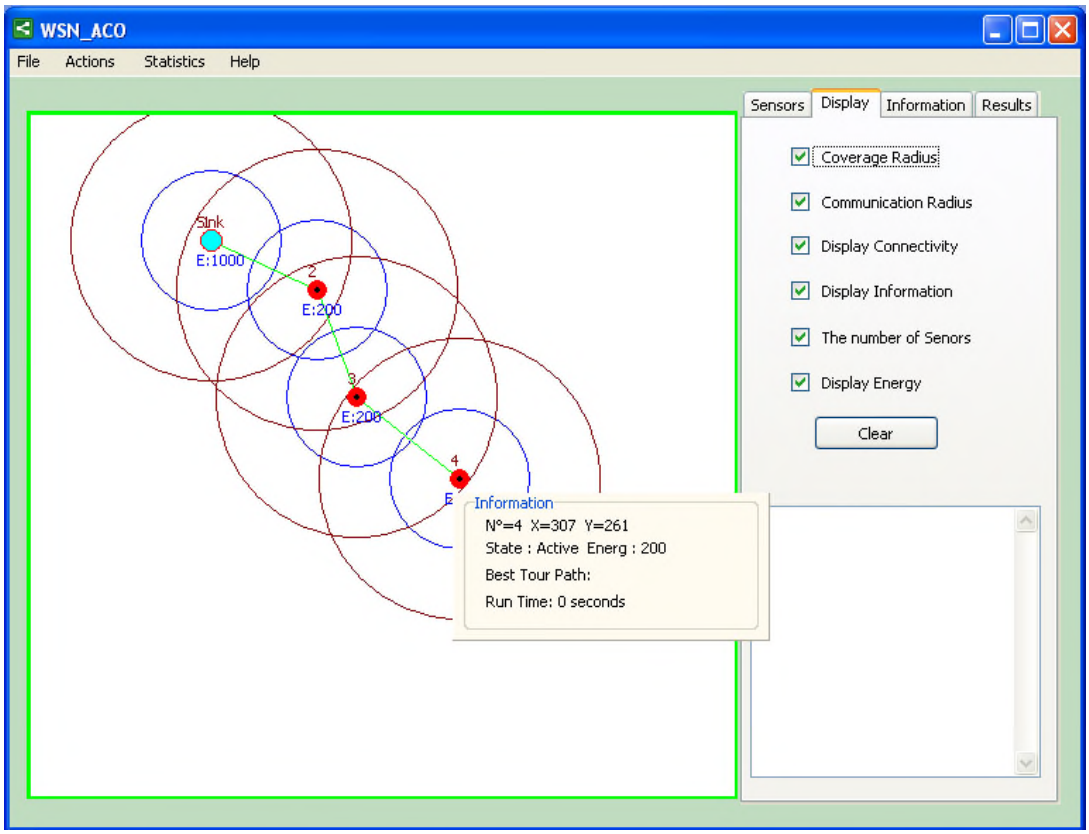

Figure 4.4 Displaying network information.

4.5.3 Simulation Run mode

After initialization of network and ant system parameters the simulation can be run with one of the two modes:

Real time: In this mode the network generate in each iteration a random event , while the algorithm of finding an optimal path from the first node that detect the event to the sink will be run.

Sequential: In this mode, in each iteration we run at every node the proposed ant colony algorithm, this mode is suitable for the results analysis.

Figure 4.5 shows an example of a real time simulation.

