

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique  
Université Ahmed Draia - Adrar  
Faculté des Sciences et de la Technologie  
Département des Mathématiques et Informatique



Mémoire de fin d'étude, en vue de l'obtention du diplôme de Master en informatique

**Option : Réseaux et systèmes intelligents**

**Thème**

**Une approche hybride pour conception d'un protocole de routage multi-saut dans les réseaux de capteur sans fil**

Préparé par

**Abdelkader TOUDJI et Abdallah DLIMI**

Présenté le 21 mai 2017, devant le jury composé de :

Mr. Mohammed DEMRI	Encadreur
Mr. Djilali CHOUGUEUR	Président
Dr. Mohammed OMARI	Examineur
Mr. Mohammed KADDI	Examineur

Année Universitaire 2016/2017

## Résumé

Les réseaux de capteurs sans fil (RCSF) représentent une technologie émergente qui visent à offrir des capacités innovantes. Leur utilisation ne devrait cesser d'augmenter et ceci dans de nombreux domaines. Cependant, la limitation de ressources des nœuds capteurs constitue une contrainte importante, principalement en termes d'autonomie d'énergie. C'est pourquoi de nombreux travaux portent sur la conception des protocoles de routage avec une gestion efficace d'énergie. L'objectif de ce travail est d'adapter et implémenter un protocole de routage bio-inspiré (AFABCMH), basé sur une hybridation entre l'Algorithme des Lucioles (Firefly Algorithm) et une stratégie de routage multi-sauts pour l'économie de l'énergie et la prolongation de la durée de vie d'un RCSF.

**Mots-clés:** Protocole, Routage, RCSFs, Lucioles, Energie, Multi-Sauts.

## Abstract

Wireless sensor networks (WSN) represent an emerging technology which aim to offer innovative capabilities. Their use is continue to be increased and this in many areas. However, the limitation of resources of the sensor nodes is an important constraint, mainly in terms of energy autonomy. That why many works were carried on the design of routing protocols with efficient energy management. The objective of this work is to adapt and implement a bio-inspired (AFABCMH) routing protocol based on hybridization between Firefly Algorithm and a multi-hop routing strategy for the energy economy and the extension of the lifetime of WSN.

**Keywords:** Protocol, Routing, WSN, Firefly, Energy, Multi-hop.

## ملخص

تعتبر شبكات الاستشعار اللاسلكية (WSN) تكنولوجيا ناشئة التي تهدف إلى توفر القدرات الابتكارية. إن استخدام هاته الشبكات في ازدياد مستمر وذلك في العديد من المجالات. إن الموارد المحدودة لعقد الاستشعار تمثل عقبة هامة، و خاصة من حيث استقلالية الطاقة. ولهذا هو السبب، فقد أنجزت العديد من الأعمال الرامية إلى تصميم بروتوكولات التوجيه التي إدارة الطاقة بكفاءة. الهدف من هذا العمل هو تكييف وتنفيذ بروتوكول توجيه مستوحاة من الطبيعة على أساس التهجين بين "خوارزمية اليراع" و استراتيجيات متعددة القفزات وذلك لاقتصاد الطاقة وتمديد مدة حياة شبكات الاستشعار اللاسلكية.

**الكلمات المفتاحية:** البروتوكول، التوجيه، شبكات الاستشعار اللاسلكية، اليراع، الطاقة. متعدد القفز

# Dédicaces

Je dédie ce travail :

A mes chers parents qui ont toujours été là pour moi.

A mes frères et mes sœurs

A mon encadreur DEMRI Mohammed

A mes collègues DLIM. R et TOWATI. M

A tous mes chers enseignants

A toute ma belle-famille ;

Et ainsi qu'à tous mes amis

TOUDJI Abdelkader

# Dédicaces

Je dédie ce travail :

A mes chers parents qui ont toujours été là pour moi.

A mes frères et mes sœurs

A mon encadreur DEMRI Mohammed

A mes collègues DLIM. R et TOUATI. M

A tous mes chers enseignants

A toute ma belle famille ;

Et ainsi qu'à tous mes amis

DLIMI Abdallah

## Remerciements

Nous remercions en premier lieu DIEU le tout puissant pour nous avoir donné la force de réaliser ce travail.

Nous tenons tous d'abord à remercier nos parents pour leurs soutiens et leurs encouragements durant nos années d'études.

Nous tenons à remercier profondément notre encadreur

Mr. DEMRI Mohammed pour son aide, ses encouragements, et ses critiques constructives qui nous ont beaucoup aidés à apprécier ce travail.

Nous remercions également tous les membres du jury d'avoir accepté à participer à l'évaluation de notre travail.

Nous exprimons aussi toute notre gratitude aux enseignants du département des mathématiques et d'informatique ainsi que tous nos collègues et nos amis (es) et à tous ce qui ont contribué de près ou de loïn à la réalisation de notre mémoire.

TOUDJI Abdelkader & DLIMI Abdallah

# ***SOMMAIRE***

Résumé .....	II
Dédicaces .....	III
Remerciement .....	V
Sommaire.....	VI
Liste des figures .....	X
Liste des tableaux .....	XII
List des Abréviation .....	XIII
Introduction générale.....	1
<b>Chapitre I : Introduction aux réseaux de capteurs sans-fil</b>	
I.1 Introduction .....	5
I.2. Les capteurs .....	5
I.2.1 Définition.....	5
I.2.2. Architecteur d'un capteur.....	5
I.3. Les réseaux des capteurs sans fil .....	7
I.3.1. Architecture et Caractéristiques d'un réseau de capteurs sans fil.....	9
I.3.1.1 Les types d'architecture des RCSF .....	9
I.3.1.2 Caractéristiques d'un réseau de capteurs sans fil.....	11
I.3.2. Collecter les informations .....	13
I. 3.2.1 Collecter les informations à la demande .....	13
I.3.2.2 Collecter les informations à suite à un événement.....	13
I.4. Pile protocolaire.....	14
I.4.1 Rôle des couches .....	15
I.5. Domain d'application .....	16
I.5.1. Applications militaires .....	16
I.5.2. Applications liées à la sécurité.....	17

I.5.3. Applications environnementales .....	18
I.5.4. Applications commerciales .....	18
I.5.5. Applications médicales .....	19
I.6. Contraintes dans la conception d'un réseau de capteurs .....	20
I.6.1. Contraintes liées à l'application .....	20
I.6.2. Contrainte énergétique .....	20
I.6.3. Contraintes liées aux déterminismes .....	20
I.6.4. Contraintes de passage à l'échelle .....	21
I.6.5. Contraintes liées à la qualité de service .....	21
I.6.6. Contraintes liées à la protection de l'information .....	21
I.6.7. Contraintes liées à l'environnement .....	21
I.6.8. Contraintes de simplicité.....	22
I.7. Conclusion .....	22

## **Chapitre II : Protocoles de routage dans les réseaux de captures**

II.1. Introduction .....	24
II.2. Le routage dans les RCSF .....	24
II.2.1. Les type de routage.....	24
II.2.1.1 Protocole proactif.....	24
II.2.1.2. Protocole réactif.....	25
II.2.2. Les protocoles de routage .....	25
II.2.2.1 Les protocoles de routage à plat (non hiérarchiques) .....	25
II.2.2.2. Protocoles de routage hiérarchiques .....	29
II.2.2.3. Les protocoles de routage géographique.....	37
II.3. Conclusion.....	38

## **Chapitre III : Algorithme des Lucioles (Firefly)**

III.1. Introduction.....	40
III.2 Algorithme des Lucioles (FireFlyAlgorithm).....	40
III.2.1. Inspiration.....	40
III.2.2. Principe de fonctionnement de l'algorithme des Lucioles.....	41
III.2.3. Description de l'organigramme .....	43
III.2.3.1. Génération de la population initiale.....	43
III.2.3.2. Fonction d'évaluation .....	43
III.2.2.3. Classement.....	44
III.2.2.4. Déplacement et mise à jour .....	44
III.2.2.5. Critère d'arrêt .....	45
III.3. Paramétrages d'algorithmes des Lucioles .....	46
III.4. Les avantages et les inconvénients d'algorithme de luciole .....	48
III.4.1 Les avantage de l'Algorithme Firefly .....	48
III.4.1 Les inconvénients du Algorithme Firefly.....	49
III.5. Conclusion .....	49

## **Chapitre IV : Implémentation et discussion**

IV.1 Introduction.....	51
IV .2 Description général de l'algorithme proposé .....	51
IV .2.1. Algorithm AFABCMH (Adaptive Firefly Algorithm Base Clustering Multi-hop).....	51
IV.3. Description détaillée de l'algorithme proposé.....	52
IV .3.1. Formulation des CHs .....	53
IV 3.2. Fonction objective.....	56
IV 4. Choix de l'environnement d'implémentation.....	58



IV .5. Les étapes d'exécution de l'application.....	59
IV 5 .1 Lancement d'application .....	59
IV 5.2 Description et paramètres de simulation.....	59
IV.5.2.1Paramètres de simulation.....	59
IV.5.2.2Paramètres d'algorithme des lucioles .....	61
IV.5.2.3 Paramètre d'affichage .....	61
IV.6. Simulation et résultat .....	61
IV.7. Conclusion .....	65
Conclusion générale.....	67
Bibliographie.....	68

## Liste des figures

### Chapitre I

<b>Figure I.1:</b>	Architecture d'un capteur sans fils. ....	6
<b>Figure I.2:</b>	Exemples des capteurs sans fils .....	7
<b>Figure I.3:</b>	Le microcontrôleur.....	8
<b>Figure I.4:</b>	Réseau de capteur sans fil (WSN).....	9
<b>Figure I.5:</b>	Architecture plate.....	10
<b>Figure I.6:</b>	Architecture hiérarchique.....	10
<b>Figure I.7:</b>	Exemple de communication multi-saut dans un réseau de capteurs .....	13
<b>Figure I.8:</b>	Collecter les informations à la demande.....	13
<b>Figure I.9:</b>	Collecter les informations Suite à un événement.....	14
<b>Figure I.10:</b>	La pile protocolaire des RCSF.....	15
<b>Figure I.11:</b>	Domaine d'application de RCSFs.....	16
<b>Figure I.12:</b>	Un service militaire utilisant les RCSF .....	17
<b>Figure I.13:</b>	Le flux d'information d'un patient.....	19

### Chapitre II

<b>Figure II.1:</b>	Diffusion pure et diffusion en utilisant les MPRs dans OLSR.....	27
<b>Figure II.2:</b>	Fonctionnement du protocole SPIN .....	29
<b>Figure II.3:</b>	Architecture en cluster.....	31
<b>Figure II.4:</b>	Répartition du temps et différentes phases pour chaque round.....	32
<b>Figure II.5:</b>	Formation des chaînes gourmandes dans PEGASIS.....	33

### Chapitre III

<b>Figure III.1:</b>	Le Luciole.....	43
<b>Figure III.2:</b>	Organigramme de l'algorithme des lucioles.....	45
<b>Figure III.3:</b>	Déplacement des lucioles .....	47
<b>Figure III.3:</b>	l'algorithme des Lucioles.....	48

### Chapitre IV

<b>Figure IV.1:</b>	Algorithme AFABCMH .....	54
<b>Figure IV.2:</b>	Algorithme MEMBRES-CH .....	56
<b>Figure IV. 3:</b>	Algorithme CHANGER-CH .....	57
<b>Figure IV.4:</b>	Algorithme UPDATE-RESEAU .....	58
<b>Figure IV.5:</b>	Algorithme MULTI-SAUT .....	59
<b>Figure IV.6:</b>	l'organigramme de protocole AFABCMH .....	60
<b>Figure IV.7:</b>	L'interface principale de l'application .....	62
<b>Figure IV.8:</b>	Modèle de consommation d'énergie pour la communication .....	63
<b>Figure IV.9:</b>	La consommation d'énergie après chaque rond. ....	66
<b>Figure IV.10:</b>	Durée de vie de réseau (Leach, AFABC, AFABCMH).....	67
<b>Figure IV.11:</b>	Courbe représenter transmission de (donnes noude-CH) .....	68

## Liste des tableaux

### Chapitre IV

<b>Tableau IV.1</b> Les paramètres de simulation .....	65
<b>Tableau IV.2</b> Les paramètres de l'algorithme des lucioles.....	65
<b>Tableau IV.3</b> la consommation d'énergie après 500 chaque rond.....	65
<b>Tableau IV.4</b> nombre de nœuds vivant après après 500 rond.....	66
<b>Tableau IV.5</b> représente la transmission des données entre nœuds et CH .....	67

## Liste des Abbreviation

<b>AFABC</b>	Adaptive Firefly Algorithm base clustering
<b>AFABCMP</b>	Adaptive Firefly Algorithm base clustering Milt hops
<b>APTEEN</b>	Adapted Threshold sensitive Energy Efficient sensor Network
<b>BS</b>	base station
<b>CH</b>	Cluster Head
<b>CDMA</b>	Code Division Multiple Access
<b>DSDV</b>	Destination Sequenced Distance Vector
<b>HEED</b>	Hybrid, Eenergy-Efficient, and Distributed approach
<b>GAF</b>	Geographic Adaptive Fidelity
<b>GEAR</b>	Geographic and Energy Aware Routing
<b>ISO</b>	International Standard Organization
<b>LEACH</b>	Low-Energy Adaptive Clustering Hierarchy
<b>MECN</b>	Minimum Energy Communication Network
<b>MPR</b>	Multi Point Relaying,
<b>OLSR</b>	Optimized Link State Routing
<b>PEGASIS</b>	Power-Efficient Gathering in Sensor Information Systems
<b>RCSF</b>	réseaux de captures sans fil
<b>SPIN</b>	Sensor Protocols for Information via Negotiation
<b>SMECN</b>	Small Minimum Energy Communication Network
<b>TEEN</b>	Threshold-sensitive Energy Efficient sensor Network protocol
<b>TDMA</b>	Time Division Multiple Access
<b>WSN</b>	Wireless Sensor Network: WSN)



# **Introduction générale**

### 1.1 Contexte général

Le besoin effréné d'informations et l'évolution rapide de la micro-électronique et des technologies sans fil, ont permis la création de petits appareils électroniques avec un coût très réduit (ressources limitées), capables de collecter et de traiter l'information d'une manière autonome et flexible. Ces appareils peuvent être interconnectés et déployés à grande échelle, donnant naissance à un nouveau type de réseaux nommé réseau de capteurs sans fil (RCSF). Le développement des RCSFs était originalement motivé par les applications militaires (surveillance des champs de bataille, localisation de l'ennemi...). Néanmoins, leurs performances remarquables en termes de fiabilité et de faible coût ont permis de proliférer leur utilisation dans le domaine d'application civil (surveillance d'environnement, l'industrie, la domotique, la santé...).

Les réseaux de capteurs sans fil sont conçus pour fonctionner en groupe et coopérer afin de transmettre les données collectées à un point central appelé station de base ou sink. Chaque nœud capteur est équipé d'un microprocesseur à faible puissance de calcul, d'une petite batterie, d'une antenne radio et d'un ou de plusieurs capteurs. Ainsi, les RCSFs doivent opérer en prenant toujours en compte leur limitation de ressources. Ces derniers sont le plus souvent déployés aléatoirement dans des zones hostiles et inexplorées, et doivent s'auto-organiser à l'aide des communications sans fil. La station de base est le seul lien avec le monde extérieur et dispose de plus de ressources par rapport aux nœuds capteurs. Le réseau de capteurs joue le rôle d'un pont entre le monde physique et le système informatique, en fournissant des mesures et des propriétés physiques du monde réel.

Cependant, un capteur est un équipement de taille très réduite, englobant des ressources très limitées en matière de mémoire et de calcul et alimenté avec des batteries de faible puissance. ces caractéristiques spécifiques impliquent nouveaux défis lors de la mise en œuvre d'un réseau pareil. Plusieurs challenges ont en été identifiés par les chercheurs incluant la découverte du réseau, le routage et le contrôle du réseau, le traitement collaboratif des informations, et la collecte et la dissémination des données.

Alors l'économie d'énergie représente l'un des grands défis à soulever pour le bon fonctionnement des réseaux de capteurs. En effet, les nœuds capteurs sont généralement alimentés au moyen d'une petite batterie limitée en puissance, et le remplacement de celle-ci est une tâche très difficile voire impossible. Par conséquent, l'épuisement des réserves d'énergie

des nœuds capteurs implique la mise hors service du réseau tout entier. La sécurité représente un autre défi très important pour les RCSFs.

### **1.2 Motivations et problématique**

Les problèmes relatifs au routage de données sur les RCSFs et plus précisément le routage hiérarchique de données. Ce thème de recherche est très important car il offre un fondement indispensable à toutes les autres couches supérieures de ces réseaux. Le routage hiérarchique est considéré comme un outil permettant plus de performance en ce qui concerne la consommation de l'énergie par rapport aux autres types de routage, à savoir, le routage à topologie plate et le routage géographique.

Le routage hiérarchique regroupe les nœuds en clusters (groupes) où le Cluster Head (représentant ou chef de cluster) de chaque groupe a pour principale fonction la gestion des communications de paquets transmis par ses membres de groupe ou dirigés vers ces derniers. Plusieurs solutions de routage hiérarchique sont présentes des techniques différentes pour minimiser la consommation de l'énergie sur l'ensemble du réseau. Néanmoins, des améliorations restent à apporter pour atteindre de meilleures performances.

L'objectif principal de notre travail est d'adapter un nouveau protocole de routage basé sur un algorithme bio-inspiré qui simule le comportement des lucioles dans l'attraction entre eux. Cet algorithme est utilisé pour le regroupement optimal des clusters dans les réseaux de capteurs sans fils, en l'hybridant avec une technique de routage multi-sauts (multi-hop) pour la transmission des données entre les clusters et station de base.

### **1.3 Contributions et structure du mémoire**

Ce mémoire est organisé comme suit;

Dans le premier chapitre nous présenterons les réseaux de capteurs sans fil : leurs architectures de communication et leurs applications. Nous discuterons également les principaux facteurs et contraintes qui influencent sur la conception des réseaux de capteurs sans fil.

Dans le deuxième chapitre, nous présenterons un état de l'art sur l'ensemble des protocoles de routage de la littérature, hiérarchique et non hiérarchique.



Dans le troisième chapitre, nous présenterons l'algorithme des lucioles (algorithme Firefly) et son principe de fonctionnement.

Le dernier chapitre est divisé en deux parties, la première est consacrée à la présentation de notre protocole de routage hiérarchique proposé, qui s'appelle **AFABCMH**. Ce dernier est basé sur l'algorithme Firefly pour la formulation des CHs selon le facteur énergétique avec l'intégration d'une technique de routage multi-sauts (multi-hop) pour la transmission entre les clusters et station de base. Dans la deuxième partie, nous détaillerons la simulation de notre protocole adapté ainsi que sa validation.

Enfin, nous discuterons les résultats obtenus.



**Chapitre I :**  
**Introduction aux**  
**réseaux de capteurs**  
**sans fil**

## **I.1 Introduction**

Les progrès réalisés ces dernières décennies dans les domaines de la microélectronique, de la micromécanique, et des technologies de communication sans fil, ont permis de produire avec un coût raisonnable des composants de quelques millimètres cubes de volume. Ces derniers, appelés micro-capteurs, intègrent : une unité de captage chargée de capter des grandeurs physiques (chaleur, humidité, vibrations) et de les transformer en grandeurs numériques, une unité de traitement informatique et de stockage de données et un module de transmission sans fil. De ce fait, les micro-capteurs sont de véritables systèmes embarqués. Le déploiement de plusieurs d'entre eux, en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome, forme un réseau de capteurs sans fil. Selon le magazine Technologie Review du MIT, il s'agit de l'une des dix nouvelles technologies qui bouleverseront le monde et notre manière de vivre et de travailler. L'optimisation des communications dans les réseaux de capteurs sans fil industriels se base sur une connaissance des technologies de réseaux d'une part et des communications sans fil d'autre part.

Dans ce chapitre, nous présentons les RCSF, leur architecture de communication, leurs différentes contraintes ainsi que leurs diverses applications.

