

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

Proposition d'un algorithme de routage efficace en énergie dans les réseaux de capteur sans fil basé sur l'intelligence en essaim inspiré de la nature

Préparés par

Fatouma BENBA et Zeyneb BENBA

Soutenu devant le Jury :

Encadreur : Mr. Mohammed DEMRI	Maître assistant classe - B- Université d'Adrar
Président: Mr. Abdelouahab OUAHAB	Maître assistant classe - B- Université d'Adrar
Examineur: Mr. Mohammed KADDI	Maître assistant classe - B - Université d'Adrar
Examineur: Mr. RABHI Saddik	Maître assistant classe - B - Université d'Adrar

Année Universitaire 2015/2016

Résumé

L'objectif de ce travail est d'adapter et implémenter un protocole de routage hiérarchique bio-inspiré basé sur le principe de comportement des lucioles (Algorithme des lucioles) ou (Firefly algorithm). Dans le protocole AFABC (Adaptive Firefly Algorithm Based Clustering) adapté, La sélection des clusters au premier rond s'effectue à l'aide d'une nouvelle méthode qui s'appelle 1R-CH (1st Round-Clustering Hierarchy) et la formation des clusters est basée sur l'attraction entre les nœuds selon leur intensité. En effet, en termes d'économie d'énergie et la prolongation de la durée de vie d'un RCSF, cette nouvelle approche apporte des améliorations remarquables en le comparant avec deux protocoles de la littérature (LEACH et Fire-LEACH).

Mots-clés: Protocole, Clustering, RCSFs, Firefly, Lucioles, Energie, AFABC, 1R-CH.

Abstract

The objective of this work is to adapt and implement a bio-inspired hierarchical routing protocol based on the behavior of fireflies (Firefly algorithm). In AFABC protocol (Adaptive Firefly Algorithm Based Clustering) adapted, the selection of clusters in the first round is accomplished using a new method called 1R-CH (1st Round-Clustering Hierarchy) and the formation of clusters is based on the attraction between nodes according to their intensity. Indeed, in terms of energy saving and prolonging the life of a WSN, this new approach brings remarkable improvements by comparing it to two protocols in the literature (LEACH and LEACH-Fire).

Keywords: Protocol, Clustering, WSN, Firefly, Fireflies, Energy, AFABC, 1R-CH.

Dédicace

Je dédie ce modeste travail à :

A les lumières de mes jours

les sources de mes efforts

Les flammes de Mon cœur

ma mère et ma grand-mère

que j'adore .

A l'homme de ma vie, celui qui s'est

toujours sacrifié pour me voir réussir

mon père.

À Tous mes frères et mes sœurs surtout

mes petite sœurs ALIYA et BOUCHRA

A mes aimables amis, collègues d'étude

A tous mes familles.....

Fatouma

Dédicace

*Je remercie Allah le tout puissant qui
m'a donné la force et le savoir pour
réaliser ce travail*

Je dédie ce modeste travail

A mon père et A ma mère ;

A mes frères et mes sœurs ;

A toute ma famille ;

Pour personne m'ayant aidé :

L. Abdeltkader.

*Pour tous les enseignants au cours
de mes études.*

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 loin à la réalisation de notre mémoire.

BENBA Zeyneb

&

BENBA Fatouma

Table de matière

Résumé.....	I
Dédicace	II
Remerciements.....	III
Table de matière.....	IV
Liste des figures	V
Liste des tableaux.....	VI
Liste de l' abréviation.....	VII

Introduction générale

1. Contexte général	1
2. Motivations et problématique.....	1
3. Contributions et structure de la thèse	1

Chapitre I: Généralités sur les réseaux de capteurs sans fil (RCSFs)

I.1 Introduction.....	4
1.2 Les capteurs	4
I.2.1 Définition	4
I.2.2. Architecteur d'un capteur.....	4
I.3. Les réseaux Ad hoc.....	6
I.3.1 Les problèmes de l'ad hoc.....	7
I.4. Les réseaux des capteurs sans fil.....	7
I.4.1. Architecture d'un réseau de capteurs sans fil.....	9
I.4.1.1 Les types d'architecture des RCSF.....	10
I.4.2. Collecter les informations.....	11
I.4.2.1 Collecter les informations à la demande.....	11
I.4.2.2 Collecter les informations à suite à un événement.....	12
I.5. Pile protocolaire.....	12
I.5.1 Rôle des couches.....	13
I.6. Domain d'application	14
I.6.1. Applications militaires.....	15
I.6.2. Applications liées à la sécurité.....	16
I.6.3. Applications environnementales.....	16

I.6.4. Applications commerciales.....	17
I.6.5. Applications médicales.....	17
I.7. Contraintes dans la conception d'un réseau de capteurs.....	18
1.7.1. Contraintes liées à l'application.....	18
1.7.2. Contrainte énergétique.....	18
1.7.3. Contraintes liées aux déterminismes.....	18
1.7.4. Contraintes de passage à l'échelle.....	19
1.7.5. Contraintes liées à la qualité de service.....	19
1.7.6. Contraintes liées à la protection de l'information.....	19
1.6.7. Contraintes liées à l'environnement.....	19
1.7.8. Contraintes de simplicité.....	20
1.8. Conclusion	20

Chapitre II: Les protocoles de routage dans RCSF

II.1. Introduction.....	21
II.2. Le routage dans les RCSFs.....	21
II.2.1. Les type de routage.....	21
2.1.1 Protocole proactif	21
2.1.2. Protocole réactif	21
II.2.2. Les protocoles de routage	22
II.2.2.1 Les protocoles de routage plat.....	22
II.2.2.2. Protocoles de routage hiérarchiques.....	23
II.2.2.3. Les protocoles de routage géographique.....	35
II.3. Conclusion	36