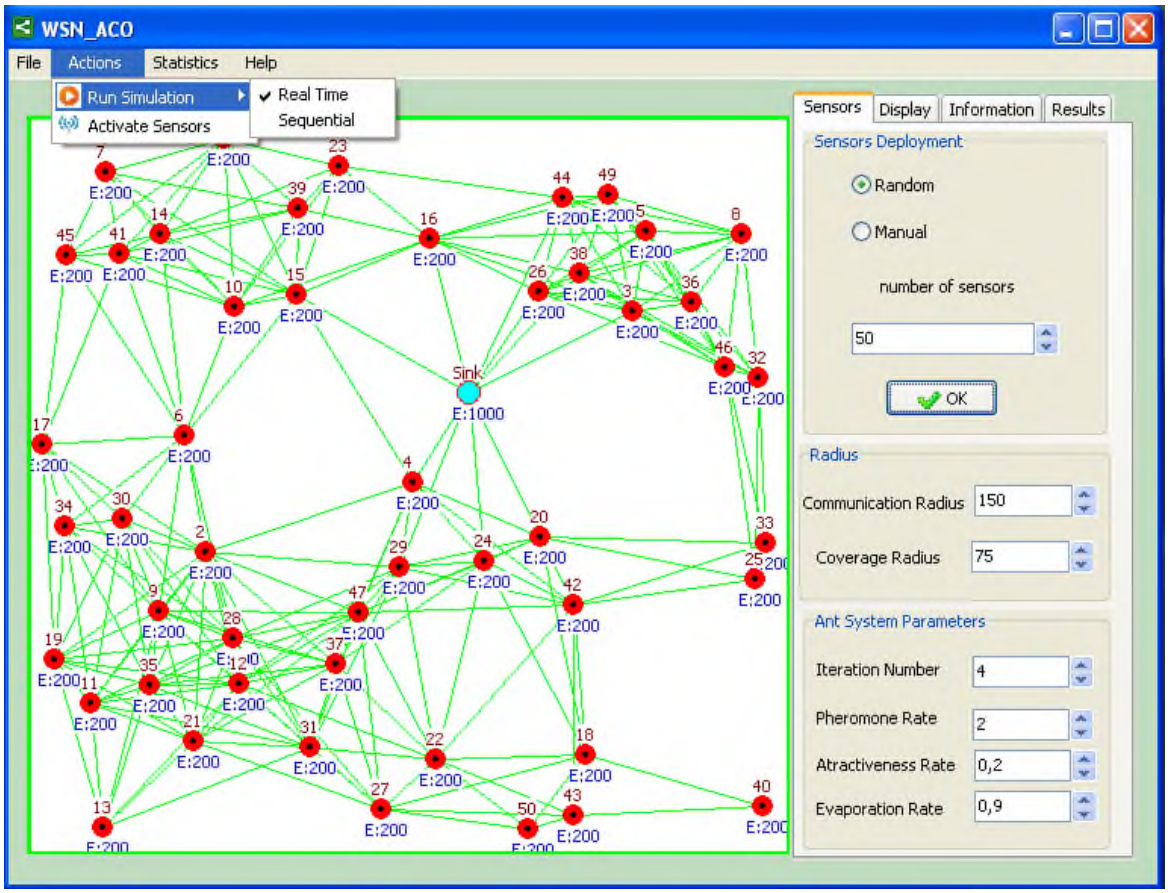

Figure 4.5 Example of real time simulation.

After the initialization of network and algorithm parameters, we show one of the simulation phases when the event is detected by the seventh node, tracing the forward discovering path presented by red color from the node to the sink figure 4.6.