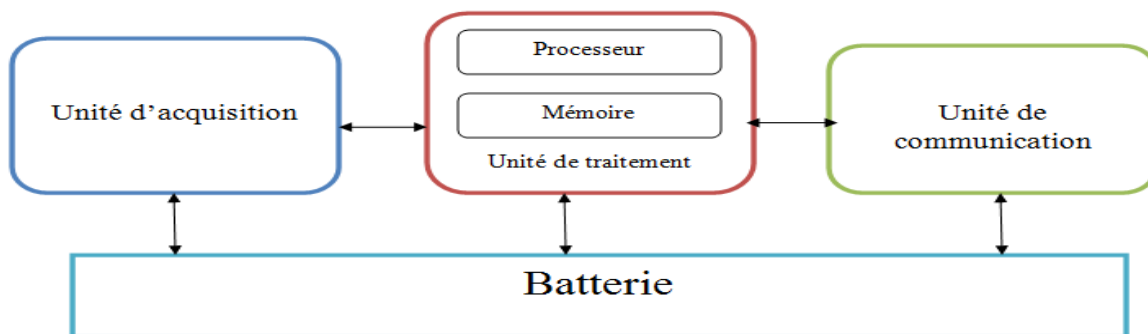
## **I.2. Les capteurs**

### **I.2.1 Définition**

Les capteurs sont des dispositifs de taille extrêmement réduite avec des ressources très limitées, autonomes, capable de traiter des informations et de les transmettre, via les ondes radio, à une autre entité (capteurs, unité de traitements...) sur une distance limitée à quelques mètres [1].

### **I.2.2. Architecteur d'un capteur**

Un nœud capteur contient quatre unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie. Il peut contenir également, suivant son domaine d'application, des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire). On peut même trouver des micro-capteurs, un peu plus volumineux, dotés d'un système mobilisateur chargé de déplacer le micro-capteur en cas de nécessité.



*Figure I.1: Architecture d'un capteur sans fils*

### **Unité de traitement**

Mote, processeur, RAM et Flash : On appelle généralement Mote la carte physique utilisant le système d'exploitation pour fonctionner. Celle-ci a pour cœur le bloc constitué du processeur et des mémoires RAM et Flash. Cet ensemble est à la base du calcul binaire et du stockage, temporaire pour les données et définitif pour le système d'exploitation. Cette unité est chargée d'exécuter les protocoles de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits.

### **Unité de communication**

Radio et antenne : les équipements étudiés sont donc généralement équipés d'une radio ainsi que d'une antenne. Cette unité est responsable d'effectuer toutes les émissions et réceptions des données sur un médium sans fil. Elle peut être de type optique (comme dans les nœuds Smart Dust), ou de type radiofréquence. Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes. Par conséquent, elles ne peuvent pas établir de liaisons à travers des obstacles.

### **Unités d'acquisition**

LED, interface, capteur : On retrouve donc des équipements de différents types de détecteur et d'autre entrée. Le capteur est généralement composé de deux sous-unités : le récepteur (reconnaissant l'analyste) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur est responsable de fournir des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique. Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement.

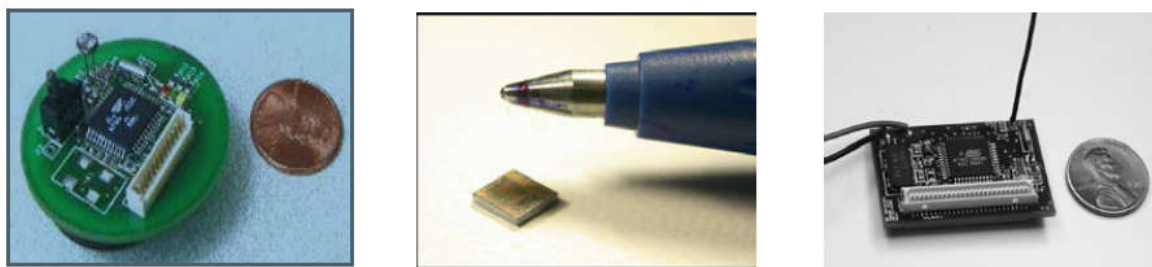
### ✍ Unités de control d'énergie

Batterie : Un micro-capteur est muni d'une ressource énergétique (généralement une batterie de type AAA) pour alimenter tous ses composants. Cependant, en conséquence de sa taille réduite, la ressource énergétique dont il dispose est limitée et généralement irremplaçable. Cette unité peut aussi gérer des systèmes de rechargement d'énergie à partir de l'environnement observé telles que les cellules solaires, afin d'étendre la durée de vie totale du réseau.

Cependant quelques différences existent suivant les fabricants. Chacun d'eux développe son type de capteurs, ces types peuvent être mica, mica2, telos ou telosb par exemple [2].

### I.3. Les réseaux des capteurs sans fil

Grâce au progrès fait dans le domaine de la miniaturisation des systèmes de micro-électro-mécanique (MEMS) et des applications sans fil, s'est créée une nouvelle branche de réseaux mobile afin d'offrir des solutions économiquement intéressantes pour la surveillance à distance et le traitement des données dans des environnements complexes: les réseaux de capteurs sans fil (Wireless Sensor networks). Ces derniers permettent de réduire l'encombrement, le coût et la consommation d'énergie et d'augmenter la précision et les performances des capteurs, des processeurs et des circuits spécifiques. Un nombre très important de capteurs peut donc être envisagé, intégré et organisé en réseau .



*Figure I.2: Exemples des capteurs sans fil*

Un réseau de capteur sans fil (Wireless Sensor Network: WSN) est un type particulier de réseau ad-hoc défini par un ensemble coopérant de nœuds capteurs dispersés dans une zone géographique appelée zone de captage afin de surveiller un phénomène et récolter ses données d'une manière autonome.

Un réseau de capteurs se compose de deux types de nœuds : des simples capteurs et des collecteurs d'informations appelés puits (sink en anglais) Le capteur est composé d'un microcontrôleur et d'un circuit radio:

✓ **Le microcontrôleur :**

Est simple et peut être embarqué aisément. Cet appareil doit répondre à l'exigence d'une faible consommation d'énergie tout en ayant la possibilité d'exécuter de simples opérations et de posséder une mémoire permettant d'emmagasiner de l'information. L'appareil doit aussi présenter la possibilité d'avoir un état oisif durant lequel il consomme une quantité d'énergie infinitésimale. Ces états oisifs peuvent parfois durer très longtemps. Le capteur peut se réveiller seulement pour capter la grandeur physique à mesurer et aussi pour effectuer des opérations de réseaux comme dialoguer avec des capteurs voisins ou relayer l'information provenant d'autres capteurs .



*Figure I.3 : Le microcontrôleur.*

✓ **Le circuit radio :**

Assure la communication du capteur avec d'autres appareils via des liens radios. Ces derniers ont facilité l'implantation massive de capteurs et ont offert une indépendance précieuse car il a réduit les coûts du câblage et de l'ingénierie nécessaire pour les installations passées. Grâce à la communication sans fil, un installateur peut déposer facilement des capteurs sans se soucier de la complexité des opérations pour les atteindre afin de relever les mesures. Il suffit d'être dans le champ de couverture radio pour transmettre ou recevoir l'information requise.

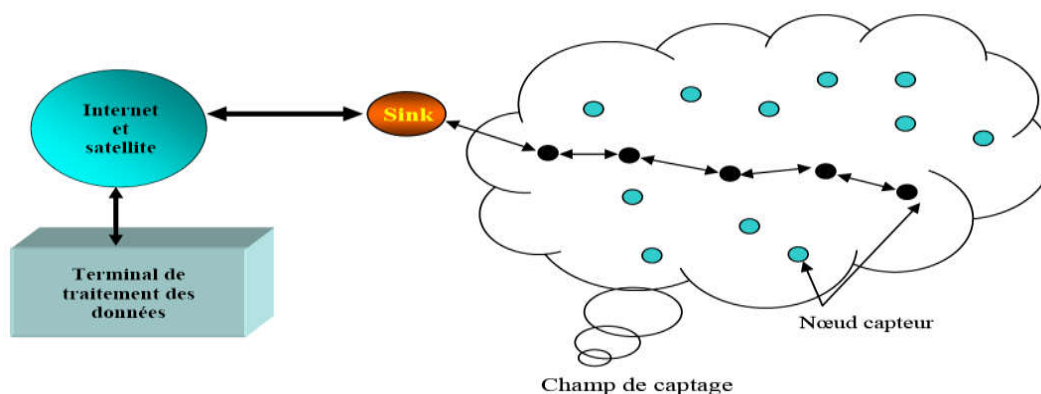
Avec ses capacités de traitement et de mémorisation, le capteur peut devenir un nœud actif dans un réseau relativement large. Lorsque le nombre de capteurs devient conséquent, la communication en réseau devient indispensable. Il n'est en effet alors plus possible d'atteindre un capteur directement par un câble ou même par une connexion radio. C'est là alors qu'on peut parler de véritables réseaux de capteurs capables de s'auto-configurer et de s'auto-organiser de manière dynamique. Ces propriétés offrent un très large spectre d'applications, notamment dans les domaines militaires, de l'environnement, de l'écologie, etc .

### I.3.1. Architecture et Caractéristiques d'un réseau de capteurs sans fil

Un réseau de capteurs sans fil est composé d'un grand nombre de nœuds. Chaque capteur est doté d'un module d'acquisition qui lui permet de mesurer des informations environnementales : température, humidité, pression, accélération ... etc.

Les données collectées par ces nœuds capteurs sont routées vers une ou plusieurs stations de base ou nœud puis (sink en anglais). Ce dernier est un point de collecte de données capturées. Il peut communiquer les données collectées à l'utilisateur final à travers un réseau de communication, éventuellement l'Internet ou un satellite. L'utilisateur peut à son tour utiliser la station de base comme passerelle, afin de transmettre ses requêtes au réseau.

En général, un RCSF est composé de quatre éléments montrés par la figure I.4 : les nœuds capteurs, une station de base, phénomène à mesurer et l'utilisateur .



*Figure 1.4: Réseau de capteur sans fil (WSN)*

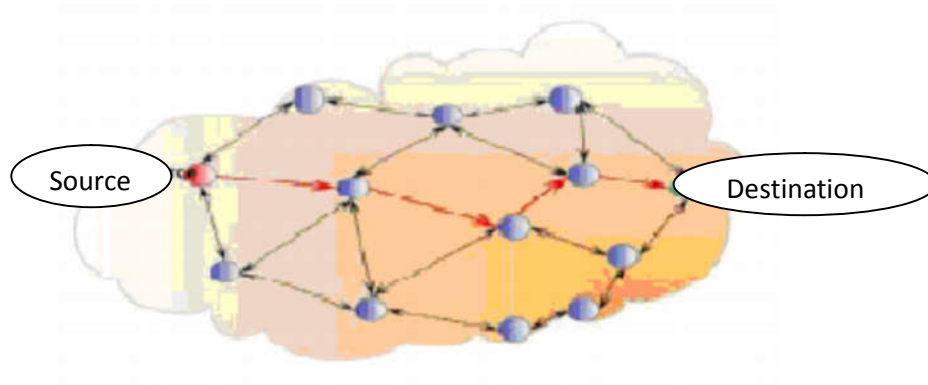
#### I.3.1.1 Les types d'architecture des RCSF

La topologie du réseau détermine l'organisation des capteurs dans le réseau. Il existe deux principales topologies dans les protocoles de routage pour les RCSF : Plate et hiérarchique [3].

Il existe deux types d'architectures pour les réseaux de capteurs sans fil

##### ✍ **Les réseaux de capteurs sans fil plats :**

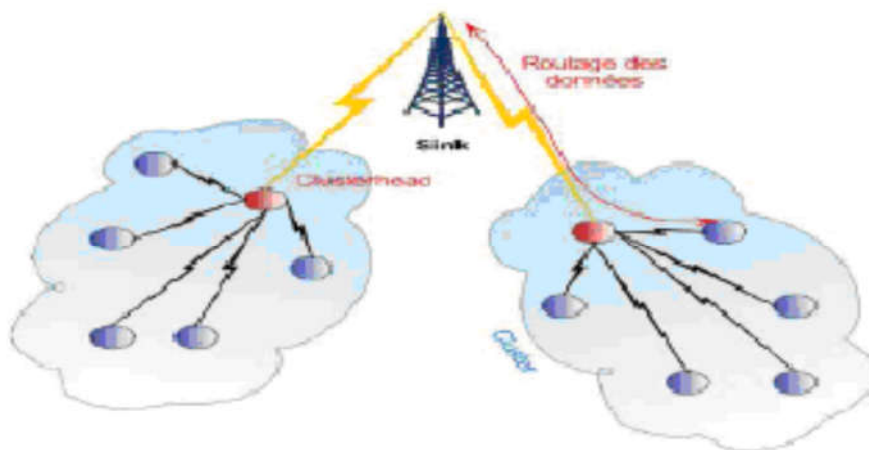
Un réseau de capteurs sans fil plat est un réseau homogène, où tous les nœuds sont identiques en termes de batterie et de complexité du matériel, excepté le Sink qui joue le rôle d'une passerelle et qui est responsable de la transmission de l'information collectée à l'utilisateur final. Selon le service et le type de capteurs, une densité de capteurs élevée (plusieurs nœuds capteurs/m<sup>2</sup>) ainsi qu'une communication multi-saut peut être nécessaire pour l'architecture plate. En présence d'un très grand nombre de nœuds capteurs, la scalabilité devient critique. Le routage et le contrôle d'accès au médium (MAC) doivent gérer et organiser les nœuds d'une manière très efficace en termes d'énergie.



*Figure I.5 : Architecture plate*

### Les réseaux de capteurs hiérarchiques :

Une architecture hiérarchique était proposée pour réduire le coût et la complexité de la plus part des nœuds capteurs en introduisant un ensemble de nœuds capteurs plus coûteux et plus puissant, ceci en créant une infrastructure qui décharge la majorité des nœuds simples a faible coût de plusieurs fonctions du réseau. L'architecture hiérarchique est composée de multiples couches : une couche de capteurs, une couche de transmission et une couche de point d'accès.



*FigureI.6 : Architecture hiérarchique*



### I.3.1.2 Caractéristiques d'un réseau de capteurs sans fil

L'intégration des réseaux de capteurs avec le monde physique a rendu leur mode de fonctionnement différent de celui des réseaux informatiques traditionnels. Ils possèdent des caractéristiques particulières qui rendent le développement d'applications non-trivial

- **Déploiement**

Le déploiement des capteurs est la première opération (phase) dans le cycle de vie d'un réseau de capteurs. On peut envisager plusieurs formes de déploiements selon les besoins des applications. Les nœuds peuvent être déployés aléatoirement d'un avion ou d'une roquette par exemple, ou bien ils peuvent être placés un par un d'une manière déterministe par un humain ou un robot. Le déploiement peut être fait d'un seul coup ou bien peut être un processus continu en redéployant d'autres capteurs dans une même zone. Dans un grand nombre d'applications, le déploiement manuel est impossible. De plus, même lorsque l'application permet un déploiement déterministe, le déploiement aléatoire est adopté dans la majorité des scénarios à cause de raisons pratiques tels que le coût et le temps. Cependant, le déploiement aléatoire ne peut pas fournir une distribution uniforme sur la région d'intérêt, ce qui déclenche de nouveaux problèmes dans les réseaux de capteurs. Les principaux problèmes engendrés sont la localisation, la couverture de la zone, la connexité et la sécurité.

- **Énergie et durée de vie**

La durée de vie est un élément essentiel pour tout réseau de capteurs sans fil, C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier nœud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années. Un nœud capteur peut se trouver dans l'un des quatre états suivants [6] : actif en mode d'écoute, actif en mode de traitement de données, actif en mode de transmission ou non actif en mode veille. Un capteur est en veille lorsque sa radio est éteinte, dans ce cas sa consommation d'énergie est presque nulle. En effet, la principale source de consommation d'énergie d'un capteur est l'utilisation du réseau sans fil via son module de radiocommunications. Cette consommation d'énergie peut être réduite par la diminution de la transmission des données, d'où la nécessité du traitement local.

- **Connectivité**

La plupart des réseaux de capteurs possèdent initialement une densité importante de capteurs, excluant ainsi l'isolement de nœuds. Pourtant, le redéploiement, la mobilité et les défaillances font varier la topologie du réseau, dont la connexité n'est pas toujours assurée.

- **Groupement « clustering »**

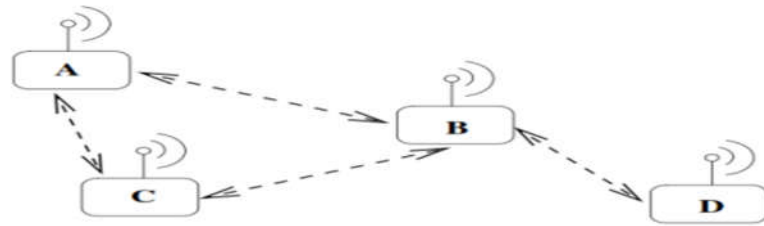
Un réseau de capteur est souvent constitué de plusieurs milliers de nœuds capteurs.

Pour réduire la complexité des algorithmes de routage, faciliter l'agrégation de données, simplifier la gestion du réseau comme l'affectation des adresses, et optimiser la consommation d'énergie, les nœuds sont regroupés dans des clusters. Les nœuds qui sont regroupés ensemble dans un cluster seront capables de communiquer facilement les uns avec les autres. On trouve plusieurs stratégies de groupement parmi lesquelles, les nœuds sont organisés en une hiérarchie en fonction de leur puissance et de leur proximité.

Un chef de cluster est élu pour effectuer plusieurs tâches, comme le filtrage, la fusion et l'agrégation, avec la possibilité d'être changé s'il tombe en panne ou s'il arrive à sa limite d'énergie. Toutes les communications de tous les nœuds seront effectuées par l'intermédiaire du chef du cluster auquel ils appartiennent. Les algorithmes de formation de clusters peuvent être centralisés ou distribués.

- **Communication multi-saut**

Contrairement aux réseaux traditionnels, un réseau de capteurs est constitué d'un grand nombre de nœuds déployés dans une zone locale, ayant une courte portée (rayon de communication), un faible débit et aucune existence d'infrastructure. Un nœud capteur peut communiquer directement avec ses voisins, c'est-à-dire ceux qui sont à sa portée de communication, et fait office de routeur pour les autres nœuds. Par exemple dans la figure I.4, le nœud B pourra relayer les messages du capteur D vers le capteur A. Dans ce cas, les nœuds capteurs communiquent en acheminant les messages par routage « multi-saut ».



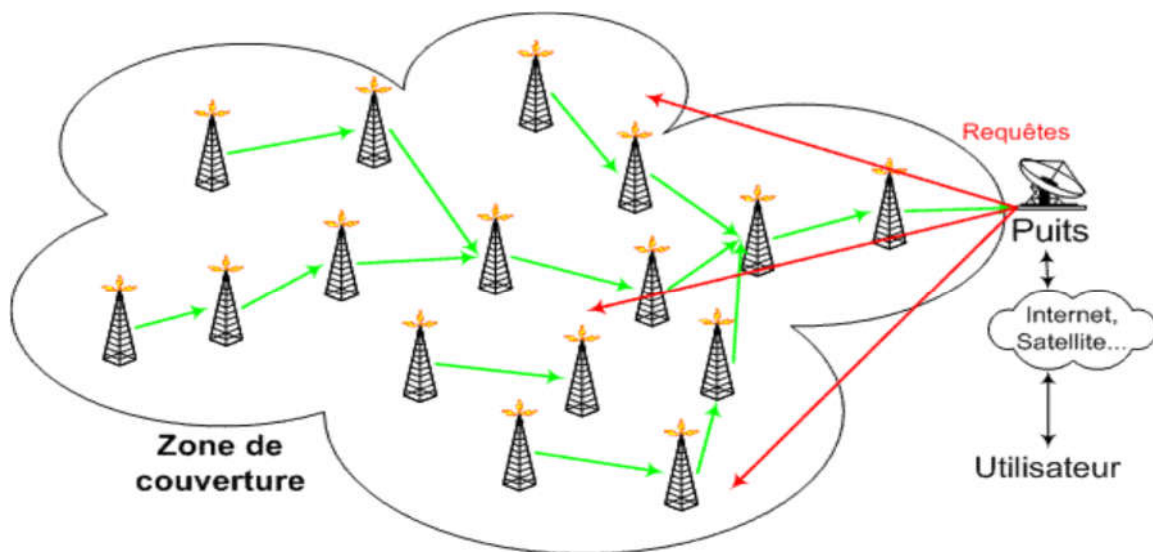
*Figure I.7: Exemple de communication multi-saut dans un réseau de capteurs.*

### I.3.2. Collecter les informations

Il y a deux méthodes pour collecter les informations d'un réseau de capteurs.