Chapitre III : Algorithme des Lucioles (Firefly)

III.1. Introduction.....	37
III.2 Algorithme des Lucioles (FireFly Algorithm).....	37
2.1. Inspiration.....	37
2.1. Principe de fonctionnement de l'algorithme des Lucioles.....	38
2.3. Description de l'organigramme.....	40
2.3.1. Génération de la population initiale.....	40
2.3.2. Fonction d'évaluation.....	40
2.2.3. Classement.....	41
2.2.4. Déplacement et mise à jour.....	41
2.2.5. Critère d'arrêt.....	41
III.3.1. Paramétrages d'algorithmes des Lucioles.....	43
III.3. Conclusion	45

Chapitre VI: Implémentation et discussion

VI.1 Introduction.....	46
VI .2 description général de l'algorithme proposé.....	46
VI.2.1 Algorithme AFABC (Adaptive Firefly Algorithme base clustering).....	46
VI.2.1.1 Formation des CHs	46
VI .2.1.1.2 Algorithme AFABC.....	48
VI. 2.1.2. Fonction objective.....	48
VI 3. Choix du langage et de l'environnement d'implémentation	51
VI 4 Les étapes d'exécution de l'application.....	51
VI 4 .1 Lancement d'application	51
VI 4 .2 Description et paramètres de simulation.....	52
VI 4 .3 Paramètres de simulation.....	53
VI 4.3.1 Energie de capture	53
VI 4.3.2 Paramètres d'algorithme des lucioles.....	54
VI 4.3.3 Paramètre d'affichage.....	55
VI.5. Simulation et résultat.....	56
VI.7. Conclusion.....	60

Conclusion et perspectives

1. Conclusion	61
2. Perspectives	61

Références bibliographique

Références bibliographique.....	62
---------------------------------	----

Liste des figures

Chapitre I

Figure I.1:	Architecture d'un capteur sans fils	4
Figure I.2	Les échanges dans les modes Infrastructure et Ad-hoc.....	6
Figure I.3:	Exemples des capteurs sans fils	7
Figure I.4:	Le microcontrôleur	8
Figure I.5:	Réseau de capteur sans fil (WSN)	9
Figure I.6:	Les types d'architecture des RCSF	9
Figure I.7:	Architecture plate	10
Figure I.8:	Architecture hiérarchique	11
Figure I.9:	Collecter les informations à la demande	11
Figure I.10:	Collecter les informations Suite à un événement.	12
Figure I.11:	La pile protocolaire des RCSFs.....	13
Figure I.12:	Domaines d'application de RCSFs	14
Figure I.13:	Un service militaire utilisant les RCSF	15
Figure I.14:	Application sur le contrôle de la qualité de l'eau.....	16
Figure I.15:	Le flux d'information d'un patient	17

Chapitre II

Figure II.1:	Les classes de protocoles de routage	22
Figure II.2:	Fonctionnement du protocole SPIN	23
Figure II.3:	Architecture en cluster	24
Figure II.4:	Décomposition du réseau en zones dans ZHLS	25
Figure II.5:	: Le Clustering dans un réseau de capteurs	27
Figure II.6:	Opérations de l'étape d'initialisation	28
Figure II.7:	Répartition du temps et différentes phases pour chaque rond.....	31
Figure II.8:	Formation des chaînes gourmandes dans PEGASIS.....	32

Chapitre III

Figure III.1:	Les Lucioles.....	37
Figure III.2:	l'organigramme d'algorithme de luciole.....	39
Figure III.3:	Déplacement des lucioles	41
Figure III.4:	l'algorithme des Lucioles.....	42

Chapitre VI

Figure IV.1:	formation des CHs avec le protocole 1R-CH.....	47
Figure IV.2 :	Algorithme AFABC.....	48
Figure IV.3 :	l'organigramme de protocole AFABC.....	50
Figure IV.4 :	L'interface initiale d'application	52
Figure IV.5:	L'interface Principale de simulation.....	52
Figure IV.6 :	La simulation.....	54
Figure IV.7 :	Résultats de simulation.....	55
Figure IV.8:	la consommation d'énergie après chaque rond.....	55
Figure IV.9:	nombre des nœuds mort.....	59
Figure IV.10:	valeurs de fonction objective après chaque rond.....	60

Liste des tableaux

Chapitre IV

Tableau IV.1 Les paramètres de simulation.....	56
Tableau IV.2 Les paramètres de l'algorithme des lucioles.....	57
Tableau IV.3 la consommation d'énergie après 1201 chaque rond.....	57
Tableau IV.4 nombre de nœuds morts après chaque rond.....	58
Tableau IV.5 représente les valeurs de fonction objective.....	59

Liste des abréviations

- RCSFs : les Réseau Capture Sans Fils .
- Rc : Rayon de Communication.
- (Rs) : Rayon de Sensation.
- MAN : Mobile Ad hoc NET Works.
- MAC : Medium Access Control .
- DSDV : Destination Sequenced Distance Vector.
- GSR : Global State Routing.
- NEXTi : Next Hop .
- FSR : Fisheye State Routing .
- AODV : Ad-hoc On Demand Distance Vector.
- RREQ : Route Requête message.
- DSR : Dynamic Source Routing .
- OLSR : Optimized Link State Routing.
- MPR : MultiPoint Relaying.
- TC : Topology Control.
- SPIN : Sensor Protocol for Information via Negotiation.
- ZHLS : Zone-based Hierarchical Link State Protocol.
- LEACH : Low-Energy Adaptive Clustering Hierarchy.
- CH : Cluster Head.
- BS : Base Station .
- HEED : Hybrid, Energy-Efficient, Distributed approach.
- FA : Firefly Algorithme .
- 1R-CH : 1 Round-Clustering Hiérarchique .
- AFABC : Adaptive Firefly Algorithme base Clustering .