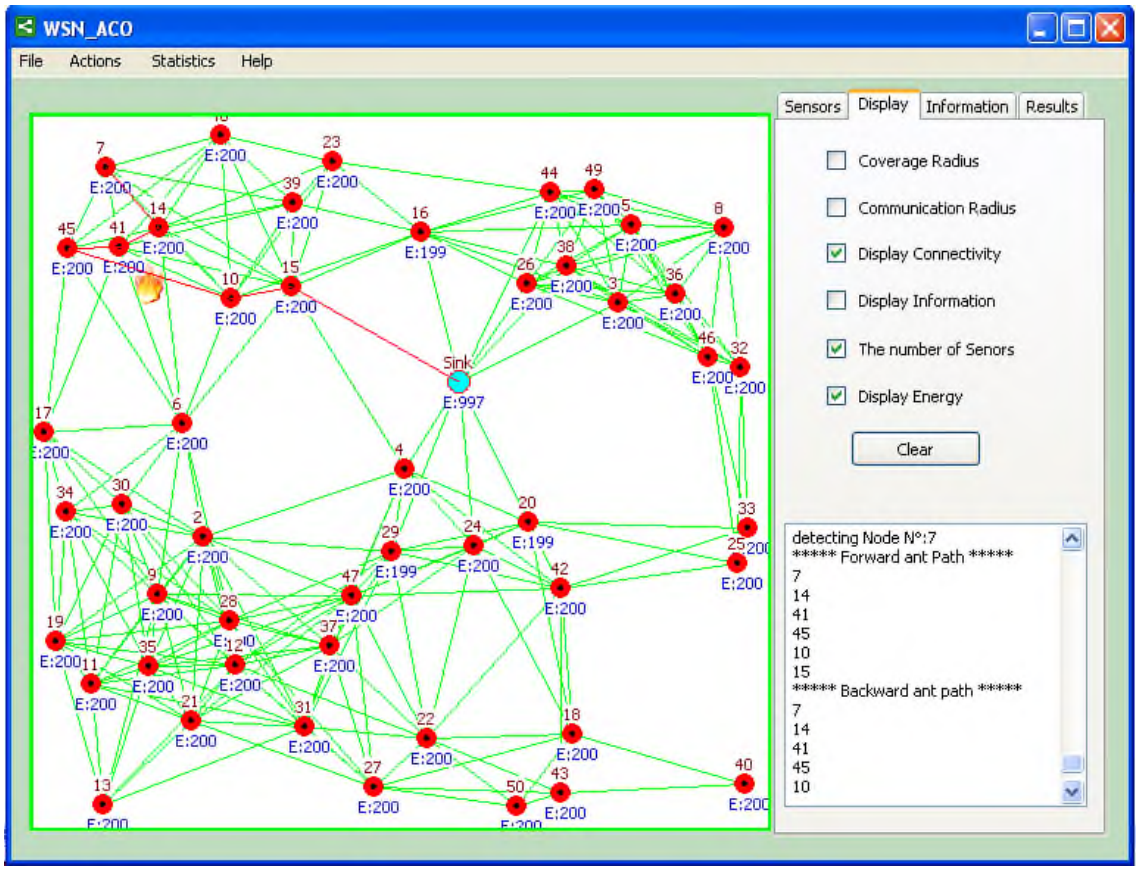

Figure 4.6 Forward path from the node 7 to the Sink.

And the reverse backward ant presented by purple color as shown in figures 4.7 with a brief running description with the path length in the memo box.

During the retracing path to source node the ant take into consideration two tasks:

- **Loop elimination**: the ant eliminates the loop if exist in the forward path, in the case of multiple loops, it eliminates the long loop.
- **Pheromone update**: the ant deposit pheromone on the edges of the forward path after eliminating loops depending on the remaining energy as described in the third chapter, until it reaches the source node.

The path constructed between the source node and the sink, will be used to transmit data to the sink resulting to the reduction of energy consumption and connectivity insurance.

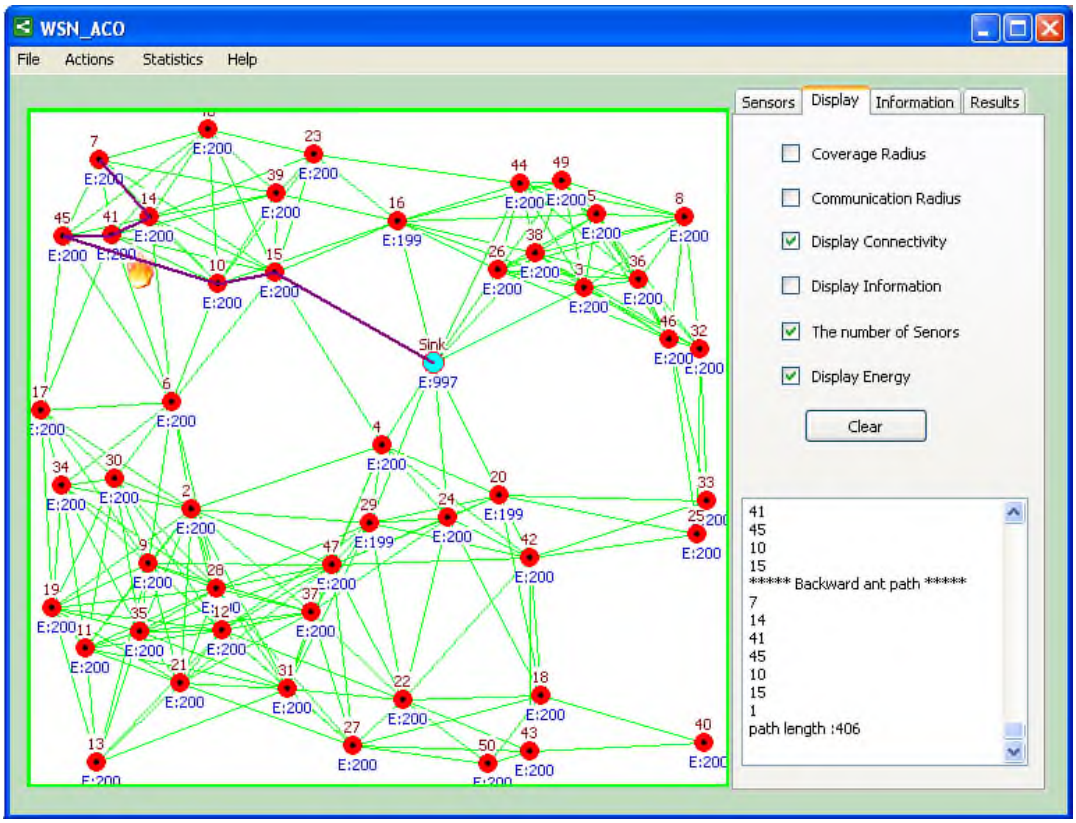

Figure 4.7 Resulting path from the node 7 to the Sink (backward Path).

The figure below illustrate the results obtained after a simulation of detecting 4 events by nodes 7, 2, 16, 5 available in the tab sheet «Results».