#### I. 3.2.1 Collecter les informations à la demande

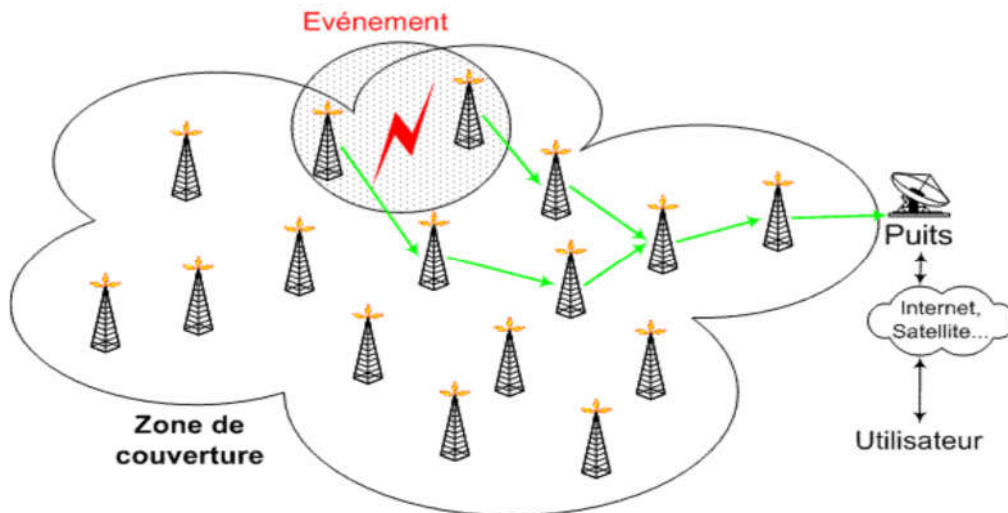
Lorsque l'on souhaite avoir l'état de la zone de couverture à un moment T, le puits émet des broadcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par le biais d'une communication multi-sauts.



*Figure I.8: Collecter les informations à la demande [3].*

#### I.3.2.2 Collecter les informations à suite à un événement

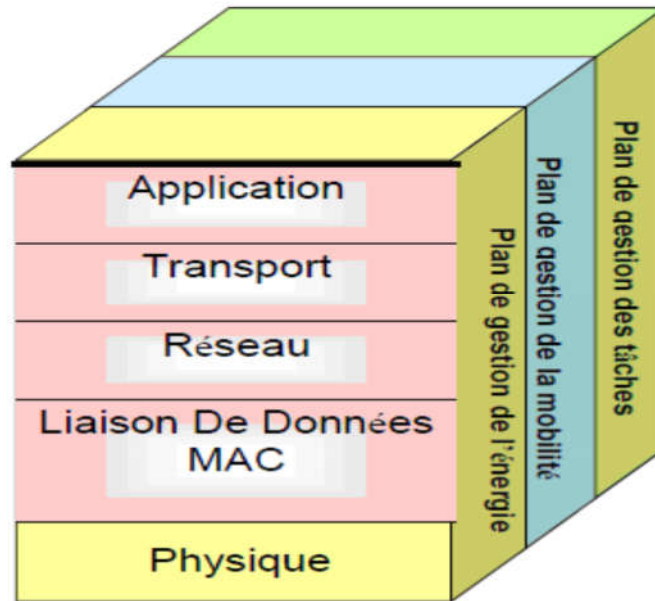
Un événement se produit en un point de la zone de couverture (changement brusque de température, mouvement...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au puits.



*Figure I.9: Collecter les informations Suite à un événement. [3]*

#### I.4. Pile protocolaire

Le rôle de ce modèle consiste à standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce modèle comprend 5 couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que 3 couches pour la gestion de la puissance d'énergie, la gestion de la mobilité ainsi que la gestion des tâches (interrogation du réseau de capteurs). Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur [4].



*Figure I.10 : La pile protocolaire des RCSF [4]*

#### I.4.1 Rôle des couches [4]

- ✍ **La couche physique** : Spécifications des caractéristiques matérielles, des fréquences porteuses, etc.
- ✍ **La couche liaison de données** : Spécifie comment les données sont expédiées entre deux nœuds/routeurs dans une distance d'un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès au media,... Elle assure la liaison point à point et multi-point dans un réseau de communication.
- ✍ **La couche réseau** : Dans la couche réseau le but principal est de trouver une route et une transmission fiable des données, captées, des nœuds capteurs vers le puits "sink" en optimisant l'utilisation de l'énergie des capteurs. Ce routage diffère de celui des réseaux de transmission ad hoc sans fils par les caractéristiques suivantes :
  - il n'est pas possible d'établir un système d'adressage global pour le grand nombre de nœuds.
  - les applications des réseaux de capteurs exigent l'écoulement des données mesurées de sources multiples à un puits particulier.
  - les multiples capteurs peuvent produire de mêmes données à proximité d'un phénomène (redondance).
  - les nœuds capteur exigent ainsi une gestion soignée des ressources. En raison de ces différences, plusieurs nouveaux algorithmes ont été proposés pour le problème de routage dans les réseaux de capteurs.

✍ **La couche transport** : Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

✍ **La couche application** : Cette couche assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

### I.5. Domain d'application

Le concept de réseaux de capteurs sans fil est basé sur une simple équation [5] :

« Capteurs + Processeur + Radio = Une centaine d'applications potentielles »

Les réseaux de capteurs peuvent être programmés à un grand nombre de fins dans des domaines différents, tels que le domaine militaire, scientifique, commercial, industriel, médical, environnement, sécurité, domotique etc, qui sont détaillées dans cette section.

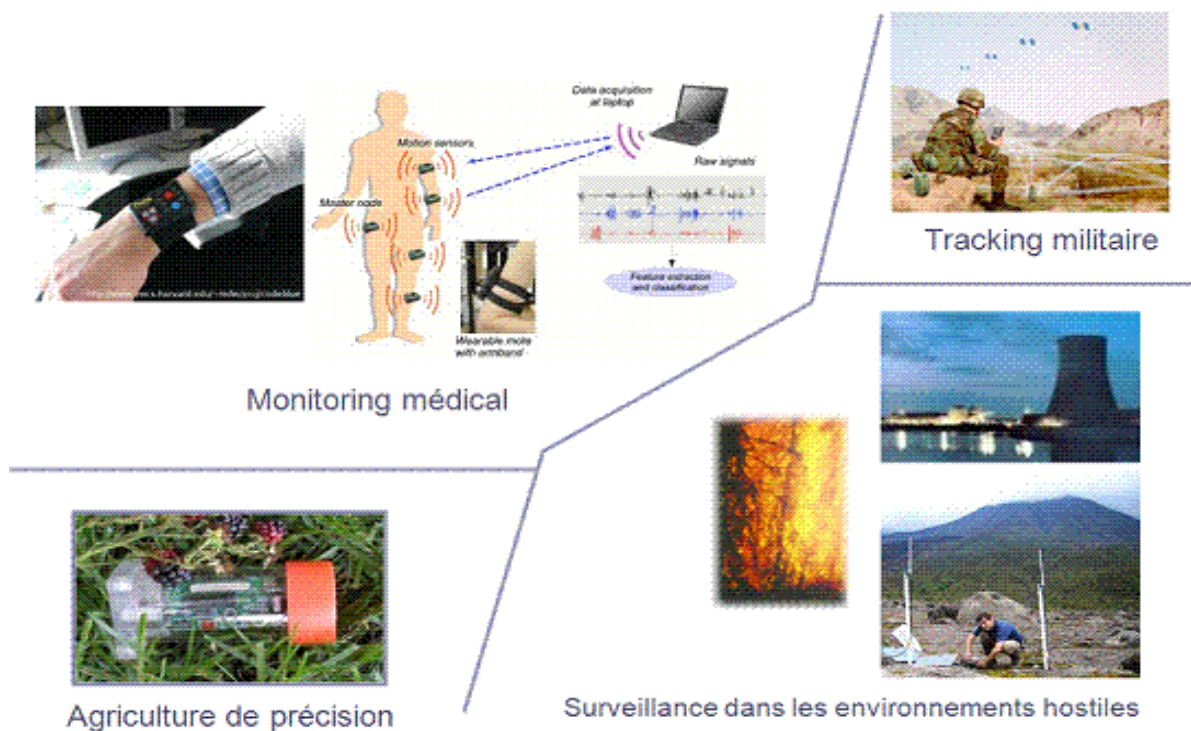


Figure I.11 : Domaines d'application de RCSFs.

#### I.5.1. Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine.

Comme exemple d'application dans ce domaine, on peut penser à un réseau de capteurs déployé sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les



activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations). Des tests concluants ont déjà été réalisés dans ce domaine par l'armée américaine dans le désert de Californie [7].



*Figure I.12 : Un service militaire utilisant les RCSF*

### **I.5.2. Applications liées à la sécurité**

Les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires. Les capteurs doivent s'activer périodiquement et peuvent ainsi fonctionner durant des années, voire des décennies.

Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. La surveillance de voies ferrées pour prévenir des accidents avec des animaux et des êtres humains peut être une application intéressante des réseaux de capteurs. La protection des barrages pourrait être accomplie en y introduisant des capteurs. La détection prompte de fuites d'eau, Permettrait d'éviter des dégâts. Les êtres humains sont conscients des risques et attaques qui les menacent. Pour cela, ils mettent à disposition toutes les ressources humaines et financières nécessaires pour leur sécurité. Cependant, des failles sont toujours présentes dans les mécanismes de sécurisation appliqués aujourd'hui, sans oublier leur coût très élevé. L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant de meilleurs résultats [8].

### I.5.3. Applications environnementales

Des capteurs dispersés à partir d'un avion dans une forêt peuvent signaler un éventuel début d'incendie dans le champ de captage ; ce qui permettra une meilleure efficacité pour la lutte contre les feux de forêt. Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace. Une grande quantité de capteurs peut être déployée dans une forêt ou dans un environnement de conservation de la faune afin de recueillir des informations diverses sur l'état du milieu naturel et sur les comportements de déplacement. Par exemple, l'université de Pise en Italie a réalisé des réseaux de capteurs pour le contrôle des parcs naturels (feux, animaux,...). Il est ainsi possible "d'observer", sans déranger, des espèces animales difficiles à étudier dans leur environnement naturel et de proposer des solutions plus efficaces pour la conservation de la faune. Les éventuelles conséquences de la dispersion en masse des micro-capteurs dans l'environnement ont soulevé plusieurs inquiétudes. En effet, chaque micro-capteur est doté d'une batterie qui contient des métaux nocifs. Néanmoins, le déploiement d'un million de capteurs de 1 mm<sup>3</sup> chacun ne représente qu'un volume total d'un litre. Même si tout ce volume était constitué de batteries, cela n'aurait pas des répercussions désastreuses sur l'environnement [7].

### I.5.4. Applications commerciales

Des nœuds capteurs pourraient améliorer le processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Un client attendant un paquet peut alors avoir un avis de livraison en temps réel et connaître la position du paquet. Des entreprises manufacturières, via des réseaux de capteurs pourraient suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Les produits en fin de vie pourraient être mieux démontés et recyclés ou réutilisés si les micro-capteurs en garantissent le bon état. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles.

Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire [7].



### I.5.5. Applications médicales

Surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité facilitant ainsi le diagnostic de maladies grâce à des micro-capteurs qui pourront être ingérés ou implantés sous la peau.

- (i) les micros-cameras qui peuvent être ingérées et sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain,
- (ii) la création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision de l'œil [7].



*Figure I.13 : Le flux d'information d'un patient*

### I.6. Contraintes dans la conception d'un réseau de capteurs

Les réseaux de capteurs diffèrent des réseaux classiques où l'on peut être relativement générique et définir seulement un certain nombre de classes de service pour satisfaire le maximum de besoins. Ici, les contraintes sont plus nombreuses et empêchent la création d'un type spécifique du réseau de capteurs. Sans être exhaustif, voici une liste de contraintes possibles lors de la conception d'un réseau de capteurs.

### **1.6.1. Contraintes liées à l'application**

Il est impossible aujourd'hui de créer un réseau de capteurs capable de répondre aux besoins de toutes les applications potentielles. On peut relever des mesures pour une infinité de situations et dans des environnements très variables tout en ayant une concentration faible ou forte des capteurs. Dans certains cas, il existe des applications qui nécessitent un grand nombre de capteurs pour être mises en place. La difficulté réside alors dans la recherche d'un dénominateur commun à toutes ces applications ce qui est pour l'instant très complexe et relève de l'impossible. C'est pourquoi, l'application devient le principal paramètre lors de la conception de protocoles très spécifiques pour que le fonctionnement des capteurs produise le résultat attendu par l'application en question.

### **1.6.2. Contrainte énergétique**

L'énergie est considérée comme la contrainte principale dans un réseau de capteurs. Déjà, comme pour tout réseau sans fil, il est important de tenir compte de cette contrainte car la plupart des machines fonctionnent sur batterie. Après la décharge de la batterie, l'utilisateur est obligé de trouver une source électrique pour la recharger.

Cependant, dans les réseaux de capteurs, il est pratiquement impossible de recharger de par le nombre élevé de capteurs déployés et de par la difficulté de l'environnement dans lesquels ils peuvent se trouver. On parle alors pour la pile ou la batterie d'âme du capteur. Une fois vide, le capteur est considéré comme mort ou hors service. L'objectif à atteindre devient l'augmentation de la durée de vie du réseau de capteurs. Ce paramètre peut être défini sous différentes formes telles que la consommation globale de tous les capteurs ou l'évitement qu'un capteur important perde son énergie ou la perte de la connectivité du réseau, etc.

### **1.6.3. Contraintes liées aux déterminismes**

La plupart des réseaux de capteurs sont destinés à être déployés dans des environnements hostiles sur des sites industriels importants ou à opérer pendant des scénarios de crises. L'information que le capteur mesure doit parfois atteindre le collecteur d'informations en un temps borné bien défini. Au-delà de ce temps, l'information est considérée comme périmée ou non existante. Atteindre le déterminisme sur un réseau de capteurs sans fil n'est pas une tâche évidente. La raison vient du fait que pratiquement tous les standards de communication sans fil aujourd'hui utilisent des méthodes probabilistes pour accéder à cette interface radio

**I.6.4. Contraintes de passage à l'échelle**

Le passage à l'échelle (scalability) indique que le réseau est suffisamment large et peut croître de manière illimitée. En d'autres termes, quand on passe à l'échelle, il est trop tard pour effectuer des mises à jour radicales au réseau. À chaque nouvel ajout, on doit prendre en considération les services existants et assurer leur pérennité. De plus, gérer un grand réseau par des humains devient une tâche difficile voire impossible à réaliser. Pour pouvoir opérer quand on passe à l'échelle, il faut que les capteurs soient capables de s'auto configurer seuls. L'auto-configuration peut aller de la simple attribution d'un identifiant jusqu'à l'application du protocole pour le bon fonctionnement du noeud dans son environnement. L'algorithmique distribué est la science la plus adaptée pour résoudre les problèmes du passage à l'échelle.

**I.6.5. Contraintes liées à la qualité de service**

La notion de qualité de service est légèrement différente ici de celle déployée dans les réseaux classiques. Souvent on parle de haut débit ou de faible débit, etc. Ici, avec des petits débits on peut parfois atteindre la qualité exigée. La qualité se définit par la capacité d'interpréter l'information collectée par le puits. Il n'existe donc pas de définition objective de la qualité. En fonction du réseau et du type de mesure, la qualité est alors précisée.

**I.6.6. Contraintes liées à la protection de l'information**

Comme pour tout réseau sans fil, l'information circule sur une interface partagée et non dédiée. N'importe quel intrus peut alors soit récupérer l'information, soit la modifier ou la rendre inexploitable. C'est pourquoi des mesures de sécurité doivent être mise en place pour protéger l'information. Cependant, tous les mécanismes de sécurité sont créés pour des réseaux où les noeuds disposent d'une forte capacité de traitement, ce qui n'est pas le cas des capteurs. À ce jour, très peu de solutions sont adaptées aux capteurs en termes de sécurité.

**I.6.7. Contraintes liées à l'environnement**

Les capteurs interagissent avec l'environnement où ils mesurent leurs grandeurs physiques. De façon générale, ces mesures sont relevées à des instants relativement espacés dans le temps puis soudainement, soit pour des raisons de catastrophe ou d'événement exceptionnel, ils se mettent en mode de forte fréquence de mesures et envoient de l'information en rafale. Il faut alors préparer le réseau à supporter ce type d'événement rare mais largement consommateur de ressources et sujet à des situations de congestions et de difficultés majeures.

**I.6.8. Contraintes de simplicité**

Enfin proposer des protocoles et des mécanismes simples et légers doit être la marque de fabrique du réseau de capteurs. Ces derniers sont de machines largement plus faibles qu'une machine de bureau ou même que des téléphones portables.

**I.7. Conclusion**

Depuis quelques années, les réseaux de capteurs ont suscité un intérêt croissant dans le monde des télécommunications, du traitement de signal et des réseaux sans fil. Les progrès de miniaturisation et d'allongement de durée de vie des batteries, annoncent un futur prometteur à cette technologie. La recherche dans le domaine des capteurs subit actuellement une révolution importante, ouvrant des perspectives d'impacts significatifs dans de nombreux domaines d'applications. Les progrès dans le domaine des communications numériques sans fil nous permettent d'imaginer des réseaux de capteurs totalement autonomes avec des durées de vie importantes. Les réseaux de capteurs souffrent de nombreuses limitations en termes de performance, de fait du manque d'infrastructure et de la nature du medium sans fil. La minimisation d'énergie est en général le vrai problème qui doit être considéré à tous les niveaux de la pile de communication. Un des éléments clés d'une telle optimisation repose sur la mise en place de protocole de gestion.



**Chapitre II :**  
**Les protocoles de**  
**routage dans les RCSF**

## II.1. Introduction

Les protocoles de routage au sein des RCSF sont influencés par un facteur déterminant à savoir la minimisation de la consommation d'énergie sans une perte considérable de l'efficacité. Dans les réseaux de capteurs, chaque nœud joue le rôle de source et de relais. De ce fait, la défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier.

Dans les réseaux sans fil, les protocoles de routage permettent d'établir des routes entre les nœuds pour acheminer les paquets entre eux. Cependant, dans les réseaux de capteurs, les protocoles de routage établissent des routes entre tout nœud du réseau et la station de base pour assurer la fidélité de routage.

Dans ce chapitre on va mettre en évidence quelques protocoles de routages et leurs classifications.

## II.2. Le routage dans les RCSF

La propagation et la délivrance des données dans un RCSF représentent la fonctionnalité la plus importante du réseau. Elle doit prendre en considération toutes les caractéristiques des capteurs afin d'assurer les meilleures performances du système : durée de vie, fiabilité, temps de réponse, ... etc.

Vu la spécificité des RCSF, un nombre important de recherches sont orientées vers une violation du découpage en couches protocolaires indépendantes, et introduisent la notion de cross layer optimisation. Par exemple, en utilisant des mécanismes d'agrégation, les routeurs intermédiaires doivent accéder à la donnée afin d'établir des résumés des lectures de la région[4].

### II.2.1. Les type de routage

#### II.2.1.1 Protocole proactif

Protocole proactif est un protocole qui construit les tables de routage avant que la demande en soit effectuée. Il identifie en fait à chaque instant la topologie du réseau[9].

### II.2.1.2. Protocole réactif

Protocole réactif est un protocole qui construit une table de routage lorsqu'un nœud en effectue la demande. Il ne connaît pas la topologie du réseau, il détermine le chemin à prendre pour accéder à un nœud du réseau lorsqu'on lui demande[13].

## II.2.2. Les protocoles de routage

Dans les réseaux de capteur sans fil, nous distinguons trois catégories de protocoles de routages : protocole de routage non hiérarchiques (ou à plat), hiérarchiques et géographique.