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 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 des clusters dans les réseaux de capteurs sans fils, en utilisant une nouvelle stratégie pour la sélection des CH au premier rond.

1.3 Contributions et structure du mémoire

Ce mémoire est organisé comme suit;

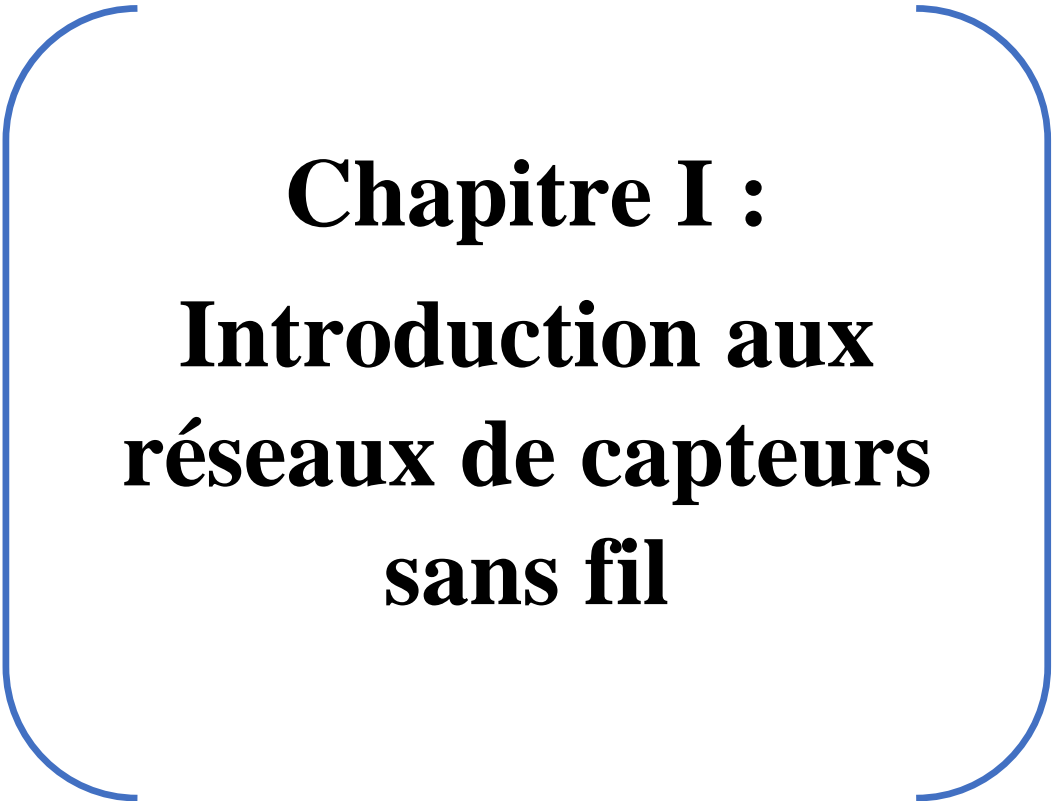
Dans le premier chapitre nous présenterons les réseaux de capteurs sans fil: leurs architectures de communication et leurs applications. Nous discuterons également les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil. Nous terminerons le chapitre par une description de la problématique de la consommation d'énergie dans les 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.

Introduction générale

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

Le dernier chapitre est divisé en deux parties, la première est consacré à la présentation de notre protocole de routage hiérarchique pour les RCSFs qui s'appelle AFABC. Ce dernier est basé sur la stratégie AF1R-CH pour la formulation des CHs selon le facteur énergétique. Dans la deuxième partie, nous détaillerons le simulateur que nous avons conçu pour la validation de notre protocole et nous discuterons les résultats obtenus.



Chapitre I :
Introduction aux
réseaux de capteurs
sans fil

I.1 Introduction

Les progrès réalisés ces dernières années dans les domaines des microsystèmes électromécaniques (MEMS) ainsi que des techniques de communication sans fil ont permis d'apparaître un nouveau type de réseau : le réseau de capteurs sans fil (RCSF). RCSF est composé d'un ensemble d'unités de traitements embarquées, appelées "capteurs", communiquant via des liens sans fil. Le but général d'un RCSF est la collecte d'un ensemble des paramètres de l'environnement entourant les capteurs, telles que la température ou la pression de l'atmosphère, afin de les acheminer vers des points de traitement.

Dans ce chapitre, nous présentons les RCSFs, 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. [2]

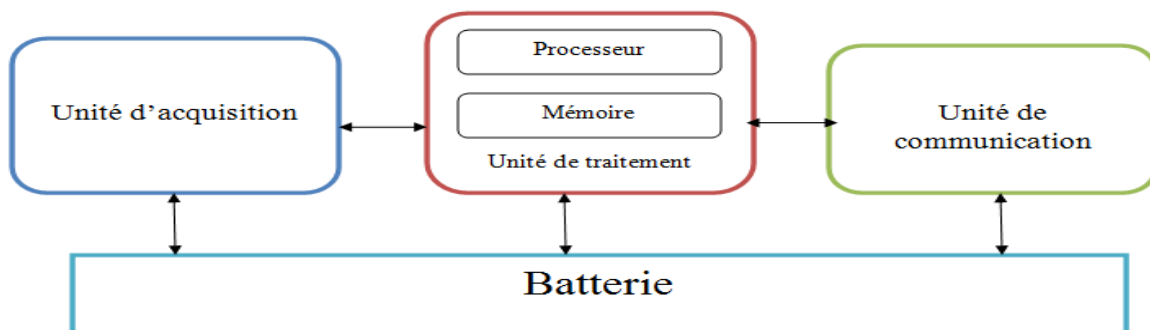


Figure 1.1: Architecture d'un capteur sans fils

✍ 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.