N°of sensor	Path to Sink	Path Lergth	Run Time
7	7,14,41,45,10,15,1,	406	6
2	2,9,28,12,21,35,11,19,34,30,6,1	760	13
16	16,1,	105	1
5	5,49,44,1,	207	3

Figure 4.8 Real time simulation results.

4.6 Simulation and Results

Various simulation studies are performed to analyze the effects of the different ant system parameters α, β, ρ related to the remaining energy, the connectivity level and the consumed energy to the network lifetime and the convergence of the algorithm to optimal path.

4.6.1 Performance Evaluation Parameters

The proposed work needs to be evaluated against various parameters to measure its Performance. These parameters are:

1) Average Path Length: average path length is the main parameter that describes the performance of the algorithm in finding the optimal path in term of distance.
2) Average Residual Energy: present the energy level of the network after running the algorithm from each node in the network to the sink.
3) Simulation Run Time: is the length of time that the proposed algorithm lasts in its discovering path.

4.6.2 Simulation parameters

The following table shows the default network construction parameters.

Number of sensors	50 nodes
Sensors energy	200 nJ
Sink energy	1000 nJ
Transmission Energy	1nJ/packet
Receiving Energy	1nJ/packet
Communication radius	150 m
Coverage radius	75 m
Number of ants	50
Number of Iterations	5
Pheromone Rate α	2

Table 4.1 Network parameters

4.6.3 The effect of evaporation rate ρ

Evaporation rate as defined earlier has two effects, one is to conserve the good path discovered when it is small and vice versa , secondly , it avoid the accumulation of the pheromone that make hard the ants move decision.

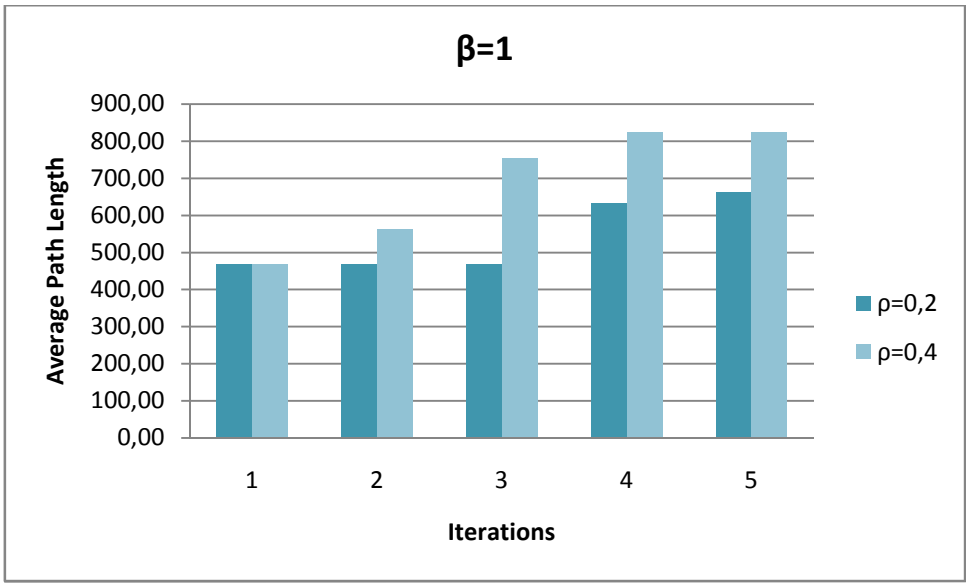

Figure 4.9 ρ effects on Average Path Length.

Figure 4.9 shows the effect of ρ variation with $\beta = 1$, during 5 iterations , average path length when $\rho = 0{,}2$ is lesser then $\rho = 0{,}4$ which is appear after each iteration. Means that evaporation in this case is not suitable.