### II.2.2.1 Les protocoles de routage à plat (non hiérarchiques)

L'objectif principal d'un protocole de routage pour un réseau Ad-hoc est l'établissement correct et efficace d'itinéraires entre une paire de nœuds afin que des messages puissent être acheminés. Le protocole de routage permet aux nœuds de se connecter directement les uns aux autres pour envoyer les messages par des sauts multiples. Par la suite, nous présentons un état de l'art des principaux protocoles de routage à plat (non hiérarchique) dans les réseaux Ad-hoc car la présentation de ces protocoles nous permettra de mieux analyser l'avantage de l'approche hiérarchique surtout dans les grands réseaux. Malgré que notre intérêt se focalise sur les protocoles hiérarchiques, nous avons rédigé cet état de l'art parce qu'un réseau de capteurs partage forcément des caractéristiques et des contraintes des réseaux Ad-hoc qu'il faut prendre en compte lors d'une proposition d'un protocole de routage [9].

#### ➤ DSDV (Destination Sequenced Distance Vector)

DSDV[15] est un protocole proactif de routage à vecteur de distance. Chaque nœud du réseau maintient une table de routage contenant le saut suivant et le nombre de sauts pour toutes les destinations possibles. Des diffusions de mises à jour périodiques tendent à maintenir la table de routage complètement actualisée à tout moment.

Afin d'éviter le bouclage (loop-freedom), DSDV utilise les numéros de séquence (Sequence Number) pour indiquer la « nouveauté » d'une route. Une route R est considérée plus favorable qu'une autre R', si R a un numéro de séquence plus grand ; si ces deux routes ont le même numéro de séquence, alors R est plus favorable s'il possède un nombre inférieur de sauts. Le numéro de séquence pour une route est initialisé par le

nœud émetteur et incrémenté pour chaque nouvel avertissement de route. Quand un nœud détecte un lien brisé vers une destination D, il met à jour le nombre de sauts pour l'entrée de la destination D dans sa table avec la valeur infini et incrémente son numéro de séquence .

Les boucles de routes peuvent survenir lorsque des informations incorrectes de routage sont présentes dans le réseau après un changement dans la topologie du réseau (lien brisé par exemple). Dans ce contexte, l'utilisation des numéros de séquence adapte DSDV à une topologie dynamique de réseau comme dans un réseau Ad-hoc .

DSDV utilise des mises à jour étiquetées lorsque la topologie change. La transmission des mises à jour est retardée afin d'introduire un effet d'amortissement quand la topologie change rapidement.

#### ➤ **OLSR (Optimized Link State Routing)**

Comme son nom l'indique, OLSR est un protocole proactif à état des liens optimisé ; il permet d'obtenir aussi des routes de plus court chemin. Alors que dans un protocole à état des liens, chaque nœud déclare ses liens directs avec ses voisins à tout le réseau, dans le cas d'OLSR, les nœuds ne déclarent qu'une sous-partie de leur voisinage grâce à la technique des relais multipoints (Multi PointRelaying, MPR) [11] décrite par la suite.

##### **a. Relais multipoints**

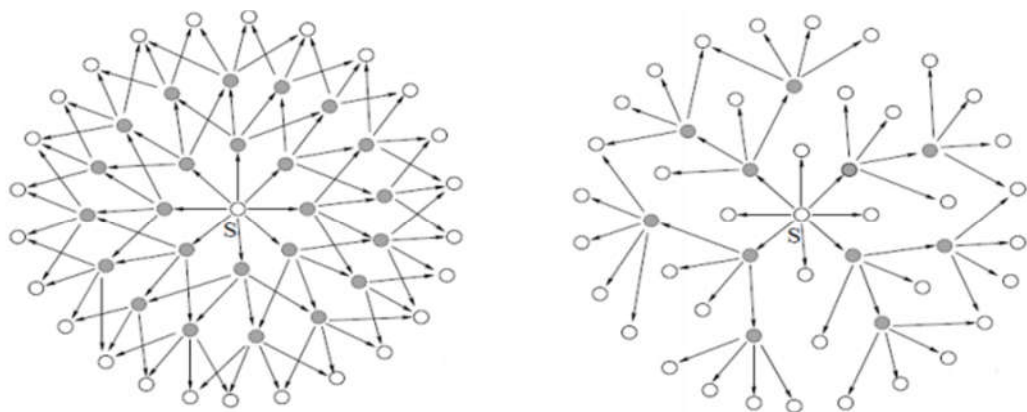
Cette technique consiste essentiellement, pour un nœud donné, à ignorer un ensemble de liens et de voisins directs, qui sont redondants pour le calcul des routes de plus court chemin. Plus précisément, dans l'ensemble des voisins d'un nœud, seul un sous-ensemble de ses voisins est considéré comme pertinent. Il est choisi de façon à pouvoir atteindre tout le voisinage à deux sauts (tous les voisins des voisins) ; cet ensemble est appelé l'ensemble des relais multipoints. L'algorithme de calcul de relais multipoints est donné dans [11].

Ces relais multipoints sont utilisés de deux façons : pour diminuer le trafic dû à la diffusion des messages de contrôle dans le réseau, et aussi pour diminuer la taille du sous ensemble des liens diffusés à tout le réseau puisque les routes sont construites à base des relais multipoints.



L'idée de MPR est de minimiser l'inondation du trafic de contrôle dans un réseau en réduisant les retransmissions dupliquées dans la même région. Chaque nœud dans le réseau sélectionne un ensemble de nœuds de son voisinage auxquels ses messages seront transmis. Un nœud sélectionne ses MPRs parmi ses voisins à un saut avec un lien symétrique. Cet ensemble est choisi de manière à couvrir tous les nœuds qui sont à deux sauts. Les nœuds sélectionnés comme MPRs annoncent régulièrement leur condition de MPR dans les messages de contrôle envoyés à son voisinage. De cette façon, un nœud annonce au réseau qu'il est capable d'atteindre les nœuds qui l'ont élu comme MPR. Dans le calcul de la route, les MPRs sont utilisés pour la mise en place des routes vers toutes les destinations du réseau. Ainsi, en sélectionnant la route par l'intermédiaire des MPRs, on évite les problèmes liés à la transmission de paquets sur des liens unidirectionnels. Chaque nœud maintient l'information sur ses voisins qui ont été sélectionnés comme MPR. Un nœud obtient cette information par les messages de contrôle reçus périodiquement de ses voisins [11].

La Figure II.1 montre la différence entre une diffusion pure et la diffusion en utilisant les MPRs. Par exemple, afin d'atteindre tous les nœuds à 3 sauts, la diffusion pure a besoin de 24 retransmissions du paquet envoyé par la source (voir la Figure II.1, côté gauche). En utilisant les MPRs ou les relais multipoints (nœuds en gris dans le côté droit de la Figure II.1), il suffit de retransmettre le paquet de la source 11 fois pour atteindre les nœuds à 3 sauts.



**Figure II.1** : Diffusion pure et diffusion en utilisant les MPRs dans OLSR

**b. Fonctionnement du protocole**

OLSR est un protocole proactif et très bien adapté aux réseaux larges et denses. Tous les nœuds du réseau serviront de routeur et OLSR maintient sur chaque nœud une table de routage complète. Le protocole est complètement distribué, il n'y a pas d'entité centrale. Chaque nœud choisit la route la plus adaptée en fonction des informations qu'il a reçues.

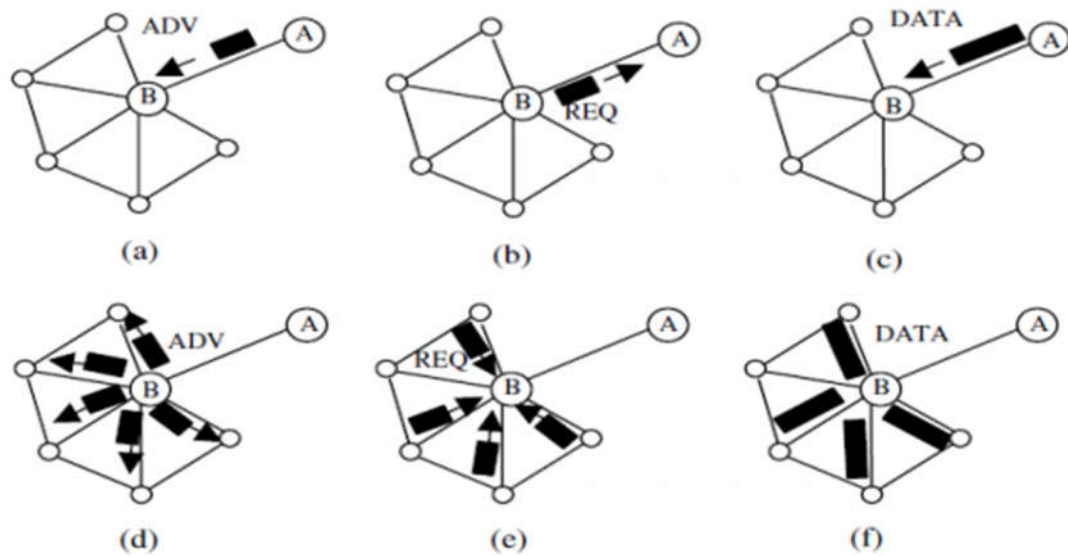
Pour maintenir à jour toutes les informations nécessaires au choix des relais multipoints (MPRs) et au calcul de la table de routage, les nœuds OLSR ont besoin de s'échanger des informations périodiquement. Pour s'informer du proche voisinage, les nœuds OLSR envoient périodiquement des messages dits HELLO contenant la liste de leurs voisins. Ces messages permettent à chacun de choisir son ensemble de relais multipoints[11].

Le deuxième type de message d'OLSR est le message TC (Topology Control). Par ce message les sous-ensembles de voisinage que constituent les relais multipoints sont déclarés périodiquement dans le réseau. Ils sont diffusés en utilisant une diffusion optimisée par relais multipoints. Ces informations offrent une carte du réseau contenant tous les nœuds et un ensemble partiel de liens, mais suffisant pour la construction de la table de routage.

La table de routage est calculée par chacun des nœuds et le routage des données s'effectue saut par saut sans l'intervention d'OLSR dont le rôle s'arrête à la mise à jour de la table de routage [11].

**➤ SPIN (Sensor Protocol for Information via Negotiation)**

Le protocole SPIN [12] permet de disséminer des informations sur le réseau de manière ciblée. Le fonctionnement du protocole SPIN permet de réduire la charge du réseau par rapport aux méthodes de diffusion traditionnelles telles que l'inondation ou l'algorithme de Gossiping.



**Figure II.2:** Fonctionnement du protocole SPIN [12].

Le protocole SPIN utilise essentiellement trois types de paquets ADV/REQ/DATA. Un nœud voulant émettre une donnée commence par envoyer un paquet ADV. Ce paquet ADV consiste d'une méta-donnée sur les données à émettre. Les méta-données peuvent décrire plusieurs aspects comme le type des données et la localisation de son origine. Les nœuds qui reçoivent ce paquet vérifient si les données les intéressent. Si oui, ils répondent par un paquet REQ. Le nœud qui a initié la communication envoie alors un paquet DATA pour chaque réponse REQ reçue (voir la Figure II.2). Un nœud peut parfaitement ne pas répondre aux messages ADV, par exemple dans le but d'économiser son énergie. Ensuite chaque nœud qui fait office de relais peut très bien agréger ses propres données aux données qui sont déjà contenues dans le paquet

### II.2.2.2. Protocoles de routage hiérarchiques

Lorsque la taille du réseau devient de plus en plus importante, sa gestion devient plus difficile. Les protocoles de routage à plat fonctionnent bien quand le réseau ne comprend pas un grand nombre de nœuds. La structuration d'un réseau est un des outils principaux pour sauvegarder l'énergie dans chaque nœud du réseau, ce qui aboutit à prolonger la vie du système. Une des structures les plus connues est la hiérarchie. La technique de hiérarchisation sert à partitionner le réseau en sous ensembles afin de faciliter la gestion du réseau surtout le routage, qui se réalise à plusieurs niveaux. Dans ce type de

protocoles, la vue du réseau devient locale ; des nœuds spéciaux peuvent avoir des rôles supplémentaires. La littérature comprend plusieurs contributions dans les techniques de hiérarchisation du réseau, Ce que nous avons classifié à l'état de l'art suivant, dans l'intérêt de faciliter leur comparaison.

Un cluster est défini par un ensemble de nœuds et possède un nœud nommé nœud-chef ou Cluster Head (CH). Le rôle du CH est de faire le relais entre les nœuds du cluster et la station de base directement ou via d'autres CHs. Le CH possède généralement des ressources énergétiques supérieures aux autres nœuds du réseau. Cette technique est appelée cautérisation (voir la Figure II.5). Le CH est élu suivant différents critères et informations sur le réseau : le niveau de l'énergie d'un capteur, la connexion avec les autres capteurs, la position géographique, etc. Une zone est définie par un ensemble de nœuds mais ne possède pas un nœud-chef (ou CH). Ainsi, un cluster est une sous-classe d'une zone.

La construction des groupes (zones ou clusters) s'appuie sur des informations sur le réseau, exigeant donc son instrumentation. Cette prise de mesures peut être, dans certaines circonstances, statique (comme la position des capteurs dans un système immobile) ou dynamique (comme le niveau énergétique des capteurs).

Une autre structure utilisée est la chaîne [13]. Le principe d'une chaîne est qu'un nœud ne peut communiquer qu'avec deux voisins. On a trouvé aussi des structures qui combinent les groupes et les chaînes. En se basant sur une architecture hiérarchique, plusieurs protocoles de routage pour les réseaux Ad-hoc de grande taille ont été proposés. Dans la suite nous en détaillerons quelques uns.

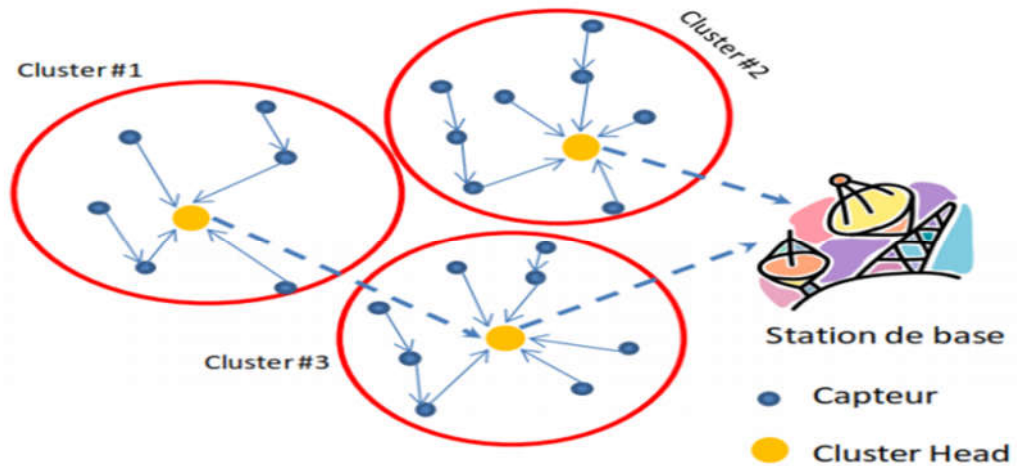


Figure II.3 : Architecture en cluster

➤ **Le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy)**

Le protocole LEACH est le plus populaire des protocoles de routage hiérarchique, proposé par Heinzelman et al. (2000) [14] pour former des clusters en se basant sur l'intensité du signal radio reçu. En effet, LEACH utilise un algorithme distribué où chaque nœud décide d'une manière autonome s'il sera ClusterHead ou non en calculant aléatoirement une probabilité  $p$  et en la comparant à un seuil  $T(u)$ ; puis, il informe son voisinage de sa décision. Chaque nœud non ClusterHead décide du cluster à joindre en utilisant un minimum d'énergie de transmission (i.e. le plus proche). L'algorithme se déroule en plusieurs rounds et pour chaque round, une rotation du rôle du ClusterHead est initiée selon la probabilité «  $p$  » choisie et comparé à la formule suivante du seuil:

$$T(u) = \begin{cases} \frac{p}{1 - p * (r * \text{mod}(\frac{1}{p}))} & \text{si } u \in G \\ 0 & \text{sinon} \end{cases} \quad (\text{II.1})$$

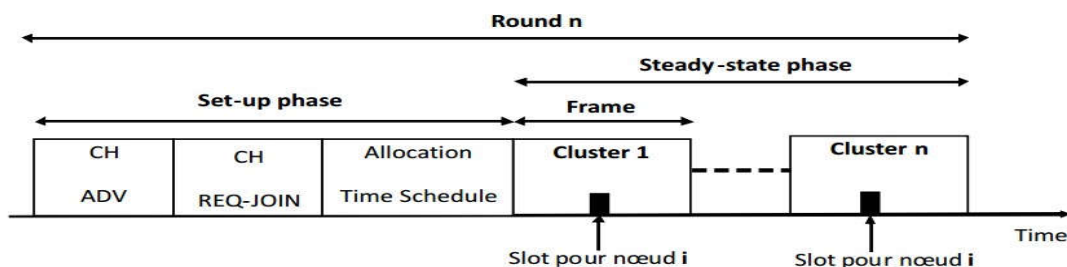
Où  $p$ : le pourcentage des CHs sur le réseau (généralement 5%);

$r$ : numéro du round en cours;

$G$ : l'ensemble des noeuds qui n'était pas CH dans les  $(1/p)$  rounds précédentes.

LEACH suppose que chaque nœud du réseau peut communiquer directement avec le puits; alors que, les nœuds non-ClusterHead ne peuvent communiquer qu'avec leurs ClusterHead choisi, en utilisant la technique TDMA instaurée par ce dernier. Cette technique permet de minimiser les collisions en allouant à chaque nœud un temps privé pour transmettre ses données vers son CH.

LEACH est exécuté en deux phases : la phase « set-up » et la phase « steady-state » suivant la *Figure II.8*. Dans la première phase, les clusters heads sont sélectionnés et les clusters sont formés, et dans la seconde phase, le transfert de données vers la station de base aura lieu. Durant la première phase, le processus d'élection des clusters heads est déclenché pour choisir les futurs clusters heads. Ainsi, une fraction prédéterminée de nœuds s'élisent comme cluster heads selon le schéma d'exécution suivant : durant une période  $T$ , un nœud  $n$  choisit un nombre aléatoire  $nb$  dont la valeur est comprise entre 0 et 1 ( $0 < nb < 1$ ). Si  $nb$  est inférieure à une valeur seuil alors le nœud  $n$  deviendra cluster head durant la période courante, sinon le nœud  $n$  devrait rejoindre le cluster head le plus proche dans son voisinage.



**Figure II.4:** Répartition du temps et différentes phases pour chaque round

LEACH préconise, également, une agrégation de données au niveau des CHs pour plus de conservation d'énergie. Cependant, plusieurs critiques sont apportées au protocole LEACH relatives à ses hypothèses contraignantes de départ, à savoir:

- La possibilité de communiquer avec le puits à travers n'importe quel nœud du réseau exige une consommation d'énergie importante des nœuds lointains. Ce qui rend le protocole moins apte au passage à l'échelle,
- L'agrégation des données est centrée au niveau des CHs, ce qui les rend les maillons faibles du réseau;
- La rotation du rôle du CH sur l'ensemble des nœuds du cluster, permet d'une

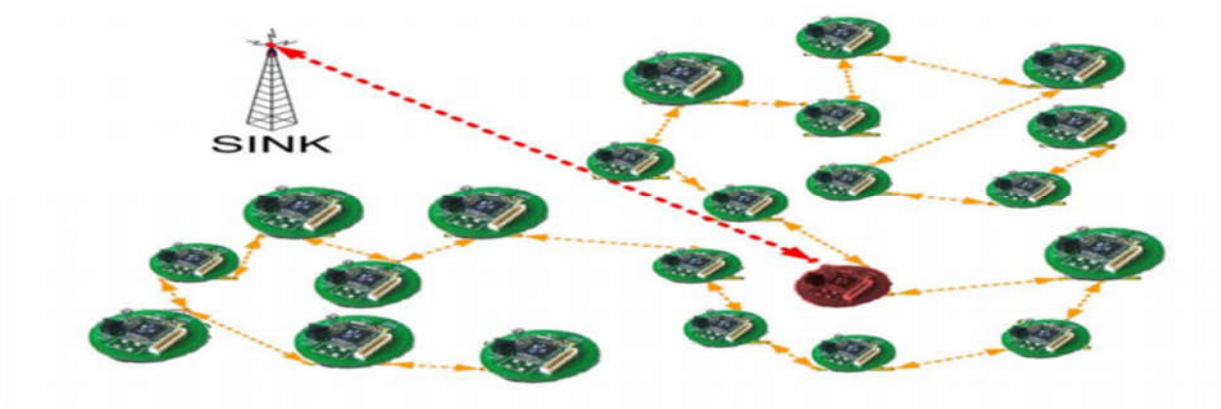
part d'équilibrer la consommation de l'énergie du cluster. Mais, elle génère une surconsommation d'énergie, car chaque rotation de CH nécessite une phase de diffusion pour faire connaître le nouveau CH;

- LEACH ne garantit pas une distribution homogène des CHs sur le réseau, car le seul critère d'élection du CH est une probabilité aléatoire. Cela n'empêche pas une concentration des CHs dans une région limitée au détriment de l'ensemble du réseau. Plusieurs variantes de LEACH ont été proposées pour palier aux problèmes de la version originale, à savoir, LEACH-C qui est une version centralisée [15], où l'algorithme s'exécute au niveau du puits pour permettre une meilleure distribution des CHs sur réseau .