I.3. Les réseaux Ad hoc

Les réseaux Ad hoc (en latin : « **qui va vers ce vers quoi il doit aller** », c'est-à-dire « formé dans un but précis », telle qu'une commission **ad hoc**, formée pour régler un problème particulier sont des réseaux sans fil capables de s'organiser sans infrastructure définie préalablement.

Les réseaux ad hoc, dans leur configuration mobile, sont connus sous le nom de MANet (pour Mobile Ad hoc NET Works). [1]

Chaque entité (nœud) communique directement avec sa voisine. Pour communiquer avec d'autres entités, il lui est nécessaire de faire passer ses données par d'autres qui se chargeront de les acheminer. Pour cela, il est d'abord primordial que les entités se situent les unes par rapport aux autres, et soient capables de construire des routes entre elles : c'est le rôle du protocole de routage.

Ainsi, le fonctionnement d'un réseau ad hoc le différencie notablement d'un réseau comme le réseau GSM, les réseaux Wi-Fi avec des points d'accès : là où une ou plusieurs stations de base sont nécessaires à la plupart des communications entre les différents nœuds du réseau (mode Infrastructure), les réseaux ad hoc s'organisent d'eux-mêmes et chaque entité peut jouer différents rôles. [1]

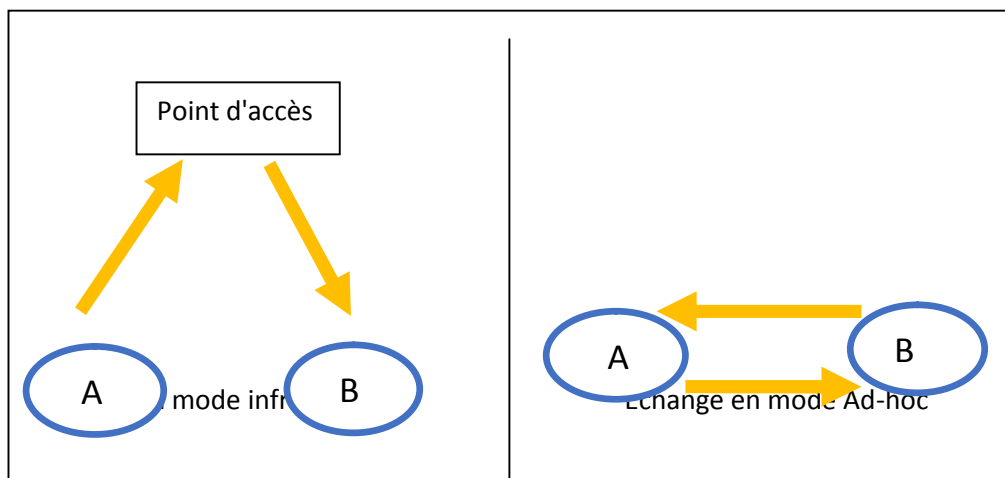


Figure I.2 : Les échanges dans les modes Infrastructure et Ad-hoc [1]

L'utilisation la plus simple et la plus courante des réseaux ad hoc est faite par les réseaux sans fil Wi-Fi en permettant une mise en place rapide d'une connexion réseau entre deux ordinateurs. [1]

I.3.1 Les problèmes de l'ad hoc

Les principaux problèmes des réseaux ad hoc, et les problématiques à gérer sont [1] :

- ✓ Absence d'infrastructure ;

- ✓ Bande passante limitée ;
- ✓ Perte de données ;
- ✓ Perte de routes ;
- ✓ Contraintes de consommation d'énergie ;
- ✓ Sécurité limitée;
- ✓ Erreur de transmission;
- ✓ Interférences ;
- ✓ Nœuds cachés;
- ✓ Auto-configuration et détection d'adresses dupliquées.

I.4. 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.

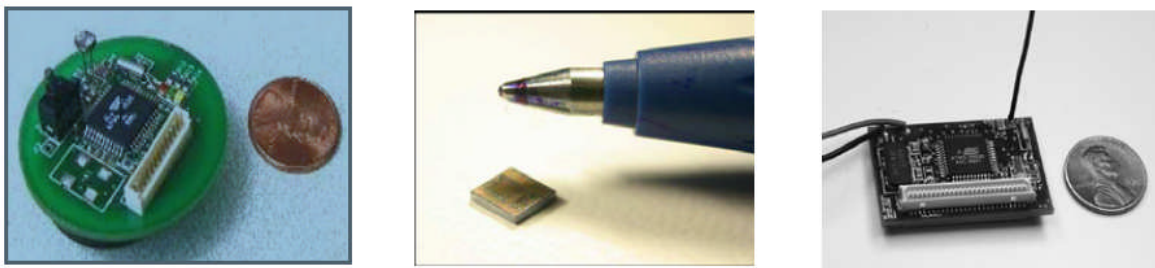


Figure 1.3: Exemples des capteurs sans fils

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. [1]



Figure 1.4 : 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.4.1. Architecture 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 puits (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. [3]

En général, un RCSF est composé de quatre éléments montrés par la figure 1.4 : les nœuds capteurs, une station de base, phénomène à mesurer et l'utilisateur.