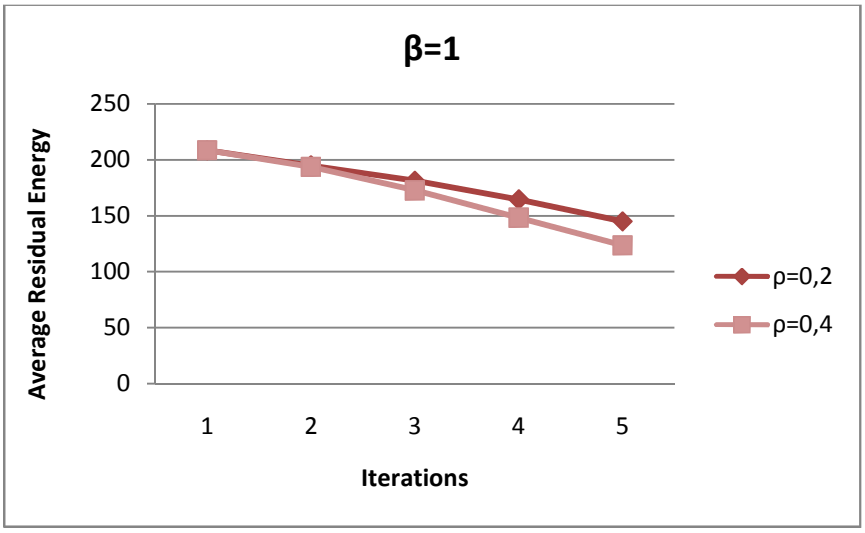

Figure 4.10 ρ effect on Average Residual Energy.

Figure 4.10 shows that the average residual energy in the two evaporation cases is approximately the same even the average path length discussed first is so different.

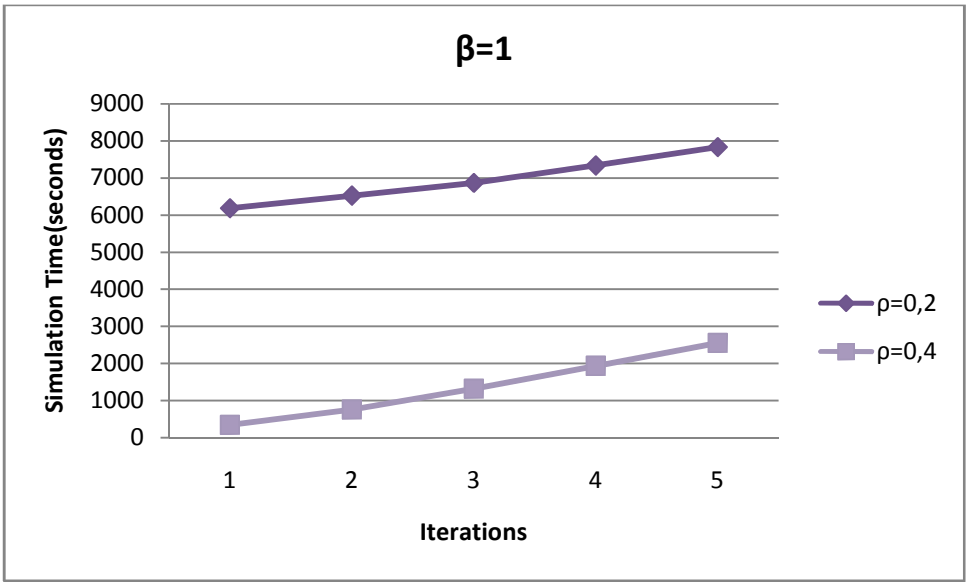

Figure 4.11 ρ effect on Simulation Time.

The figure above shows the discovering path simulation time , the result illustrates that when $\rho = 0{,}2$ the ants takes more time to reach the sink , means either the decision is to take the long path or they construct more cycle in that path (the past iterations is not good), contrary to the results obtained when $\rho = 0{,}4$. When evaporation rate is increased its lead to accelerate the searching process means delete the bed results of the algorithm iterations.

4.6.4 The effect of Attractiveness rate β

In the initial state the ant decide to move to the next node depending on the attractiveness of edges defined in the proposed algorithm depending on the energy and connectivity level of sensor node. The attractiveness rate presents its importance.

In the following we analyze the results of attractiveness rate on the algorithm running , taking that evaporation rate $\rho = 0{,}9$. The evaporation is coming to be total.

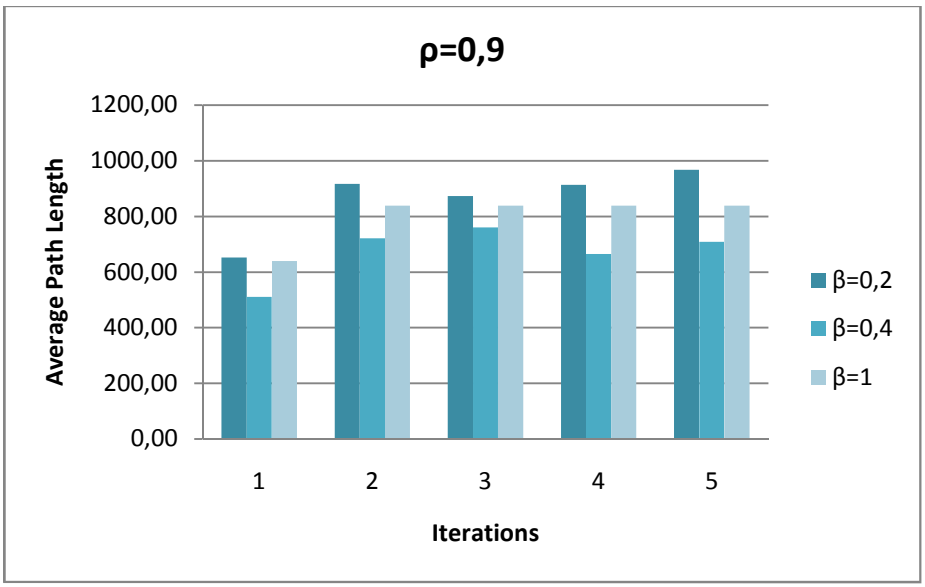

Figure 4.12 β effect on Average Path Length.