➤ **Le protocole PEGASIS (Power-Efficient GATHERing in Sensor Information Systems)**

Il est considéré comme une optimisation de LEACH, proposé par Lindsey et autres en 2002, PEGASIS regroupe les nœuds du réseau sous forme d'une longue chaîne en se basant sur le principe qui stipule qu'un nœud ne peut communiquer qu'avec le nœud le plus proche de lui. Ainsi, il ajuste sa radio pour une communication très courte pour conserver son.

Énergie. Pour communiquer avec le puits, le processus est organisé en rounds; au cours de chaque round un seul nœud est autorisé à communiquer avec le puits directement. Ce privilège est accordé à l'ensemble des nœuds du réseau à tour de rôle. Une meilleure conservation d'énergie est obtenue, également, en agrégeant les données sur chaque nœud du réseau[13].



**Figure II.5:** Formation des chaînes gourmandes dans PEGASIS

**Algorithme de la chaîne Greedy**

- ✍ Commencez avec noeud le plus éloigné de BS
- ✍ Ajouter à chaîne voisin le plus proche de ce noeud qui n'a pas été visitée
- ✍ Répétez jusqu'à ce que tous les noeuds ont été ajoutés à la chaîne
- ✍ Construits avant le 1er tour de la communication, puis reconstruite lorsque les noeuds meurent

**La fusion de données à chaque noeud (sauf noeuds d'extrémité)**

- Un seul message est transmis à chaque nœud Calcul Delay:  $N$  unités pour un noeud réseau  $N$
- Transmission séquentielle est supposée Noeud  $i$  (mode  $N$ ) est le chef de file dans  $i$  ronde

La simulation montre que PEGASIS garantit une durée de vie de réseau deux fois plus importante que LEACH [14]. Cette performance est obtenue en utilisant l'agrégation des données qui minimise le nombre des transmissions, et en éliminant la phase de la construction des clusters pour chaque round qui génère une surconsommation d'énergie importante [13].

Cependant, dans PEGASIS chaque nœud nécessite une connaissance actualisée de la topologie pour s'informer de l'état énergétique de ses voisins afin de router ses données efficacement. Cela génère une surconsommation d'énergie importante surtout pour les réseaux à grande échelle. En plus, PEGASIS stipule que tous les nœuds du réseau peuvent atteindre le puits ce que nécessite une transmission réglable avec un surcoût énergétique non négligeable.

En plus, le délai de livraison des données est très important lorsque la chaîne formée est très longue. Et, le nœud qui transmet les données vers le puits peut devenir un point de congestion du réseau.

Une amélioration de PEGASIS dite H-PEGASIS a tenté de résoudre le problème de délai de livraison de données en adoptant des communications parallèles avec le puits pour les nœuds géographiquement distants entre eux.



➤ **TEEN (Threshold-sensitive Energy Efficient sensor Network protocol)**

Manjeshwar et Agrawal [15] ont proposé une technique de clustering appelée TEEN pour les applications critiques où le changement de certains paramètres peut être brusque. L'architecture du réseau est basée sur un groupement hiérarchique à plusieurs niveaux où les nœuds les plus proches forment des clusters. Puis ce processus de clustering passe au deuxième niveau jusqu'à ce que la station de base soit atteinte. Après la formation des clusters, chaque cluster-head transmet à ses membres deux seuils : un seuil Hard HT (hard threshold), qui est la valeur seuil du paramètre contrôlé (surveillé) et un seuil Soft ST (soft threshold) représentant une petite variation de la valeur du paramètre contrôlé. L'occurrence de cette petite variation ST permet au nœud qui la détecte de la signaler à la station de base en transmettant un message d'alerte. Par conséquent, le seuil Soft réduira le nombre de transmissions puisqu'il ne permet pas la transmission s'il y a peu ou pas de variation de la valeur du paramètre contrôlé.

Au début, les nœuds écoutent le médium continuellement et lorsque la valeur captée du paramètre contrôlé dépasse le seuil Hard, le nœud transmet l'information. La valeur captée est stockée dans une variable interne appelée SV. Les nœuds ne transmettront des données que si la valeur courante du paramètre contrôlé est supérieure au seuil hard HT ou diffère du SV d'une quantité égale ou plus grande que la valeur du seuil Soft ST. Puisque la transmission d'un message consomme plus d'énergie que la détection des données, alors la consommation d'énergie dans TEEN est moins importante que dans les protocoles proactifs ou ceux qui transmettent des données périodiquement tels que LEACH.

Cependant, l'inconvénient principal de ce protocole est que, si les seuils HT et ST ne sont pas reçus, les nœuds ne communiqueront jamais, et aucune donnée ne sera transmise à l'utilisateur, ainsi la station de base ne connaît pas les nœuds qui ont épuisé leur énergie. TEEN ne convient pas aux applications qui nécessitent des envois périodiques de données.

➤ **APTEEN (Adaptive Threshold-sensitive Energy Efficient sensor Network protocol)**

Pour remédier aux limitations du protocole TEEN, les auteurs ont proposé une extension de TEEN appelée APTEEN [16]. APTEEN est un protocole hybride qui change

la périodicité et les valeurs seuils utilisées dans TEEN selon les besoins de l'utilisateur et le type d'application. Dans APTEEN, les cluster-heads transmettent à leurs membres les paramètres suivants :

- l'ensemble de paramètres physiques auxquels l'utilisateur est intéressé pour obtenir des informations (A),
- les seuils : seuil Hard HT et seuil Soft ST,
- un Schedule TDMA permettant d'assigner à chaque nœud un intervalle fini de temps appelé slot,
- un compteur de temps (CT) : c'est la période de temps maximum entre deux transmissions successives d'un nœud.

Dans APTEEN[16], les nœuds surveillent en continu l'environnement. Ainsi, les nœuds qui détectent une valeur d'un paramètre qui dépasse le seuil HT, transmettent leurs données. Une fois qu'un nœud détecte une valeur qui dépasse HT, il ne transmet les données au cluster head que si la valeur de ce paramètre change d'une quantité égale ou supérieure à ST. Si un nœud ne transmet pas de données pendant une période de temps CT, il devrait faire une capture de données et les retransmettre. APTEEN offre une grande flexibilité qui permet à l'utilisateur de choisir l'intervalle de temps CT, et les valeurs seuils HT et ST pour que la consommation d'énergie soit contrôlée par la variation de ces paramètres. Cependant, APTEEN nécessite une complexité supplémentaire pour implémenter les fonctions de seuils et de périodes de temps CT. Ainsi, le surcoût et la complexité associés à la formation des clusters à plusieurs niveaux par TEEN et APTEEN sont assez élevés.

➤ **HEED (Hybrid, Eenergy-Efficient, Distributed approach)**

Younes et Fahmy[17] ont proposé un algorithme de clustering distribué appelé HEED pour les réseaux de capteurs. Contrairement aux techniques précédentes, HEED ne fait aucune restriction sur la distribution et la densité des nœuds. Il ne dépend pas de la topologie du réseau ni de sa taille mais il suppose que les capteurs ont la possibilité de modifier leur puissance de transmission. HEED sélectionne les cluster-heads selon un critère hybride regroupant l'énergie restante des nœuds et un second paramètre tel que le degré des nœuds. Il vise à réaliser une distribution uniforme des clusterheads dans le

réseau et à générer des clusters équilibrés en taille. Un nœud  $u$  est élu comme cluster head avec une probabilité  $P_{ch}$  égale à :

$$P_{ch} = C_{prob}(E_n/E_{total}) \quad (II.2)$$

Où  $E_n$  est l'énergie restante du nœud  $n$ ,  $E_{total}$  est l'énergie initial dans le réseau et  $C_{prob}$  est le le nombre optimal de clusters. Cependant, l'évaluation de  $E_{total}$  présente une certaine difficulté, à cause de l'absence de toute commande centrale. Un autre problème réside dans la détermination du nombre optimal de clusters. De plus, HEED ne précise pas de protocole particulier à utiliser pour la communication entre les clusters heads et le sink. A l'intérieur du cluster, le problème ne se pose pas car la communication entre les membres du cluster et le cluster head est directe (à un saut). D'autre part, avec HEED, la topologie en clusters ne réalise pas de consommation minimale d'énergie dans les communications intra cluster et les clusters générés ne sont pas équilibrés en taille.

### II.2.2.3. Les protocoles de routage géographique

Dans ce type de routage, les nœuds capteurs sont adressés en fonction de leurs localisations. La distance entre les nœuds voisins peut être estimée sur la base des forces entrantes de signal. Des coordonnées relatives des nœuds voisins peuvent être obtenues en échangeant une telle information entre les voisins. Alternativement, la location des nœuds peut être disponible directement en communiquant avec un satellite en utilisant GPS (système de positionnement global).

Dans la plupart des protocoles de routage, l'information sur la localisation des nœuds est nécessaire afin de calculer la distance entre deux nœuds particuliers de sorte que la consommation d'énergie puisse être estimée. On peut mentionner parmi ces protocoles :

#### ➤ GAF

GAF (Geographic adaptive fidelity) conserve l'énergie par la mise en veille des nœuds inutiles dans le réseau sans affecter le niveau de fidélité du routage. Il forme une grille virtuelle pour le domaine couvert. Chaque nœud emploie sa position indiquée par le GPS pour s'associer à un point dans la grille virtuelle. Des nœuds liés au même point sur la grille sont considérés équivalents en terme de coût de routage. Une telle équivalence est exploitée en maintenant quelques nœuds, situés dans un secteur particulier de la grille, dans l'état de sommeil afin d'économiser de l'énergie.

➤ **GEAR (geographic and energy-aware routing)**

L'idée est de restreindre le nombre d'intérêts dans la diffusion dirigée en considérant seulement certaines régions plutôt que d'envoyer les intérêts au réseau tout entier.

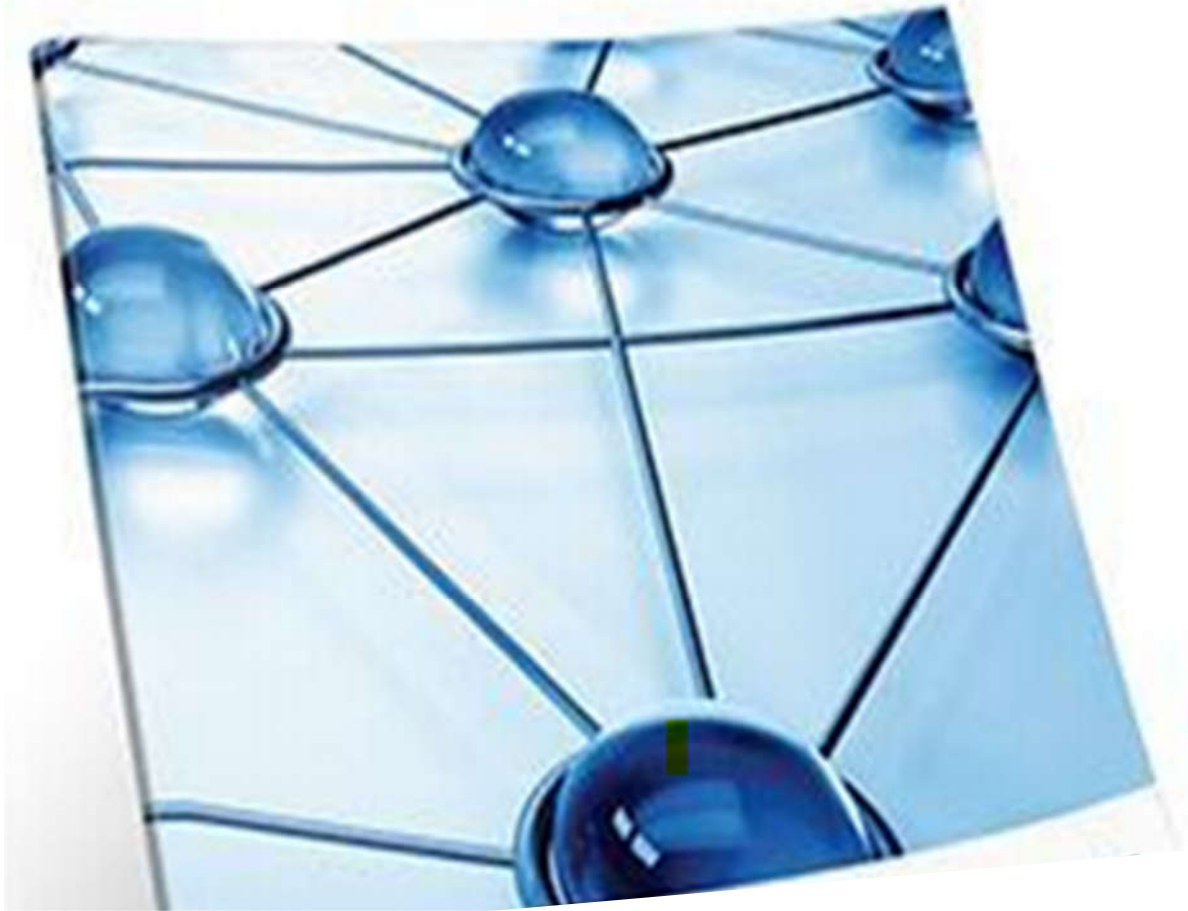
Le routage géographique suppose que tous les nœuds connaissent leur position. Une solution basée sur le GPS peut être trop coûteuse, d'autant plus que le nombre de nœuds à équiper est très grand [18].

➤ **MECN et SMECN**

MECN (Minimum energy communication network) utilise le GPS à basse puissance. L'idée principale est de trouver un sous réseau, qui aura moins de nœuds et qui exige moins de puissance pour la transmission entre deux nœuds particuliers quelconques. SMECN (Small MECN) est une extension de MECN. Le sous- réseau construit par SMECN est probablement plus petit (en terme de nombre d'arcs) que celui construit par MECN[19].

### **II.3. Conclusion**

Dans les réseaux sans fil, plusieurs protocoles de routage ont été conçus et ces protocoles donnent de bons résultats en termes de temps nécessaire pour envoyer une information à la station de base et d'énergie dissipée pour les réseaux de taille modeste. Cependant, ces protocoles perdent leurs performances lorsque nous passons à l'échelle puisqu'ils utilisent des versions itératives pour établir les routes entre les nœuds et la station de base.



**Chapitre III :**  
**Algorithme des Lucioles**  
**(Firefly)**

### III.1. Introduction

Les méta-heuristiques bio-inspirées sont devenues de plus en plus intéressantes grâce à leur efficacité vis-à-vis de la résolution des problèmes de l'optimisation combinatoire.

Parmi les algorithmes bio-inspirés les plus connus on trouve les algorithmes évolutifs, inspirés de l'évolution biologique.

L'algorithme des Lucioles ou algorithme Firefly est un algorithme évolutif. Dans ce chapitre nous allons présenter ce méta heuristique bio-inspirée. Pour cela, nous allons commencer par introduire le coté biologique de l'algorithme. Nous allons par la suite décrire le mécanisme de l'algorithme qui constitue d'un point de vue informatique une source d'inspiration assez riche.

### III.2 Algorithme des Lucioles (FireFlyAlgorithm)

#### III.2.1. Inspiration

Les lucioles (en anglais Firefly) sont de petits coléoptères ailés capables de produire une lumière clignotante froide pour une attraction mutuelle.

Dans le langage courant entre les lucioles, ils sont également utilisés synonymes bogues d'éclairage ou des vers luisants. Ce sont deux coléoptères qui peuvent émettre de la lumière, mais les lucioles sont reconnues comme des espèces qui ont la capacité de voler.

Ces insectes sont capables de produire de la lumière à l'intérieur de leur corps grâce à des organes spéciaux situés très près de la surface de la peau. Cette production de lumière est due à un type de réaction chimique appelée bioluminescence .

Les femelles peuvent imiter les signaux lumineux des autres espèces afin d'attirer des mâles qu'elles les capturent et les dévorent. Les lucioles ont un mécanisme de type condensateur, qui se décharge lentement jusqu'à ce que certain seuil est atteint, ils libèrent l'énergie sous forme de lumière. Le phénomène se répète de façon cyclique [20].



**Figure III.1 : Le Luciole**

### III.2.2. Principe de fonctionnement de l'algorithme des Lucioles

L'algorithme des Lucioles est un méta heuristique, bio-inspirée, introduite par Dr Xin-She Yan à l'université Cambridge en 2007. L'algorithme est basé sur le principe d'attraction entre les lucioles et simule le comportement d'un essaim de lucioles dans la nature. Cet algorithme prend en considération les trois règles suivant :

1. Toutes les lucioles sont unisexes, ce qui fait l'attraction entre celles-ci n'est pas en fonction de leur sexe;
  2. L'attraction est proportionnelle à leurs luminosités, donc pour deux lucioles, la moins lumineuse se déplacera vers la plus lumineuse. Si aucune luciole n'est lumineuse qu'une luciole particulière, cette dernière se déplacera aléatoirement;
  3. La luminosité des lucioles est déterminée en fonction d'une fonction objective (à optimiser)[21].
- Le processus de l'algorithme de luciole commence avec l'initialisation de la population des lucioles et donc chaque luciole dans une population représente une solution candidate.
  - La taille de la population détermine le nombre de solutions ou la taille de l'espace de recherche dont le but est d'orienter la recherche à la meilleure localisation[22].
  - Dans l'étape suivante, chaque luciole est évaluée en fonction de leur condition physique (intensité lumineuse). À chaque nouvelle étape itérative, la luminosité et l'attraction de chaque luciole est calculée.
  - La fonction de l'attractivité est définie en utilisant l'intensité lumineuse, la distance entre lucioles, et un coefficient d'absorption .
  - Le mouvement de luciole est défini par une fonction de mouvement, en utilisant la position actuelle, l'attractivité et une marche aléatoire, après avoir comparé la luminosité de chaque luciole avec toutes les autres lucioles, les positions des lucioles sont mises à jour en se basant sur les règles de connaissances sur les lucioles et sur leurs voisins.
  - Après le déplacement, la nouvelle luciole est évaluée et l'intensité de sa lumière est mise à jour. Pendant la boucle de comparaison en paire, la meilleure solution actuelle est la mise à jour d'une manière itérative. Le Processus de comparaison par pair est répété jusqu'à la satisfaction des critères de résiliation [23].