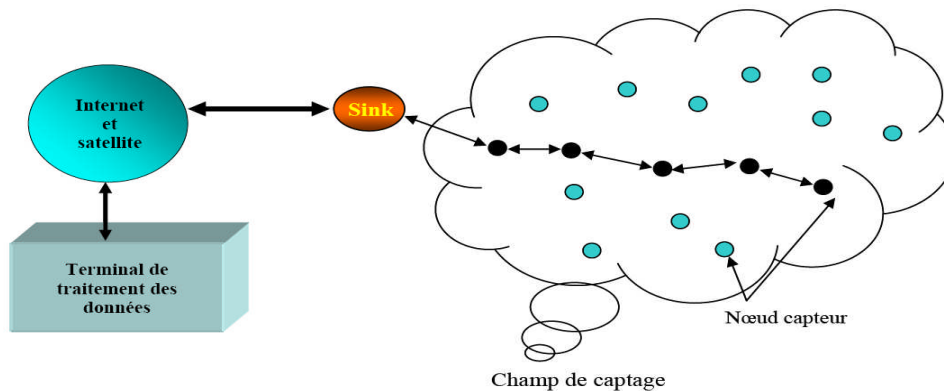


Figure 1.5: Réseau de capteur sans fil (WSN)

I.4.1.1 Les types d'architecture des RCSF

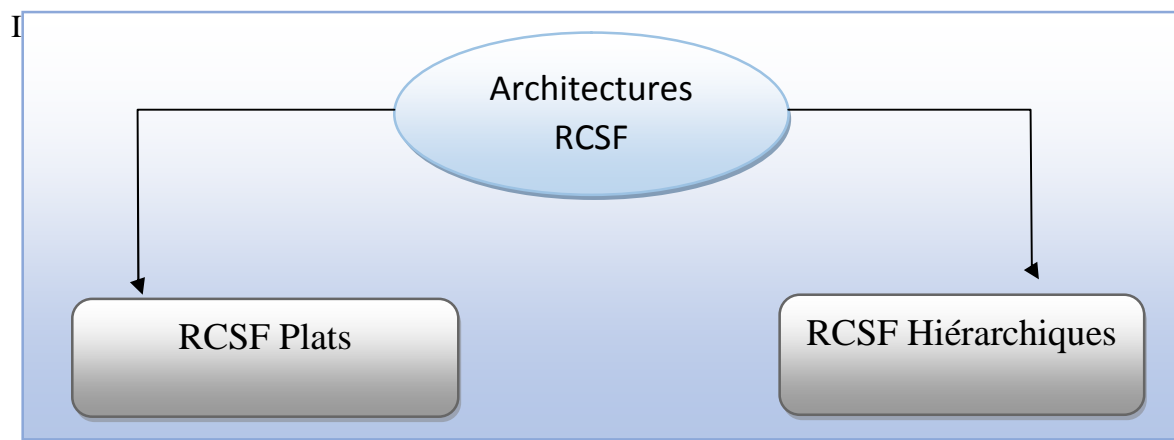


Figure 1.6: Les types d'architecture des RCSF

✍ 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/) 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 (M AC) doivent gérer et organiser les nœuds d'une manière très efficace en termes d'énergie. [4]

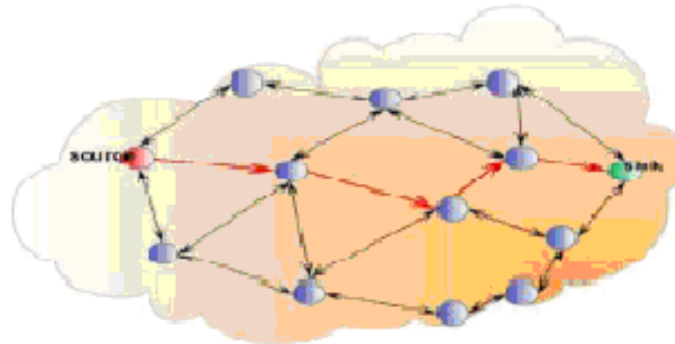


Figure1.7 : Architecture plate

✍ 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.

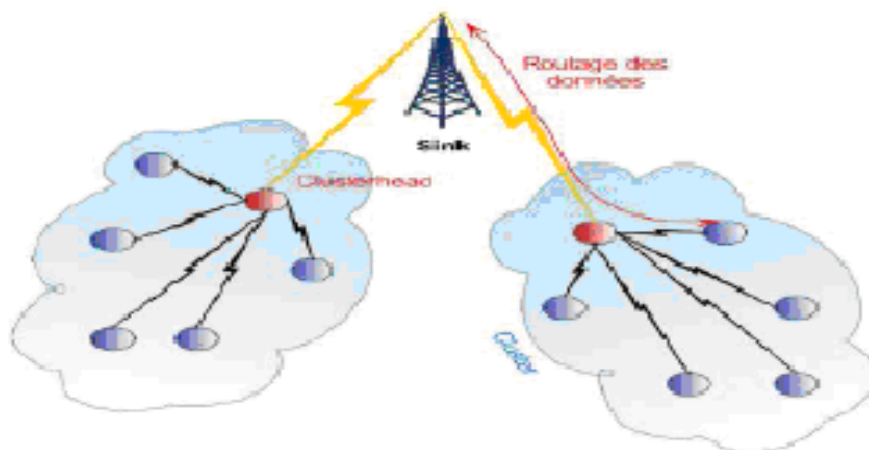


Figure1.8 : Architecture hiérarchique [4]

I.4.2. Collecter les informations

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

I.4.2.1 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.