The figure 4.12 shows a perturbation in finding optimal path , in term of distance , after each iteration the length of the path decrease. In general , the good value of attractiveness parameter is $\beta = 0{,}4$.

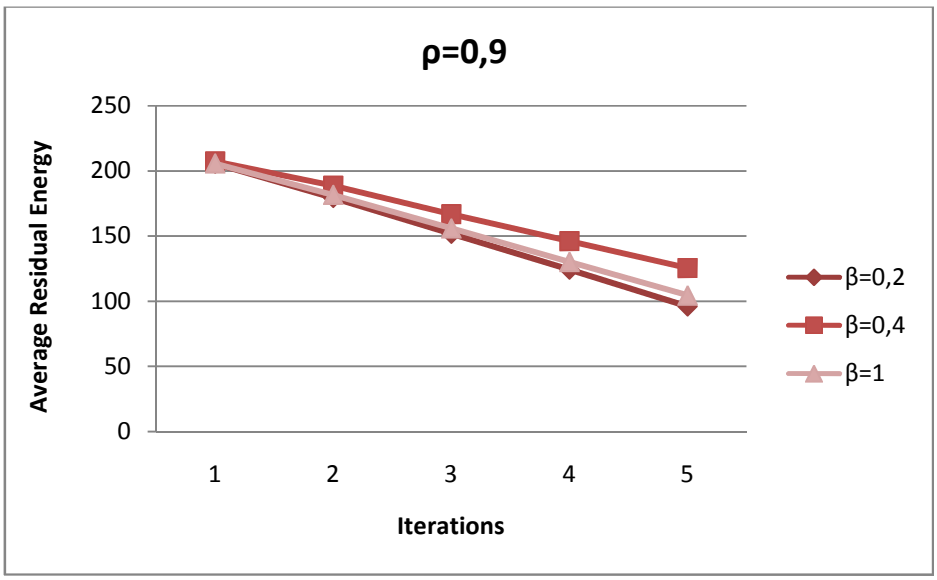

Figure 4.13 β effect on Average Residual Energy.

Average residual energy as figure 4.13 illustrates , is approximately the same for $\beta = 0{,}2$ and $\beta = 1$, where it is better in the three last iterations when $\beta = 0{,}4$.

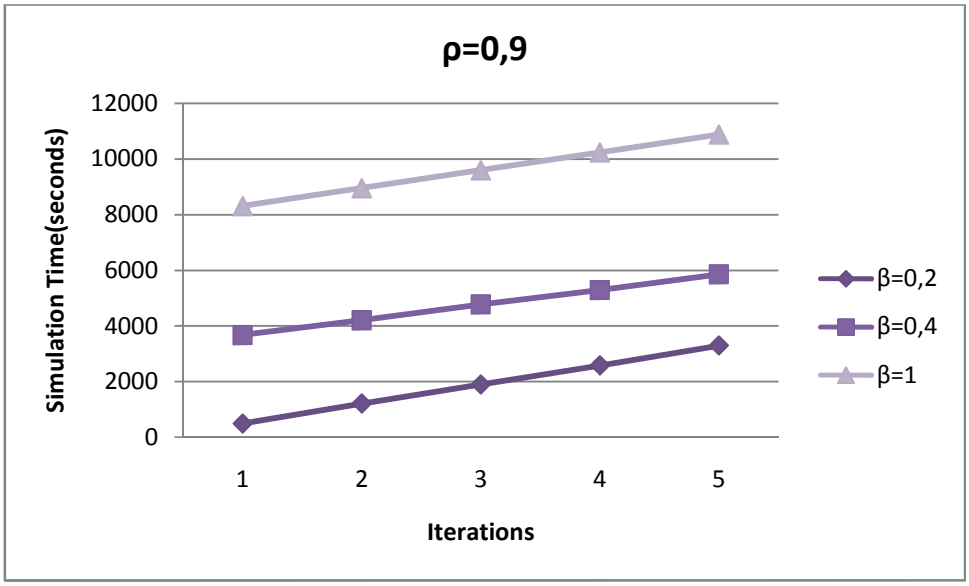

Figure 4.14 β effect on Simulation time.