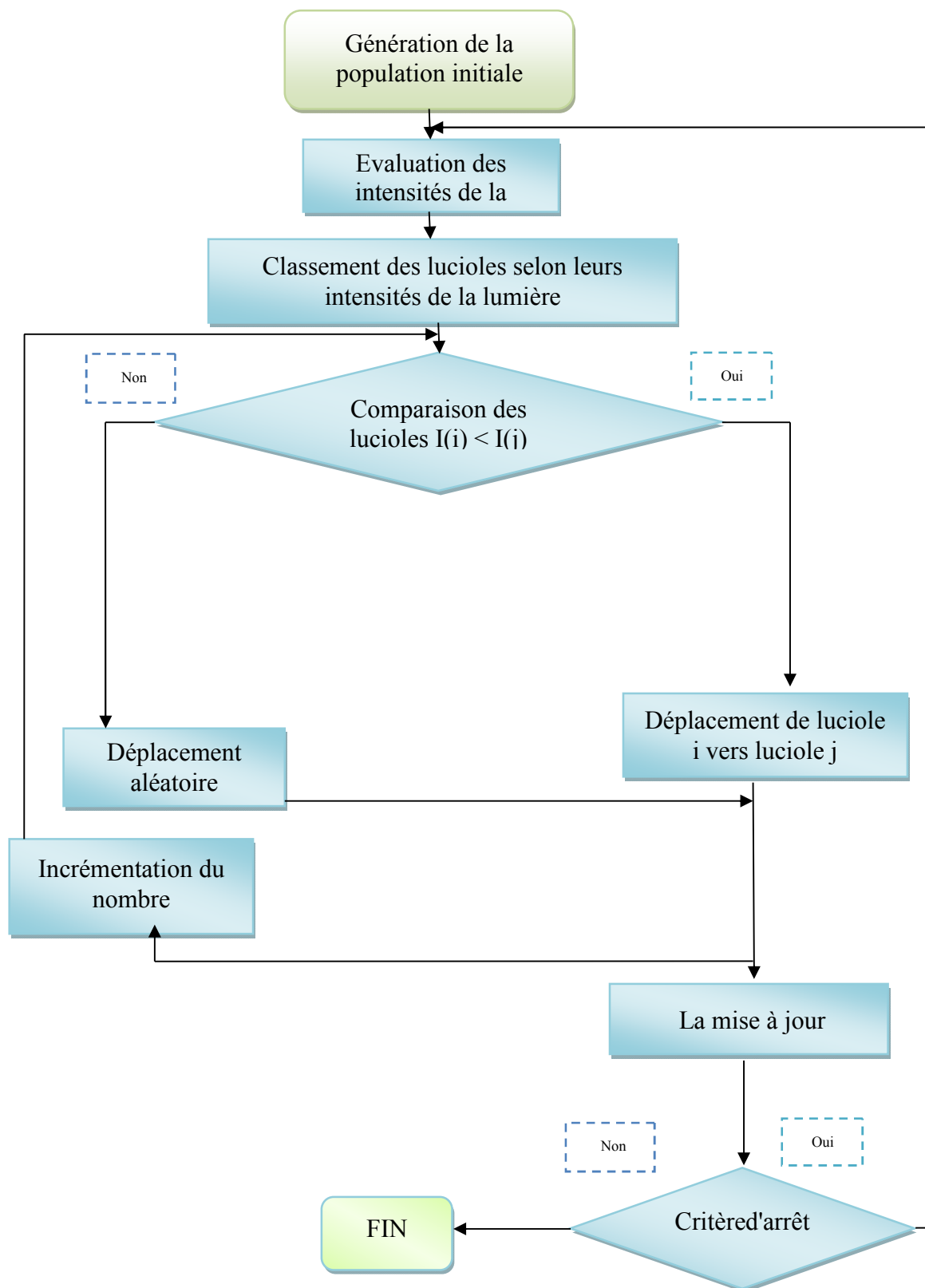




Figure III.2:l'organigramme d'algorithme de luciole

### III.2.3.Description de l'organigramme

#### III.2.3.1. Génération de la population initiale

Dans cette étape, l'algorithme des lucioles "Firefly algorithm" génère une population initiale qui représente un ensemble de solutions possibles.

Cette population initiale est généralement générée aléatoirement, le choix de la population initiale est très important car il peut rendre plus ou moins rapide la convergence vers l'optimum global.

Dans le cas où l'on ne connaît rien sur le problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

En contrepartie, le choix d'une population trop élevée peut augmenter considérablement le temps de calcul, et si la taille de la population est trop petite, il y aura une convergence prématurée car l'algorithme n'a pas un grand échantillon de l'espace de recherche .

#### III.2.3.2. Fonction d'évaluation

La fonction d'évaluation (fitness), attribut a chaque luciole une valeur numérique qui représente un coût de performance, elle est utilisée pour coder la luminosité des Fireflies. Grace à cette fonction l'algorithme converge vers l'optimum.

L'efficacité de l'algorithme en termes de pertinence de la solution et le temps de calcul dépend principalement de la fonction objective, pour cela elle doit définir les fonctions objectives de façon plus fidèle que possible .

Il existe deux types de fonction d'évaluation, soit mono critère ou multicritère :

- **Une fonction d'évaluation mono critère**

Signifie que la fonction dépend d'une seule et même fonction objectif. La résolution de la fonction d'adaptation (fitness), dans ce cas est simple et ne pose généralement aucun problème.

- **Une fonction d'évaluation multicritère**

Généralement, les problèmes d'optimisation doivent souvent satisfaire des objectifs multiples. Une méthode classique consiste à définir des fonctions objectifs élémentaires dont certains sont concurrents, traduisant chaque objectif à atteindre, et de les fusionner au sein d'une seule fonction.

### III.2.2.3. Classement

Le classement s'effectue par rapport à la fonction objective, donc dans notre algorithme le classement se fait selon l'intensité de la lumière de chaque luciole. Généralement le classement sert à déterminer le meilleur individu ou le mauvais individu, si la fonction objective cherche à maximiser les critères, le classement par ordre croissant et donc le meilleur est le maximum et le mauvais est le minimum, sinon si la fonction objective cherche à minimiser les critères, le classement se fait par ordre décroissant et donc le meilleur est le minimum.

### III.2.2.4. Déplacement et mise à jour

Après avoir comparé la luminosité de chaque luciole avec toutes les autres lucioles, les positions des lucioles sont mises à jour en se basant sur des règles de connaissance de la luciole et de leurs voisins, ces règles sont généralement la position initiale, la distance entre deux lucioles comparés et un mouvement aléatoire.

Après le déplacement, la nouvelle luciole est évaluée, sa position et son intensité de lumière sont mises à jour.

Pendant la boucle de comparaison en deux à deux, la meilleure solution est mise à jour de manière itérative. Le processus de comparaison par pair est répété jusqu'à la satisfaction des critères de résiliation[24].

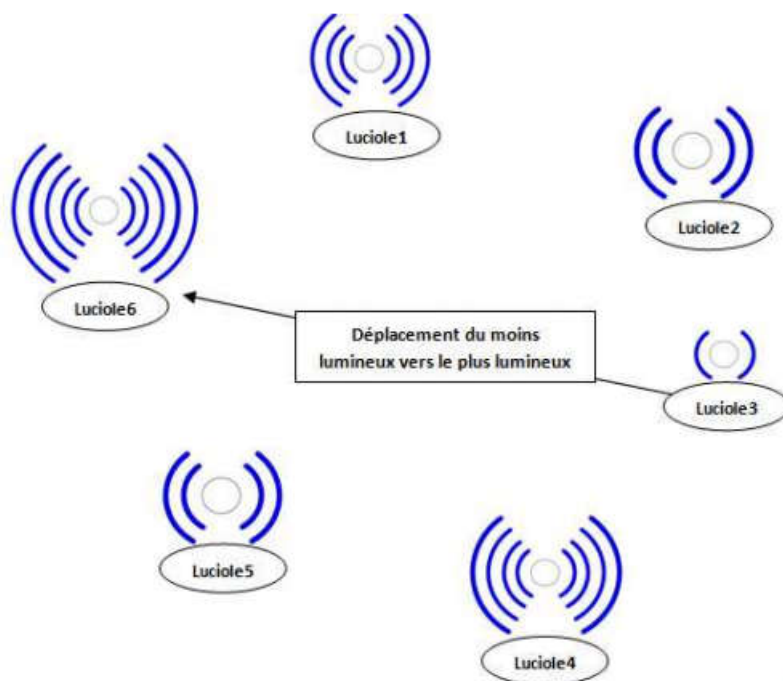


Figure III.3: Déplacement des lucioles

### III.2.2.5. Critère d'arrêt

Les étapes précédentes (déplacement et mise à jour) appliquées d'une manière itérative, cette boucle s'arrête jusqu'à ce que la condition d'arrêt soit satisfaite. Cette condition correspond soit à un nombre maximum de génération fixée au départ, ou quand une solution satisfaisable proche de la solution optimale est atteinte. Aussi on peut arrêter la boucle quand les résultats de l'algorithme sont devenus stable, pour éviter la perte de temps.

L'algorithme de Lucioles se présente comme suite :

#### Algorithme

*Initialisation :*

- 1  $u_i$ :  $i$  Lucioles,  $i \in [1, n]$
- 2  $n$  : le nombre de Lucioles
- 3 *MaxGeneration* : le maximum nombre de génération
- 4  $\gamma$ : coefficient d'absorption
- 5  $r$ : la distance entre deux firefly
- 6  $d$ : la dimension

*Debut*

Définir une fonction objective  $f(x)$ ,  $x = (x_1, \dots, x_d)$

Générer une population de Lucioles  $x_i (i = 1, 2, \dots, n)$

Définir l'intensité de lumière  $I$  à un point  $x_i$  par la fonction objective  $f(x_i)$

Déterminer le coefficient d'absorption  $\gamma$

**Tant que** ( $t < \text{Max Génération}$ )

**Pour**  $i = 1$  jusqu'à  $n$

**Pour**  $j = 1$  jusqu'à  $n$

**Si** ( $I_i < I_j$ )

Déplacer la Lucioles  $i$  vers la Lucioles  $j$

**Fin Si**

Varier l'attraction en fonction de la distance  $r$  via  $\exp[-\gamma r^2]$

Evaluation des nouvelles solutions et mettre à jour l'intensité de lumière

**Fin Pour j**

**Fin Pour i**

Classer les Lucioles et trouver la meilleure solution

**Fin Tant que**

*Fin*

**Figure III.4:**l'algorithme des Lucioles

### III.3. Paramétrages d'algorithmes des Lucioles

La mise en œuvre de l'algorithme Firefly ne pose pas de grandes difficultés par contre le paramétrage de l'algorithme, plutôt, le choix des valeurs est l'étape critique. Ces paramètres dépendent étroitement du type de problème à résoudre, le plus souvent les valeurs de ces paramètres sont réglées en fonction des résultats expérimentaux obtenus.

Dans l'algorithme des Lucioles ou (FA) on prend en compte les paramètres suivants :

- **Nombre de luciole**

Le nombre de lucioles aussi appelé la taille de la population initiale, à une influence directe sur l'algorithme des Lucioles pour cela il est très important de bien choisir ce paramètre pour garantir un meilleur compromis entre la qualité de la solution et la rapidité de l'exécution d'après les différents tests effectués dans , on constate que plus la taille de la population est grande, sa diversité augmente est donc la qualité de solution est meilleure.

Par conséquent si le temps d'exécution de l'algorithme augmente, il affecte l'efficacité de l'algorithme.

Par contre si le nombre de luciole est petit, il y aura alors une probabilité de converger vers un optimum local et donc il est plus efficace d'avoir un nombre important de lucioles pour assurer une diversité et éviter le problème des minima locaux.

- **L'intensité et L'attractivité**

- **Intensité**

Dans l'algorithme des lucioles, il ya deux points importants: la variation de l'intensité de la lumière et la formulation de l'attractivité. Par souci de simplicité, Xin-SheYang suppose que l'attraction d'une luciole est déterminée par sa luminosité qui à son tour est associée à la fonction d'objectif.

Dans la forme la plus simple, l'intensité lumineuse  $I(r)$  varie en fonction de la loi du carré inverse  $I(\mathbf{r}) = I_0 / r^2$  où  $I_0$  est l'intensité à la source, ou dans un milieu donné avec une lumière fixe  $\gamma$  coefficient d'absorption, l'intensité lumineuse varie avec la distance, qui est  $I(\mathbf{r}) = I_0 e^{-\gamma r}$ ,

Afin d'éviter la singularité à  $r = 0$  dans l'expression  $I(\mathbf{r}) = I_0 / r^2$ , l'effet combiné de la fois le loi carrée inverse et l'absorption peuvent être estimés à l'aide de la forme suivante :

$$I(\mathbf{r}) = I_0 e^{-\gamma r^2} \quad (\text{III .1})$$

Parfois, on peut avoir besoin d'une fonction qui diminue de façon monotone à un rythme plus lent. Dans ce cas, nous pouvons utiliser l'approximation suivante :

$$I(r) = \frac{I_0}{1+\gamma r^2}$$

- **Attractivité**

L'attractivité varie en fonction de la distance  $r_{i,j}$  entre les lucioles  $i$  et  $j$ . En outre, l'intensité lumineuse diminue en s'éloignant de sa source, et la lumière est également absorbée par l'entourage, de sorte de permettre à l'attractivité de varier avec le degré d'absorption.

Sachant que l'attraction d'une luciole est proportionnelle à l'intensité des lucioles adjacentes, La formule de cette attractivité  $\beta$  d'une luciole peut être définie comme :

$$\beta(r) = \beta_0 e^{-\gamma r^2} \quad (\text{III.2})$$

Où  $\beta_0$  est l'attraction à  $r = 0$ . Comme il est souvent plus rapide pour calculer  $\frac{1}{1+r^2}$  d'une fonction exponentielle, le cas échéant, peut avantageusement être

remplacé par  $\beta = \frac{\beta_0}{1+\gamma r^2}$ , équation (2) définit une caractéristique de distance  $r = \frac{1}{\sqrt{\gamma}}$ , sur laquelle l'attractivité change sensiblement de  $\beta_0$  à  $\beta_0 e^{-1}$ . Dans la mise en œuvre, la forme réelle de la fonction de l'attractivité  $\beta(r)$  peut être des fonctions décroissantes de façon monotone comme la généralisation suivante : [25]

$$\beta(r) = \beta_0 e^{-\gamma r^m} \quad m \geq 1 \quad (\text{III.3})$$

- **Coefficient d'absorption**

Le coefficient d'absorption  $\gamma$  contrôle la variation de l'attractivité en fonction de la distance entre deux lucioles communiquées. Il est dans l'intervalle  $[0, \infty]$ .

$\gamma = 0$  correspond à aucun changement, pas de variation ou attractivité constante,

$\gamma = \infty$ , correspond à une recherche aléatoire complète.

Nous préférons garder la valeur de  $\gamma \in [0,1]$ ,  $\gamma = 1$  entraîne une attractivité proche de zéro qui est encore équivalente à la recherche aléatoire complète.

Ce coefficient d'absorption personnalisé pourrait être basé sur la "longueur caractéristique" de l'espace de recherche optimisé.

- **La distance**

La distance entre n'importe quelles deux lucioles  $i$  et  $j$  aux emplacements  $X_i$  et  $X_j$  respectivement est la distance cartésienne :

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \quad (\text{III.4})$$

Où  $X_i$ , est la  $k^{\text{ème}}$  composante de la  $i^{\text{ème}}$  luciole, et  $d$  est le nombre de dimensions.

Pour

$d = 2$ , on a :

$$r_{ij} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2} \quad (\text{III.5})$$

- **Mouvement**

Le déplacement d'une luciole  $i$  attirée par une luciole  $j$  plus lumineuse (attrayante), est déterminé par:

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha (rand - 1/2) \quad (\text{III.6})$$

Où le premier terme présente la position actuelle d'une luciole, le second terme est utilisé pour prendre en compte l'attractivité d'une luciole à l'intensité lumineuse vue par des lucioles adjacentes, et le troisième terme est utilisé pour le mouvement aléatoire d'une luciole dans le cas où il n'y a pas d'autre plus lumineuse. Le coefficient  $\alpha$  est un paramètre de répartition aléatoire, tandis que  $rand$  est un générateur de nombre aléatoire distribué de façon uniforme dans l'espace  $[0,1]$  [25].

- **Nombre de génération**

La convergence vers la solution optimale globale n'est pas garantie dans tous les cas même si les expériences dénotent la grande performance de la méthode. De ce fait, il est fortement conseillé de doter l'algorithme d'une portée de sortie en définissant un nombre maximum d'itération.

### III.4. Les avantages et les inconvénients d'algorithme de luciole

#### III.4.1 Les avantages de l'Algorithme Firefly

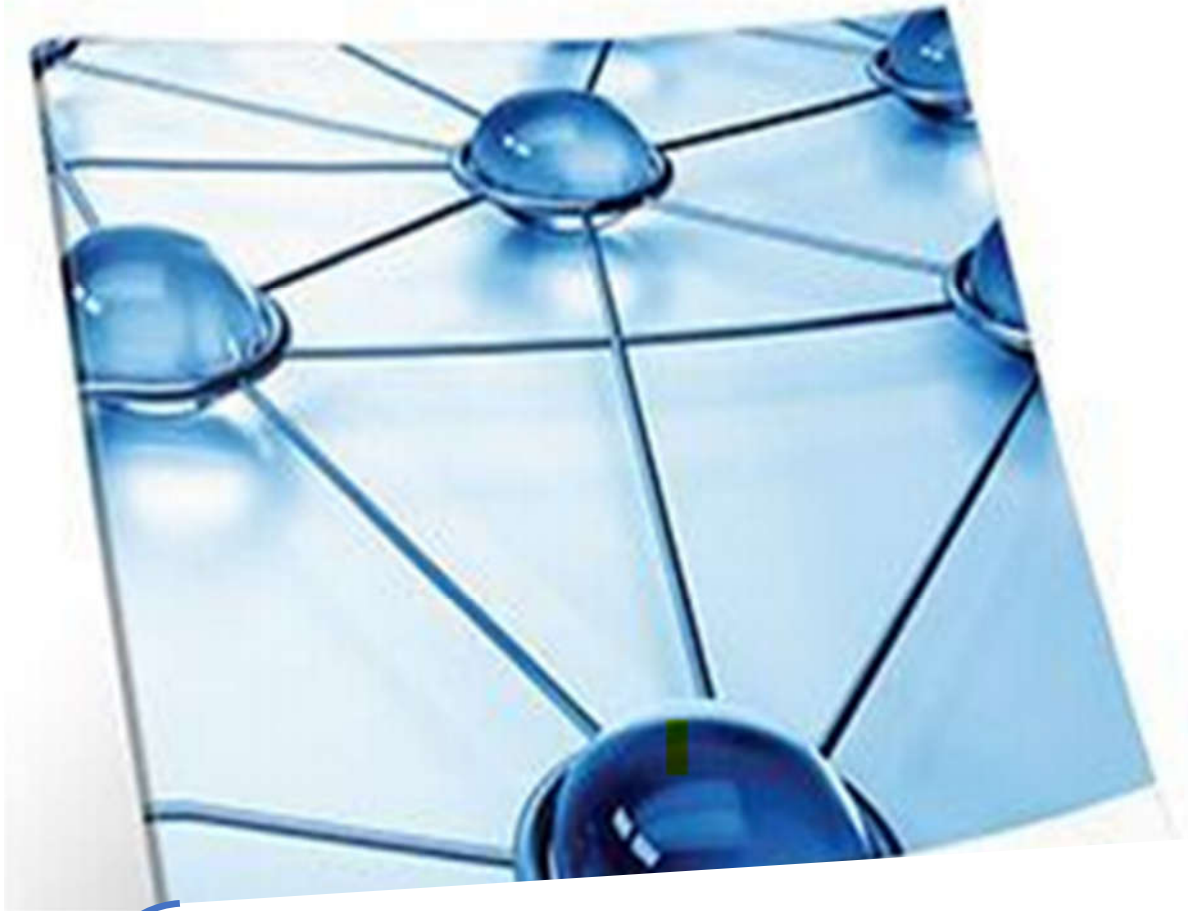
- Algorithme Firefly semble être un phénomène d'outil d'optimisation favorable dû à l'effet de la fonction d'attraction, ce qui est un comportement unique de la luciole.
- Firefly inclut non seulement le processus d'auto-amélioration avec l'espace actuel, mais il comprend également l'amélioration au sein de son propre espace à partir des étapes précédentes.

### III.4.1 Les inconvénients du Algorithme Firefly

- Algorithme Firefly a quelques inconvénients tels que se faire piéger en plusieurs optima locaux.
- Firefly algorithme effectue parfois la recherche locale ainsi et parfois est incapable de débarrasser complètement d'eux.
- Paramètres de l'algorithme Firefly sont fixés et ils ne changent pas avec le temps.
- En outre Firefly ne pas mémoriser ou de se rappeler toute l'histoire de meilleure situation, et ils peuvent finir par manquer leurs situations..

### III.5.Conclusion

L'algorithme des lucioles "Firefly Algorithm" permet de fournir rapidement des solutions qui sont proche de la solution optimale grâce aux mouvements des lucioles en fonction d'attractivité et d'intensité qui est défini par la fonction objectif et qui est le centre de tous les calculs. Les emplacements changent a chaque itération et donc le changement des solutions aussi afin de déterminer le meilleur compromis entre la qualité de la solution et la rapidité de l'algorithme.