[4]

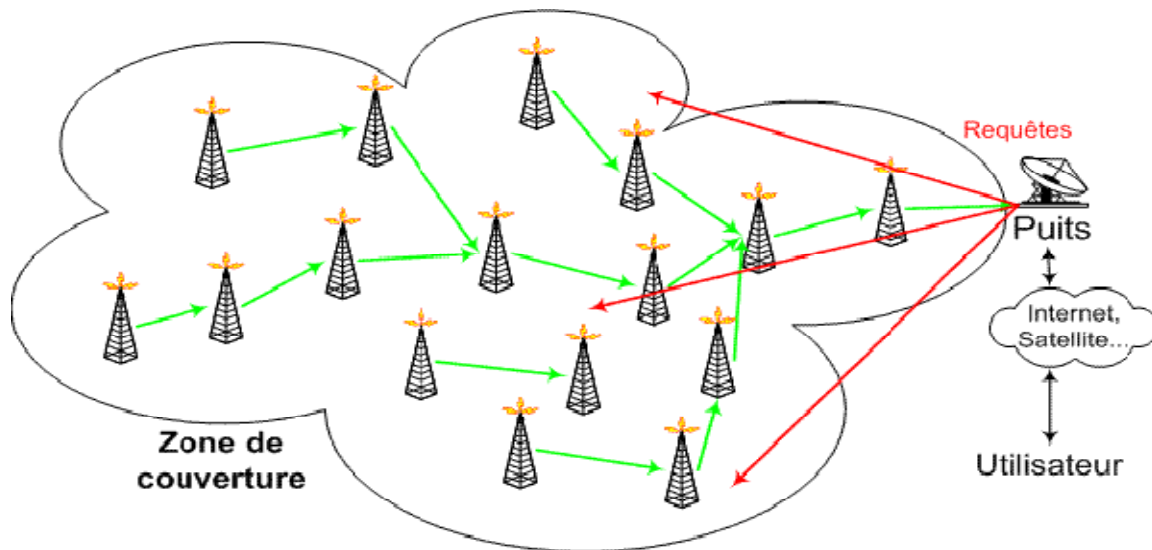


Figure I.9: Collecter les informations à la demande. [4]

I.4.2.2 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.

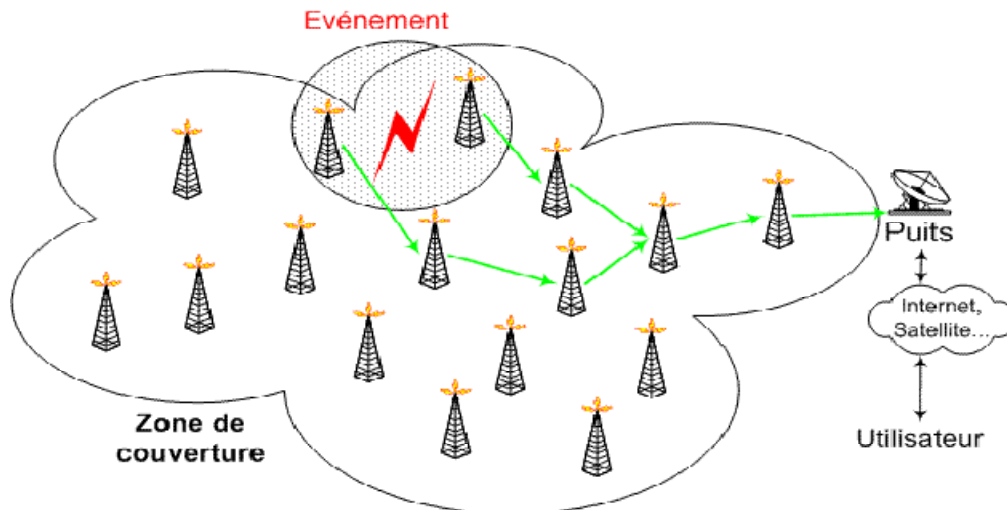


Figure I.10: Collecter les informations Suite à un événement. [4]

I.5. 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 [5].

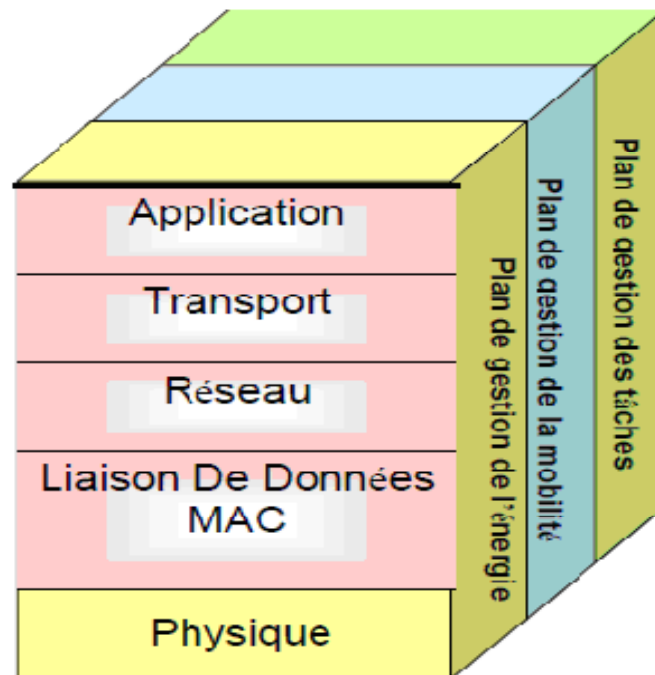


Figure I.11 : La pile protocolaire des RCSF [5]

I.5.1 Rôle des couches [4]

- ✍ **La couche physique** : Spécifications des caractéristiques matérielles, des fréquences porteuses, etc.
- ✍ **La couche liaison** : 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.

[5]

I.6. Domain d'application

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

« 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.12 : Domaines d'application de RCSFs.

I.6.1. 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'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 [6].