The figure above shows as that the losing time of simulation (discovering path time) is the greatest in case of $\beta = 1$ and the least when $\beta = 0{,}2$ during all the 5 iterations. Where it is acceptable when $\beta = 0{,}4$.

So , when $\rho = 0{,}9$ and $\beta = 0{,}4$ the average path length , residual energy are good compared to other attractiveness rate values , where the simulation time is not bad.

4.7 Conclusion

After studying the effects of both evaporation and attractiveness rate on each other , we conclude that the proposed algorithm is more sensitive to the variation of the evaporation rate when we search to find the optimal path distance, and to the variation of the attractiveness rate when we search to conserve energy and gain time.

Conclusion and future work

Routing in WSNs is an essential issue, where the protocol try to find the optimal path without affecting the network performance. Also, it should be energy aware to prolong the network life time. An optimal path Ant Colony Algorithm was proposed , where the main decision parameters are energy level, connectivity level and the energy consumed by a sensors node. Depending on the application and traffic types, different weights may be given.

A study of this algorithm was performed, and a different values was analyzed for the effect of pheromone rate α, attractiveness rate β or what called heuristic importance value and evaporation rate ρ on the decision of finding the optimal path , the energy consumed and the time spent to find it.

Increasing evaporation rate lead to make the path longer if the chosen pheromone and attractiveness values are good, where it will be the best solution if the later parameters are not initialized with suitable values for network. In some cases the high evaporation rate decrease the chance of finding the shortest path but it makes the process faster.

Increasing attractiveness rate give a big importance to the energy and connectivity level of the sensor node, this two metrics have a result in the algorithm convergence to find the less energy consumption path in hope that the high connectivity level of a node can help it to discover new path that may be optimal. So, the variation of attractiveness rate has a noticeable result into network remaining energy and path discovering time.

The big challenge in application of the Ant colony Algorithm in wireless sensors network is to describe the goal behind the routing algorithm design and define its primary needs where good ant parameters adjustment will take the routing protocol to reach to high level efficiency.

As a future work it could be proposed to apply the ant colony algorithm to an hierarchical clustering network where there are two kinds of nodes, the cluster heads and their members, where we will try to apply the ant probability decision into cluster head formation and the vice-cluster head selection.

References

[1]-M. Saleem, Gianni A. Di Caro, F. Muddassar, « Swarm Intelligence Based Routing Protocol For Wireless Sensor Networks: Survey And Future Directions», Information Sciences 181 (2011) 4597–4624, 23 July 2010.

[2]- E. DHIB, « Routage avec QoS temps réel dans les réseaux de capteurs», RAPPORT DE PROJET DE FIN D'ETUDES, Ecole supérieure des communications en Tunis, 2006-2007.

[3]- K. BEYDOUN, «Conception D'un Protocole De Routage Hiérarchique Pour Les Réseaux De Capteurs», PhD thesis, University of Franche-Comte, 2009.

[4]-Tao Wu, « Off-Network control processing for scalable routing in very large sensor networks », PhD thesis, Michigan State University, 2008.

[5]- W. MASRI, « QoS requirements mapping in TDMA-based Wireless Sensor Networks», PhD thesis, University of Toulouse, 8 July 2009.

[6]- J. Lester Hill,« System Architecture for Wireless Sensor Networks », PhD thesis, Univeritsy Of California, Berkeley, 2003.

[7]- R. Mulligan, H. M. Ammari, « Coverage in Wireless Sensor Networks: A Survey», ISSN 1943-3581 Vol. 2, No. 2, 2010.

[8]-A. Ghosh , S. K. Das ,« Coverage and Connectivity Issues in Wireless Sensor Networks», In Rajeev Shorey, Akkihebbal L. Ananda, Mun Choon Chan, and Wei Tsang Ooi, « Mobile, Wireless, And Sensor Networks Technology, Applications, And Future Directions », pp.221-256, 2006.

[9]-Wei Mo, Daji Qiao and Zhengdao Wang, « Mostly-Sleeping Wireless Sensor Networks: Connectivity , k-Coverage, and -Lifetime », Iowa State University , Ames, IA 50011.

[10]- M. Demri, E. Mamouni, « Mécanisme D'auto-Organisation Pour La Tolérance Aux Fautes Dans Un Réseaux De Capteurs Sans Fil », engineering thesis, University of Bechar, June 2009.

[11]- P. Kumar & al., « A Review of Routing Protocols in Wireless Sensor Network », International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 4, ISSN: 2278-0181, June 2012.