## **Chapitre IV: Implémentation et discussion**



## IV.1 Introduction

Nous avons présenté dans les chapitres précédents l'une des contraintes majeures dans la conception des RCSF, il s'agit de l'énergie. Le déploiement des RCSFs dans des zones non surveillées et dont l'intervention humaine est minimale ou quasiment impossible, exige aux capteurs un effort important pour survivre, et les ressources des nœuds de capteurs sont assez limitées.

Dans ce dernier chapitre nous avons implémenté et évalué le nouveau protocole **AFABCMH** qui est basé sur l'adaptation de l'algorithme des lucioles (firefly algorithm), pour bien minimiser la distance totale de transmission dans le réseau à l'aide de clustering ce qui réduirait aussi le nombre de transmissions par la technique multi saut (multi hop), et avec l'agrégation des données dans le RCSF par conséquent moins d'énergie consommée.

Alors pour étudier la performance de notre protocole de routage et son efficacité dans la consommation d'énergie et la prolongation de la durée de vie du réseau, nous allons comparer notre protocole avec un autre protocole de la littérature.

## IV.2 Description générale de l'algorithme proposé

### IV.2.1. Algorithme AFABCMH (Adaptive Firefly Algorithm Base Clustering Multi-hop)

#### *Algorithme AFABCMH*

##### Début

1. Pour le premier rond exécuter **IR-CH (first Random CH)**
2. Appliquer l'algorithme Firefly adapté pour Choisir les membres de chaque CH par **MEMBRE\_CH**  
 Pour i de 1 à max Rond Faire  
 Pour j de 1 à max itération Faire
3. Changer les CH selon **CHANGER\_CH**
4. Adapter le réseau **UPDATE\_RESEAU**
5. Appliquer l'algorithme Firefly adapté pour Choisir les membres de chaque CH par **MEMBRE\_CH**
6. Calculer la fonction objective
7. Sauvegarder la meilleure itération selon sa fonction objective  
 Fin Pour
8. On applique la méthode **MULTI\_SAUT**  
 Fin Pour

##### Fin

*Figure IV.1 : Algorithme AFABCMH*

### IV.3. Description détaillée de l'algorithme proposé

#### IV .3.1. Formulation des CHs

##### ✓ 1R-CH (first Random CH)

Initialement, les CHs sont choisis aléatoirement par la station de base avec de 5% des nœuds du réseau.

On déroule le premier sous algorithme MEMBRE\_CH pour construire les membres (son groupe) de chaque cluster CH selon l'Energie et la distance a la station de base.

La formule suivante.

$$I(\mathbf{r}) = I_0 e^{-\gamma r} \quad (\text{IV } 1)$$

I : Intensité.

$\gamma$  : Absorption

r : distance entre un nœud et la station de base.

**Algorithme MEMBRE\_CH**

$y$  : Coefficient d'absorption

$I_0$  : L'intensité initial

Batterie: Energie de nœud

Range :Rayon de CH

NbClusters: nombre de CH

MeilleurCH: Meilleur CH de ce nœud

Members: nombre de nœud dans un cluster

Génération de population initiale N:nombre de nœud

Debut :

Pour  $i$  de 1 à N Faire

Max=0

Meilleur CH = 0

Pour  $j$  de 1 à NbClusters Faire

Distance (Noeud( $i$ ) et CH( $j$ ))

Newmax=  $I_0 * \exp(-y * \text{Distance})$ ;

Si Newmax>Max &&Distance <=Range

Max=Newmax;

Mindistance = Distance

MeilleurCH = CH( $j$ )

Fin Si

Fin Pour

Si MeilleurCH !=0

CHdeNoeud( $i$ ) = MeilleurCH;

Distance( $i$ ) = Mindistance

SiMindistance> 0

Members(MeilleurCH) = Members(MeilleurCH) + 1

Fin Si

Sinon

CHdeNoeud( $i$ ) = 0

Fin Si

Fin Pour

Fin

**Figure IV.2** : Algorithme MEMBRES-CH

- ✓ Dans la deuxième rond On exécute MEMBRE\_CH pour choisir pour chaque nœud le meilleure CH .

En suite ,le CH donne son rôle à l'un de ces membres dont l'énergie sur la distance de nœud qui plus grand cette changement se faire par la algorithmne suivante CHANGER\_CH

#### Algorithme CHANGER\_CH

Batterie: Energie de nœud

Numberdevoisin : nombre de voisin de chaque nœud

Indexvoisin: l'indexe de voisin

Members: nombre de noeud entre le clustre

NbClusters: nombre de CH

Debut :

Pour I de 1 à NbClusters Faire

MinCost=0 min=CH(i) f=0

Si Members(CH(i))>0

NumberdevoisinCH(i)

Si Numberdevoisin>0

indexvoisin CH(i)

Pour j de 1 à Numberdevoisin Faire

Si Batterie(indexvoisin(j))>0 &&SensorCluster (indexvoisin(j))=CH(i) &&CH(i)~indexvoisin(j)

MinDistanceBase=Distance (indexvoisin(j) et Station de base)

MinCost1=Batterie (indexvoisin(j))/MinDistanceBase;

Si MinCost1>MinCost

f=1 MinCost=MinCost1 min=indexvoisin(j)

Fin si

Fin Si

Fin Pour

Si f=1 Type(min)=2 Type(CH(i))=1

Fin Si

CH(i)=min

Fin Si

Fin Si

Fin Pour

Fin

**Figure IV. 3** :Algorithme CHANGER\_CH

- ✓ En suite on adapter notre réseau par l'algorithme UPDATE\_RESEAU pour donner le rôle de CH a les noudes isolés (nœud ne partaient pas aucun CH) dans le réseau.

**Algorithme UPDATE\_RESEAU**

N:nombre de *nœud*

Batterie: Energie de *nœud*

Type:Type de *nœud* (CH ou bien *nœud simple*)//si 1 *nœud* ou 2 CH

Numberdevoisin :nombre de voisin de chaque *nœud*

Indexvoisin:l'indexe de voisin de chaque *nœud*

T:variable

**Debut**

Pour i de 1 à N Faire

T=0

Si Batterie(i)>0 && Type(i) !=2

Numberdevoisin (i)

Si Numberdevoisin == 0

Type(i)=2

Sinon

Indexvoisin(i);

    Pour j de à Numberdevoisin Faire

Si Type(indexvoisin(j))==2

T=1

Fin Si

Fin Pour;

Si T==0

    Type(i)=2

Fin si

Fin si

Fin si

Fin Pour

**Fin**

*Figure IV.4* : Algorithme UPDATE\_RESEAU

- ✓ Après la formulation des clusters, le réseau entre dans la phase de l'état d'équilibre, où les nœuds commencent effectivement à transmettre leurs données à la station de base. Cela se produit dans les tours et généralement une phase régulière est accompagnée de plusieurs tours. Après avoir terminé la phase régulière, le réseau entre dans le set-up de nouveau et le processus se répète. Il est à remarquer que la communication au sein du cluster est accompagnée d'une communication TDMA et CH- BS est accompagné par CDMA en intégrant une technique de routage multi-sauts (multi-hops) par Algorithme Multi-saut.

```

Algorithme MULTI-SAUT
Début :
Pour i de NmbreNiveau à 1 Faire
  Pour j de 1 à NbClusters Faire
    CH recevoir le données à partir de membre
    Agrégation les données
    Si i==1
      Le CH envoyer les données vers le station de base
    Sinon
      Envoyer les données vers le niveau haut
  Fin Si
  Fin Pour
Fin Pour
Fin

```

*Figure IV.5* : Algorithme MULTI\_SAUT

### IV 3.2. Fonction objective

Cette fonction utilisé pour détermine pour chaque itération la meilleure configuration dans le réseau.

$$\text{Fonction objective} = \text{Echc} \times q / (f1 + f2) \quad (\text{IV } 2)$$

$$f1 = \sum \forall N_i \in C_{p,k} d(N_i, CH_{p,k}) \quad (\text{IV } 3)$$

$$f2 = \sum \forall N_i \in C_{p,k} d(BS, CH_{p,k}) \tag{IV 4}$$

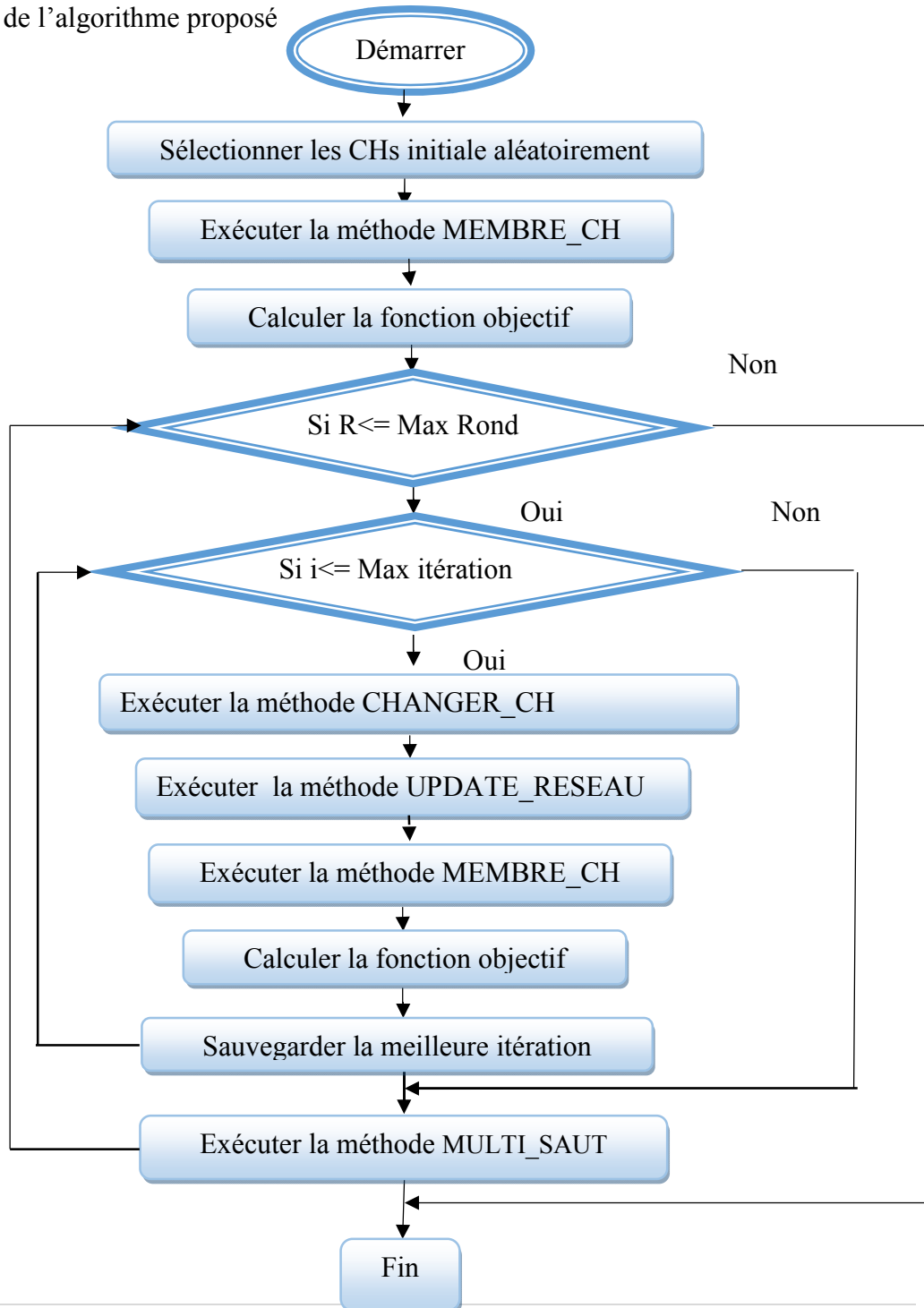
Ehc: est l'énergie de cluster CH candidate.

Q :est un constant.

ou f1: est la somme de distance de nœuds à leurs CH de cluster associés.

f2: est la somme de distance entre station de base et les CH

La figure suivante représente l'organigramme résumant les différentes phases d'exécution de l'algorithme proposé



**Figure IV.6** : l'organigramme de protocole AFABCMH

#### **IV 4. Choix de l'environnement d'implémentation**

Nous avons choisi comme langage d'implémentation pour notre protocole, on a utilisé le MATLAB (MATrixLABoratory), qui est un langage de calcul scientifique de haut niveau et un environnement interactif pour le développement des applications, la visualisation et l'analyse de données, ou encore le calcul numérique.

Nous utilisons MATLAB®13.0 pour résoudre des problèmes de calcul scientifique plus rapidement. Le langage MATLAB met à la disposition du développeur les opérations vectorielles et matricielles, fondamentales pour les problèmes d'ingénierie et scientifiques Il permet un développement et une exécution rapide à l'égard de langage MATLAB nous pouvons programmer et tester des algorithmes plus rapidement qu'avec les langages traditionnels, car il n'est pas nécessaire d'effectuer les tâches de programmation de bas niveau, comme la déclaration des variables, la spécification des types de données et l'allocation de la mémoire.

Les IHM (Interface Homme Machine), sont appelés GUI (Graphique User Interfaces) dans MATLAB. Elles permettent à des objets graphique (boutons, menus, cases à cocher, ...) d'interagir avec un programme informatique.

L'environnement de MATLAB possède 4 fenêtres : a. Au centre l'invite de commande (command window). b. En haut à droite le contenu de l'espace courant de travail (workspace). c. A gauche la liste des fichiers du répertoire courant (currentfolder). d. En bas à droite l'historique des commandes tapées (command history).



## IV .5. Les étapes d'exécution de l'application

### IV 5.1 Lancement d'application

La première interface présentée est l'interface principale de l'application.

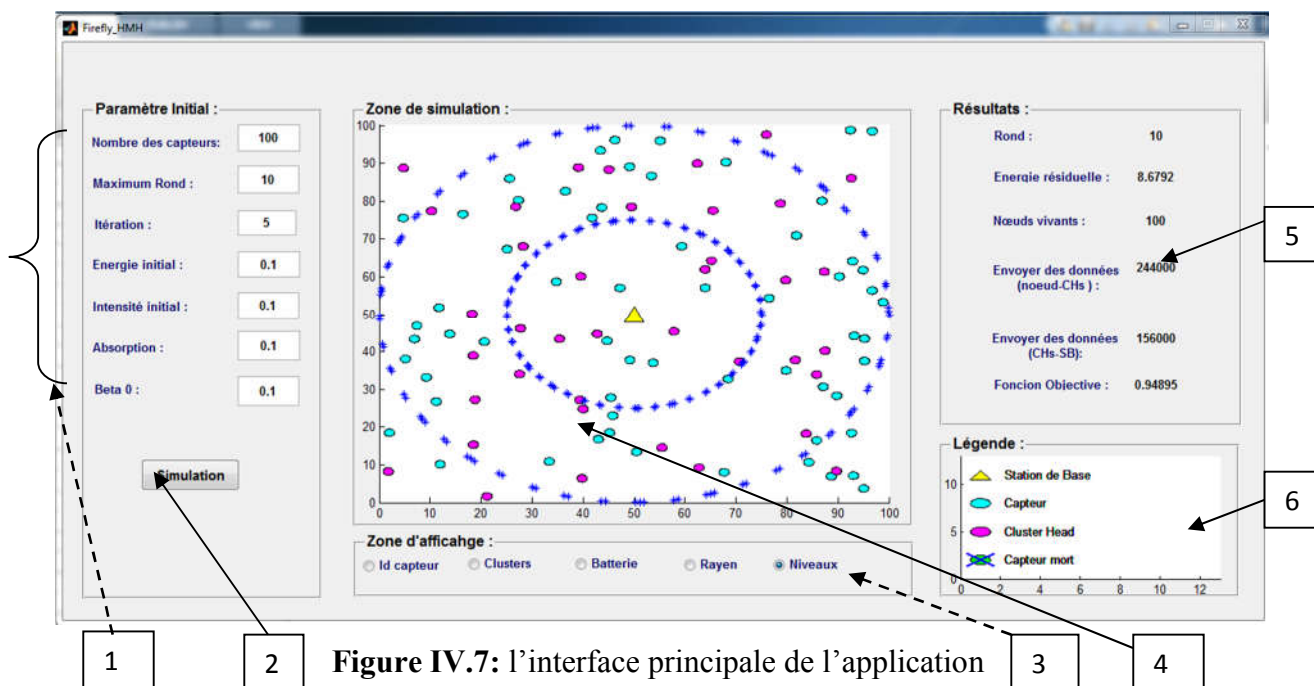


Figure IV.7: l'interface principale de l'application

- 1 : zone des paramètres initial: détermine le paramètre initiale d'algorithme des lucioles;
- 2 : bouton de simulation : la simulation de protocole AFABCMH;
- 3 : zone d'affichage : la fichage des caractéristique des nœuds ;
- 4 : zone de simulation :
- 5 : zone de résultat :le nombre des donnée transmettre dans le réseau et l'énergie résiduelle;
- 6 : légende.

### IV 5.2 Description et paramètres de simulation

Les différents paramètres nécessaires pour la mise en ouvre de notre algorithme proposé, nous avons conçu un simulateur d'un RCSF avec une interface conviviale. Il ya plusieurs paramètres dont nous avons besoin et qui conduisent a bien adapter la configuration le réseau et l'algorithme proposé par rapport au l'objectif de notre travail.

#### IV.5.2.1 Paramètres de simulation

##### ✓ Energie de capture

Un capteur utilise son énergie pour réaliser trois actions principales : l'acquisition, la communication et le traitement des données.

- ✓ **Acquisition** : L'énergie consommée pour effectuer l'acquisition n'est pas très importante. Néanmoins, elle varie en fonction du phénomène et du type de surveillance effectué.
- ✓ **Communication** : Les communications consomment beaucoup plus d'énergie que les autres tâches. Elles couvrent les communications en émission et en réception.

Pour transmettre un message de k bits sur une distance de d mètres, l'émetteur consomme :

$$E_{Tx}(k, d) = E_{Tx}(l) + E_{Tx\_amp}(k, d) \tag{1}$$

$$E_{Tx}(k, d) = \begin{cases} k \cdot E_{elec}(k, d) + k \cdot \epsilon_{friss} \cdot d^2 & \text{si } d < d_{crossover} \\ k \cdot E_{elec}(k, d) + k \cdot \epsilon_{two\_ray\_amp} \cdot d^4 & \text{sinon} \end{cases} \tag{2}$$

Pour recevoir un message de k bits, le récepteur consomme :

$$E_{Rx}(k) = E_{Rx\_elec}(k) = k \cdot E_{elec} \tag{3}$$

Avec :

$E_{elec}$  : Energie de transmission/réception électronique ;

k : Taille d'un message ;

d : Distance entre l'émetteur et le récepteur ;

$E_{TX-amp}$  : Energie d'amplification;

$\epsilon_{amp}$  : Facteur d'amplification;

$d_{crossover}$ : Distance limite pour laquelle les facteurs de transmission changent de valeur.

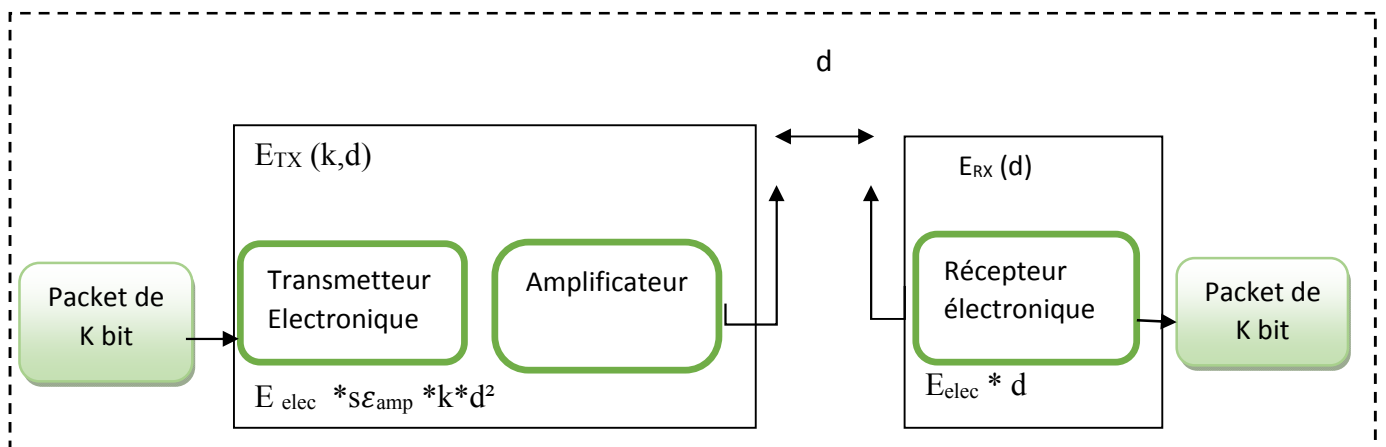


Figure IV.8: Modèle de consommation d'énergie pour la communication

- ✓ **Traitement des données** : L'énergie consommée pour les opérations de calculs est beaucoup plus faible que l'énergie de communication. Un exemple est cité dans [43]. L'énergie nécessaire pour transmettre 1 KB sur une distance de 100m est approximativement équivalente à l'énergie nécessaire pour exécuter 3 millions d'instructions avec une vitesse de 100 millions d'instructions par seconde (MIPS). Ce niveau peut être dépassé en fonction des circuits installés dans les nœuds et des fonctionnalités requises.