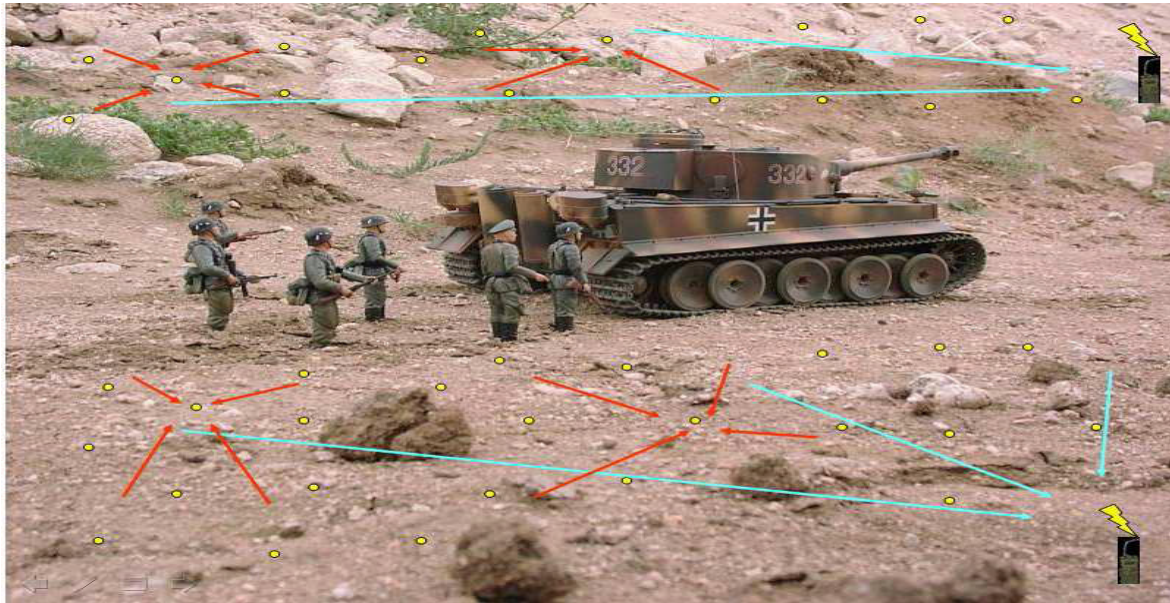


Figure I.13 : Un service militaire utilisant les RCSFs

I.6.2. 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.

I.6.3. 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 mm³ 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.

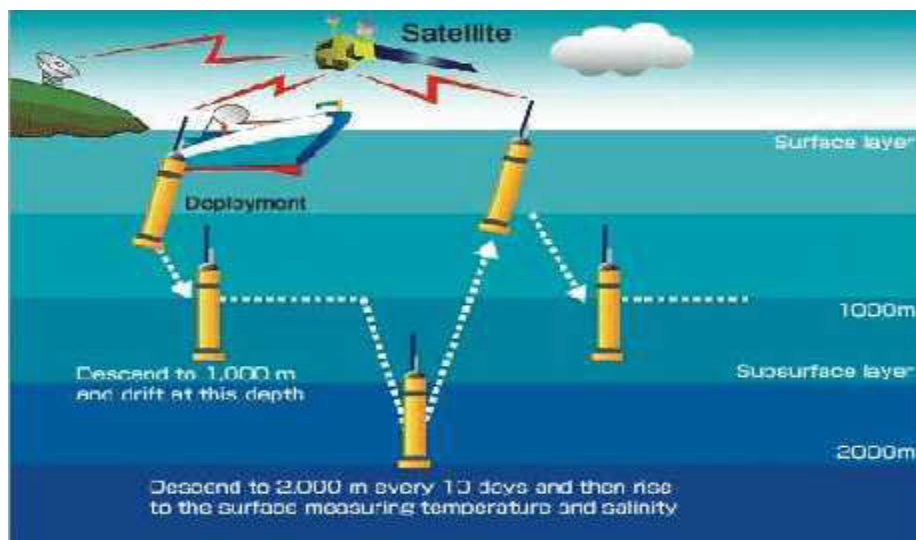


Figure I.14 : Application sur le contrôle de la qualité de l'eau.

I.6.4. 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.6.5. Applications médicales



Figure I.15 : Le flux d'information d'un patient

I.7. 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. [8]

1.7.1. 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.

1.7.2. 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.

1.7.3. 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.

1.7.4. 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 nœud dans son environnement. L'algorithmique distribué est la science la plus adaptée pour résoudre les problèmes du passage à l'échelle.

1.7.5. 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.

1.7.6. 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 nœuds 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.

1.6.7. 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.

1.7.8. 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. [8]

1.8. 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, tout comme les réseaux Ad Hoc, souffrent de nombreuses limitations en termes de performance, de fait du manque d'infrastructure et de la nature du médium 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 RCSFs 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 RCSFs

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 RCSFs, 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.

II.2.1. Les type de routage

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]

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. [9]

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.