[12]- S. K. Singh & al., « Routing Protocols in Wireless Sensor Networks – A Survey », International Journal of Computer Science & Engineering Survey (IJCSES) Vol.1, No.2, November 2010.

[13]- J. S. Rauthan, S. Mishra, « An improved Cluster Based Multi-hop Routing in SelfOrganizing Wireless Sensor Networks », International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 4, ISSN: 2278-0181, June 2012.

[14]- P. Nelly,« A Location Routing Protocol Based On Smart Antennas For Wireless Sensor Networks », thesis, University of Montréal, June 2010.

[15]- A. Norouzi, A. H. Zaim, «An Integrative Comparison of Energy Efficient Routing Protocols in Wireless Sensor Network», Wireless Sensor Network,pp. 65-75, 2012.

[16]- R. V. Biradar & al., « Inter-Intra Cluster Multihop-LEACH Routing In Self-Organizing Wireless Sensor Networks », International Journal of Research and Reviews in Computer Science (IJRRCS), Vol. 2, No. 1, March 2011.

[17]-T. N. Quynh, & al., « Multipath Routing for Cluster-based and Event-based Protocols in Wireless Sensor Networks », SoICT 2012, August 23–24, 2012, Ha Long, Vietnam.

[18]-A. A. Femi, « Energy Management Techniques in Wireless Sensor Networks: Protocol Design and Evaluation», Master Thesis, University of Otago, Dunedin, New Zealand, March 2010.

[19]-F. Dressler, « Self-Organization in Ad Hoc Networks: Overview and Classification », TECHNICAL REPORT, University Of Erlangen, February 2006.

[20]-F. Dressler, 2007, « Self-Organization in Sensor and Actor Networks », John Wiley & Sons, Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO 19 8SQ, England, http://www.docin.com/p-645743420.html .

[21]-F. Dressler,2008, « Bio-Inspired Networking Self-organizing Networked Embedded Systems », In Rolf P.Wurtz, «Organic Computing», Springer- Verlag Berlin Heidelberg.

[22]-S. Rizvi, « Self Organization and Survivability of Wireless Sensor Networks », University of Tartu, Estonia.

[23]-F. Dressler, «Bio-Inspired Networking: From Theory to Practice » , IEEE Communications Magazine, November 2010, pp.176-183.

[24]-V. Selvi, R. Umarani, «Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques », International Journal of Computer Applications (0975 – 8887) ,Vol.5, No.4, August 2010, pp.1-6.

[25]-M. Melanie , «Genetic Algorithms: An Overview» in «An Introduction to Genetic Algorithms» , A Bradford Book The MIT Press Cambridge, England ,Fifth printing, 1999,158 pages.

[26]-F. Dressler, «Benefits of Bio-inspired Technologies for Networked Embedded Systems: An Overview », Autonomic Networking Group, University of Erlangen-Nuremberg ,Martensstr. 3, 91058 Erlangen, Germany.

[27]-S. C. Chu & al., «Overview of Algorithms for Swarm Intelligence », Springer-Verlag Berlin Heidelberg, 2011,pp. 28-41.

[28]-V. Maniezzo & al. , « Ant Colony Optimization», pp.1-21.Christian Blum, «Ant colony optimization: Introduction and recent trends», Physics of Life Reviews, 2005, pp.353–373.

[29]-M. Dorigo, T. Stutzle, «The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances », pp.1-42.

[30]-T. H. Labella, F. Dressler, «A Bio-Inspired Architecture for Division of Labour in SANETs », Autonomic Networking Group, University of Erlangen-Nuremberg , Martensstr. 3, 91058 Erlangen, Germany.

[31]-K.Saleem & al., «Ant based Self-organized Routing Protocol for Wireless Sensor Networks », International Journal of Communication Networks and Information Security (IJCNIS), Vol. 1, No. 2, August 2009, pp.42-46.

[32]-Y. Hamouda, C. Phillips, «Biologically Inspired, Cooperative Target Tracking Framework for Wireless Sensor Networks »,London Communications Symposium 2009, Queen Mary University of London, United Kingdom.

[33]-K. Saleem & al., «Ant Colony inspired Self-Optimized Routing Protocol based on Cross Layer Architecture for Wireless Sensor Networks », WSEAS Transactions on Communications, Issue 10, Vol.9, October 2010, pp.669-678.

[34]-M. Dorigo, T. Stützle, «Ant Colony Optimization», A Bradford Book The MIT Press, Cambridge, Massachusetts, London, 2004,305 pages.

[35]-M. Dorigo, K. Socha, «An Introduction to Ant Colony optimization» in «Approximation Algorithms and Metaheuristics», IRIDIA-Technical Report Series, Bruxelles, T. F. Gonzalez, Technical report N° TR/IRIDIA/2006-010, April 2006, 19pages.