Par la suite, elle sera calculée en appliquant la formule suivante :

$$E_{DA} = 5 \text{ nJ/bit/signal}$$

#### IV.5.2.2 Paramètres d'algorithme des lucioles

La population initiale présente l'ensemble des lucioles, la simulation de dans le réseau est réalisé par l'algorithme AFABCMH, alors il faut adapte les paramètre suivant : le nombre du nœud;

- l'intensité de chaque nœud;
- coefficient d'absorption;

#### IV.5.2.3 Paramètre d'affichage

Leur de déroulement de l'algorithme, nous pouvons contrôler le paramètre liés ou l'affichage de réseau comme le rayon de connectivité, le numéro de capture, niveau du cluster et le rapporte d'énergie de capture dans chaque rond et le cluster former. On à aussi afficher le résultat de la simulation concernant le nombre d'itération, la valeur de fonction objective dans chaque rond.

### IV.6. Simulation et résultat

Dans cette section, nous allons présenter les résultats de simulation que nous avons obtenus. Ces résultats valides les performances de notre modèle de simulation que nous avons conçu, et pour cela, ces résultats seront comparés avec celles des autres travaux de la littérature LEACH, et AFABC

La simulation de notre algorithme constitue la plus importante étape de notre travail puisque on peut prouver les améliorations effectuées en termes d'économie d'énergie et de prolongement de la durée de vie global du réseau en analysant les résultats fournis.

Dans cette simulation, et pour comparer les performances de notre algorithme, en termes de l'économie d'énergie et la durée de vie du réseau par rapport aux deux autres algorithmes LEACH et AFABC Tout au long de la simulation et après chaque itération, nous avons mesuré l'énergie résiduelle de réseau, afin de calculer le nombre total

des nœuds morts, nos paramètres de simulation sont les mêmes que dans et nous assumons que tous les nœuds ont une position fixe durant toute la période de simulation et la station de base est positionnée au centre de la zone de captage

<b>Zone de couverture</b>	<b>100*100 m<sup>2</sup></b>
<b>Nombre de nœuds</b>	100
<b>L'énergie initiale des nœuds</b>	0.1 J
<b>Langur de paquet</b>	4000
<b>Eelec</b>	50 nJ/bit
<b>Efs</b>	10 pJ/bit/m <sup>2</sup>
<b>Emp</b>	0.0013 pJ/bit/m <sup>2</sup>
<b>Eda</b>	5pJ/bit/sig
<b>Type de distribution</b>	aléatoire

*Tableau IV.1:* Les paramètres de simulation

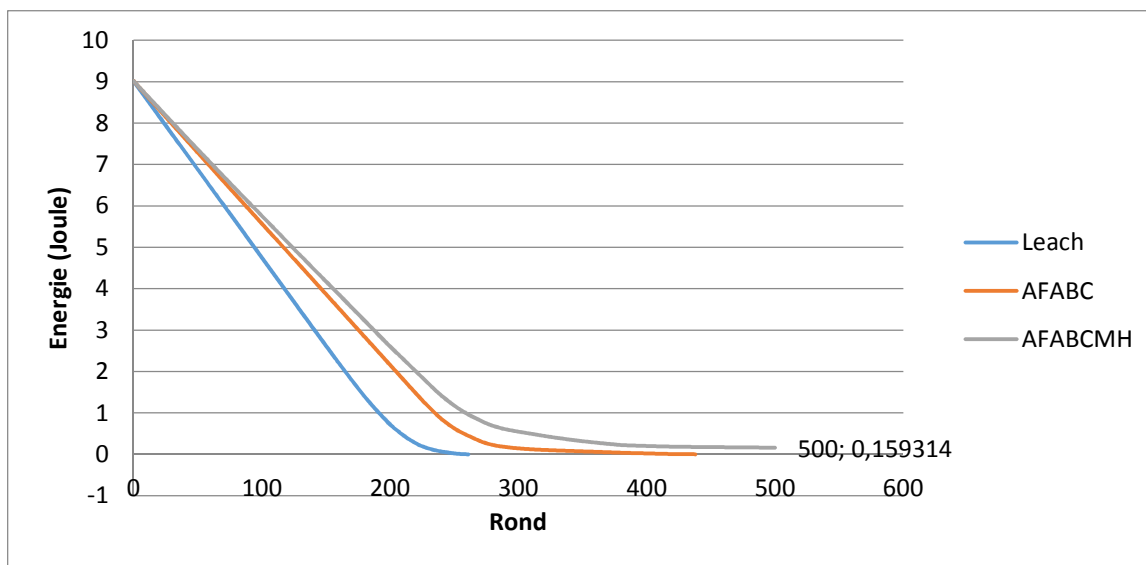
<b>paramètre de l'algorithme des lucioles</b>	<b>Valeur proposé</b>
Intensité initiale des lucioles $I_0$	<b>0.1</b>
Coefficient d'absorption	<b>0.1</b>
$\beta$	<b>0.1</b>

*Tableau IV.2:* Les paramètres de l'algorithme des lucioles

Le tableau IV.3, montre un résultat de notre algorithme en terme d'énergie consommé selon le protocole AFABC, LEACH et le protocole **AFABCMH** par rapport à chaque rond.

<b>Rond</b>	<b>Leach</b>	<b>AFABC</b>	<b>AFABCMH</b>
<b>1</b>	9	9	9
<b>100</b>	4.75	5.57	5.75
<b>200</b>	0.71	2.15	2.60
<b>300</b>	0	0.14	0.74
<b>400</b>	0	0.02	0.20
<b>500</b>	0	0	0.15

*Le tableau IV.3:* la consommation d'énergie après 500 ronds.



**Figure IV.9:** la consommation d'énergie après 500 ronds.

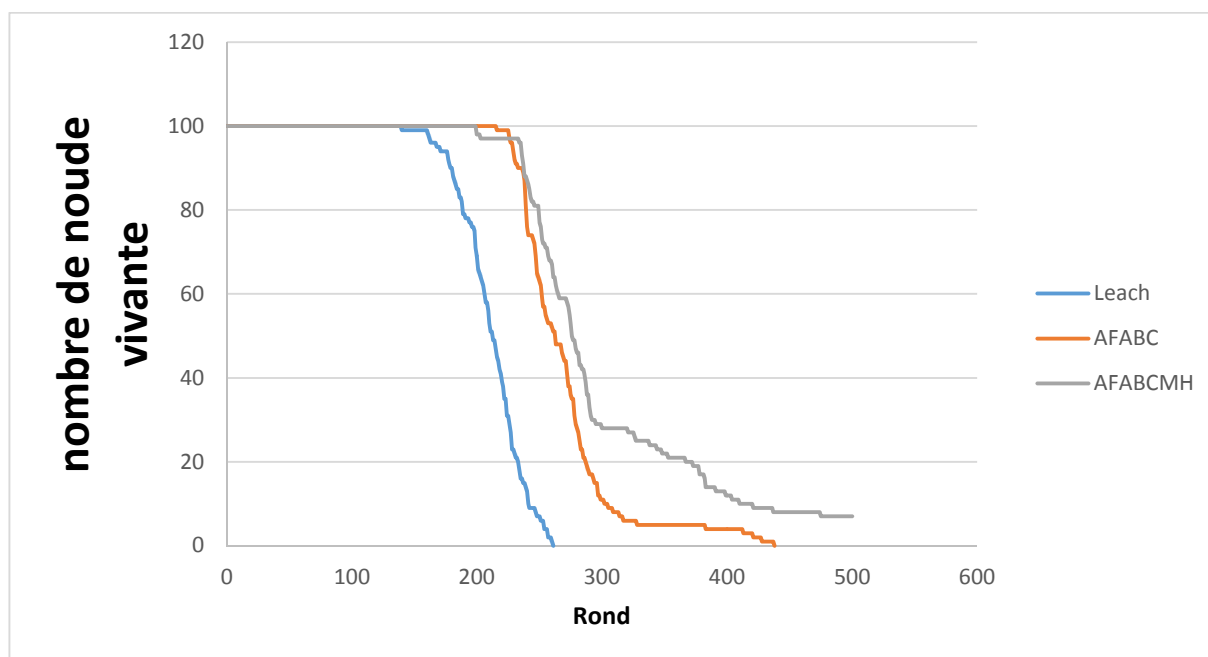
On remarque que pour les trois protocoles ( LEACH, AFABC et AFABCMH) la valeur d'énergie consommée est diminuée avec l'augmentation des nombre des ronds du réseau.

Après 600 ronds le protocole LEACH et AFABC, L'énergie est consommée entièrement, par contre dans le protocole AFABCMH le réseau elle reste vivant.

La Figure IV.4, montre un résultat de notre algorithme en termes de nombre du nœud mort :

Rond	Leach	AFABC	AFABCMH
1	100	100	100
100	100	100	100
200	69	100	100
300	0	11	28
400	0	4	12
500	0	0	7

**Le tableau IV.4:** nombre de nœuds vivants après 500 ronds



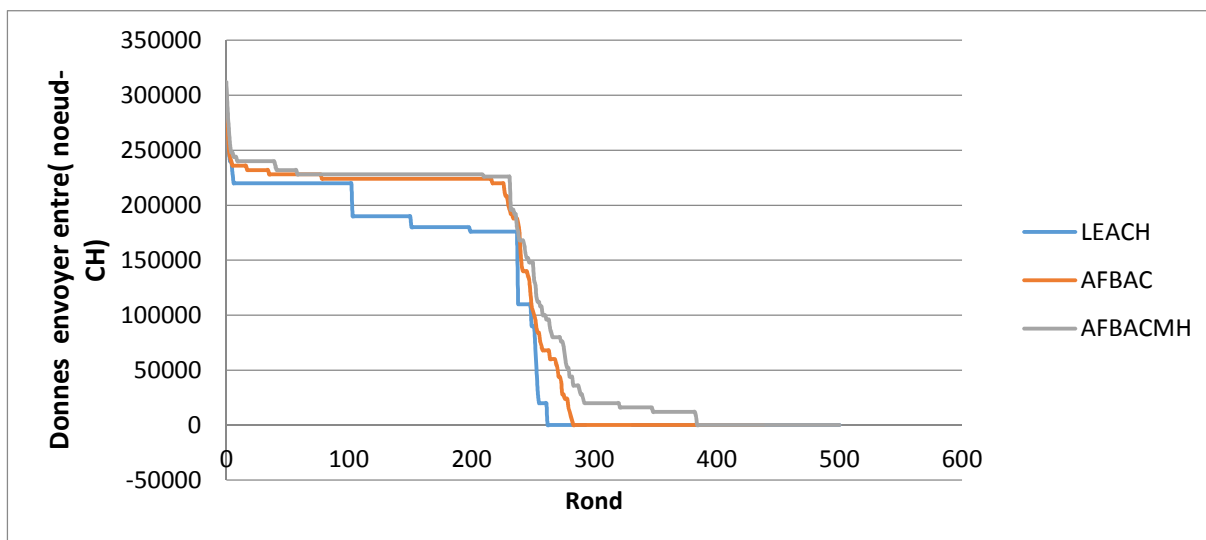
**Figure IV.10:** Durée de vie de réseau (Leach, AFABC, AFABCMH).

On remarque que pour les trois protocoles ( LEACH, AFABC et AFABCMH) le nombre des nœuds vivant diminue avec l'augmentation des nombre des rounds du réseau mais Le LEACH après le 140 rounds, mais AFABC resté à un régime stable jusqu'à la 216 ème rounds , contrairement au AFABCMH qui resté toujours stable jusqu'à la 240 ème round. Alors la durée de vie notre protocole AFABCMH est toujours plus performant par rapport au LEACH.

Le tableau IV.5 représente la transmission des données entre nœuds et CH d'après les protocoles suivante.

Rond	Leach	AFABC	AFABCMH
1	280000	300000	312000
100	220000	224000	228000
200	176000	224000	228000
300	0	0	200000
400	0	0	0
500	0	0	0

Le tableau IV.5 représente la transmission des données entre nœuds et CH



**Figure IV.11:** courbe représenter transmission de (donnes nœuds-CH) après 500 ronds

On remarque que pour les trois protocoles ( LEACH, AFABC et AFABCMH) la transmission du donnes entre (nœuds-CH) diminue avec l'augmentation des nombre des rounds du réseau après 261 rond LEACH est stoppés la transmission mais AFABC on le 300 ème rond par contre notre protocoles **AFABCMH** rester envoyer les donnes jusqu'a 400 rond.

#### IV.7. Conclusion

Les résultats obtenus après différents tests confirment l'efficacité de notre algorithme AFABCMH qui permet de retrouver rapidement des solutions qui sont proche de la solution optimale.

En se basant sur les résultats de la stimulation, nous avons démontré que notre protocole améliore la dissipation d'énergie à l'intérieur des clusters, augmente le gain d'énergie par conséquent, prolonge considérablement la durée de vie du réseau de 10% à 16% comparé au protocole LEACH.



**Conclusion générale**



## Conclusion générale :

Les Réseaux de capteurs sans fil (WSN) sont constitués de nombreux minuscules capteurs déployés à haute densité dans les régions nécessitant une surveillance et un suivi. Ces capteurs peuvent être déployés à un coût beaucoup plus faible que le système traditionnel câblé.

Un capteur typique est constitué d'un ou plusieurs éléments de détection (mouvement, température, pression, etc...), une batterie, récepteurs radio faible puissance, le microprocesseur et une mémoire limitée.

Un aspect important de ces réseaux est que les nœuds sont sans surveillance, ont une énergie limitée et la topologie de réseau est inconnue. Beaucoup de défis de conception qui se posent dans les réseaux de capteurs sont en raison des ressources limitées dont elles disposent et de leur déploiement dans des environnements hostiles.

Dans ce travail, nous avons adapté un algorithme de routage pour l'optimisation de clustering pour un routage efficace en énergie dans les réseaux de capteurs sans fil, il s'agit le protocole **AFABCMH**. Ce protocole est inspiré le comportement des insectes lucioles.

Afin de montrer sa performance en termes de conservation d'énergie, nous avons simulé le fonctionnement de protocole **AFABCMH** avec le simulateur MATLAB 13 et comparé avec deux autres protocoles de la littérature à savoir LEACH et AFABC. Les simulations ont montré des bons résultats dans la plupart des cas, une consommation énergétique très réduite, et par conséquent une prolongation de la durée de vie des réseaux.

Comme une perspective, et pour améliorer les résultats obtenus, nous envisageons, par la suite, les adaptations suivantes:

- ✓ Une hybridation entre notre algorithme avec d'autre méthode de la littérature (ACO, PSO...);
- ✓ Implémenter et tester d'autres techniques intelligentes (Termite Algorithm, Wolf Search Algorithm ...etc.
- ✓ Une technique de mettre en état veille des nœuds redondants, afin de conserver leur énergie et par suite prolonger la durée vie du système.

### Bibliographiques

- [1] I.F. Akyildiz, et al. "Wireless sensor networks : a survey. Computer Networks (Elsevier)", vol.38, no.4, March 2000, pp.393- 422.
- [2] CAYIRCI, E. (2004). "Wireless sensor networks". In : D. Katsaros et al. (éd), Wireless information highways (pp. 273-301). Hershey : Idea group inc.
- [3] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. « Wireless sensor networks >> a survey Computer Networks, Vol. 38(200), pp.393-422.
- [4] Yacine CHALLAL « réseau de capteur sans fil », support de cours, 17/11/2008.
- [5] A. MAKHOUL, Réseaux de capteurs : localisation, couverture et fusion de données, Thèse de Doctorat, Université de Franche-Comté, novembre 2008.
- [6] AKYILDIZ (I. F.), SU (W.), SANKARASUBRAMANIAM (Y.) et CAYIRCI (E.), « Wireless sensor networks : a survey. », IEEE Communications Magazine, vol. 40, n°8, August 2002, p. 102–114.
- [7] D.E. BOUBICHE, Protocole de routage pour les réseaux de capteurs sans fil, Pour l'obtention du diplôme de Magister en Informatique, Université de l'Hadj Lakhdar-Batna, 2008.
- [8] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks. IEEE Communications Magazine, vol 40, pp. 102-114, August, 2002.
- [9]. A. ADOLF, « Conception d'un protocole de routage réactif sécurisé à l'aide de processeurs sécurisé embarqués pour les réseaux ad hoc », master recherche, université de Limoges, 2007
- [10]. Charles E. Perkins, Pravin Bhagwat. « Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers ». ACM SIGCOMM Computer Communication Review. October 1994, Vol. 24, 4.
- [11] Anis Laouiti, Cédric Adjih. Mesures de performances du protocole OLSR. Projet Hipercom. 2003. Rapport technique.
- [12] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, et D.E. Culler, "SPINS: Security Protocols for Sensor Networks," *Wireless Network*, Vol. 8, pp. 521–34, 2002.

## Références Bibliographiques

---

- [13] Lindsey, S. Raghavendra, C.S. PEGASIS: « Power-efficient gathering in sensor information systems. IEEE Aerospace Conference Proceedings. 2002, Vol. 3, pp.31-130.
- [14] HEINZELMAN, W., CHANDRAKASAN, A. et BALAKRISHNAN, H, "Energy-efficient communication protocol for wireless micro sensor networks". Proc. 33rd Annual Hawaii International Conference on System Sciences. vol. 2, pp.1-10, 2000.
- [15] A. Manjeshwar, D.P. Agrawal. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. Proceedings 15th International Parallel and Distributed Processing Symposium. 2001,
- [16] Manjeshwar, Arati and Agrawal, Dharma P. APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. IPDPS '02, 16th International Parallel and Distributed Processing Symposium. 2003, p. 48.
- [17] O. Younis, S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing 03 (4). 2004, pp. 366–379.
- [18] O. Yu, D. Estrin et R. Govindan, « Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Network » rapport technique du département d'informatique de l'université UCLA, May 2001.
- [19]. W.R. Heinzelman, A. Chandrakasan, and H Balakrishnan. "energy-efficient communication protocol for wireless microsensor networks". In IEEE Proc, pages 1–10, Janvier 2002.
- [20] H.K.F. Stanger, J.E. Lloyd., D.M. Hillis, "Phylogeny of North American fireflies (Coleoptera: Lampyridae)", Vol. 45, Issue 1, 2007, pp. 33–49.
- [21] Yang X.S, "Nature-Inspired Metaheuristic Algorithms". Luniver Press, UK. 2008.
- [22] Romana CAPOR-HROSIK, Adis ALIHODZIC, Milan TUBA, Mirjana VUKOVIC, Milenko PIKULA, "Firefly Algorithm for Constrained Optimization Problems". ISBN: 978-960-474-330-8.
- [23] Ali Saoucha, N, K. Ghanem, and B. Benmammar. "On applying firefly algorithm for cognitive radio networks." Communications and Vehicular Technology in the Benelux (SCVT), 2014 IEEE 21st Symposium on. IEEE, 2014.
- [24] Romana CAPOR-HROSIK, Adis ALIHODZIC, Milan TUBA, Mirjana VUKOVIC,

## Références Bibliographiques

---

Milenko PIKULA, "Firefly Algorithm for Constrained Optimization Problems". ISBN: 978-960-474-330-8.

[25] Xin-She Yang, "Nature-Inspired Metaheuristic Algorithms", 2nd Edition Copyright © 2010 Luniver Press.