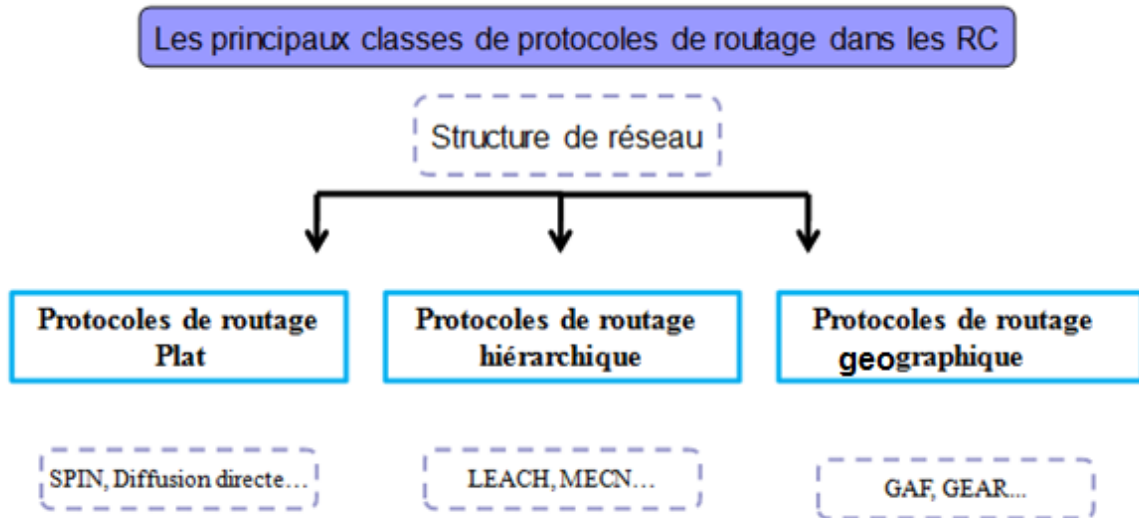


Figure II.1 : Les classes de protocoles de routage [10]

II.2.2.1 Les protocoles de routage plat

L'objectif principal d'un protocole de routage pour un réseau de capteur sans fil 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 plat (non hiérarchique) dans les RCSFs car la présentation de ces protocoles nous permettra de mieux analyser l'avantage de l'approche hiérarchique surtout dans les grands réseaux.

a) SPIN (Sensor Protocol for Information via Negotiation)

Le protocole SPIN 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.

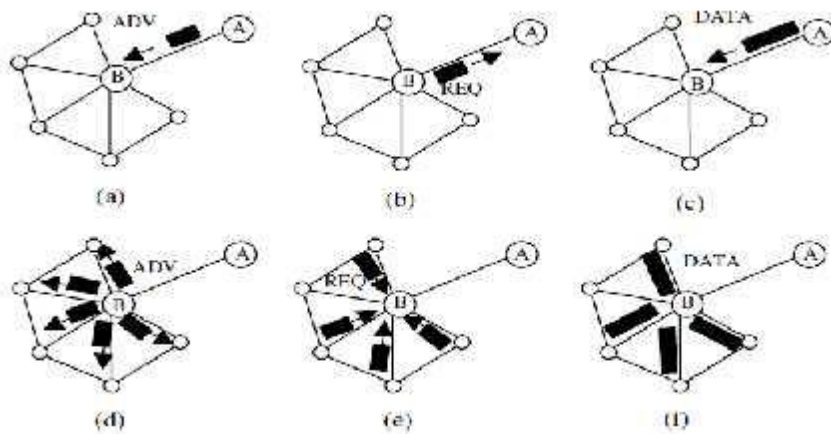


Figure II.2 : Fonctionnement du protocole SPIN . [10]

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 [10].

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.3). 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. 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. [10]

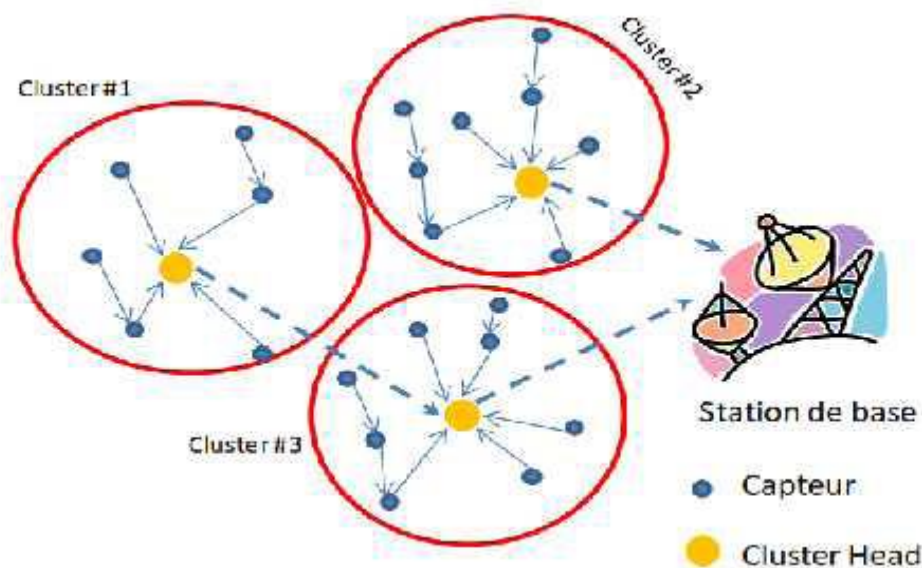


Figure II.3: Architecture en cluster [10]

b) ZHLS (Zone-based Hierarchical Link State Protocol)

Le protocole ZHLS est basé sur la décomposition du réseau en un ensemble de zones disjointes. Dans ce protocole, les membres d'une zone n'élisent pas de représentants contrairement à d'autres protocoles hiérarchiques. ZHLS utilise la technique GPS5 afin que chaque nœud sache sa position et la zone dans laquelle il se trouve. Avec cette décomposition, on a deux niveaux de topologies : le niveau nœud, et le niveau zone. La topologie basée sur le premier niveau renseigne sur la manière dans laquelle les nœuds d'une zone donnée sont connectés physiquement. Un lien virtuel peut exister entre deux zones, s'il existe au moins un nœud de la première zone, qui soit physiquement connecté à un nœud de l'autre zone (voir la *Figure II.4*). La topologie basée sur le niveau zone, donne le schéma de la connexion des différentes zones. La taille de la zone dépend de la mobilité des nœuds, la densité du réseau, la puissance de transmission et les caractéristiques de propagation. Le protocole est proactif quand le nœud destinataire est dans la même zone que le nœud émetteur. D'autre part, il est réactif si le nœud destinataire n'appartient pas à la même zone que le nœud émetteur. [11]

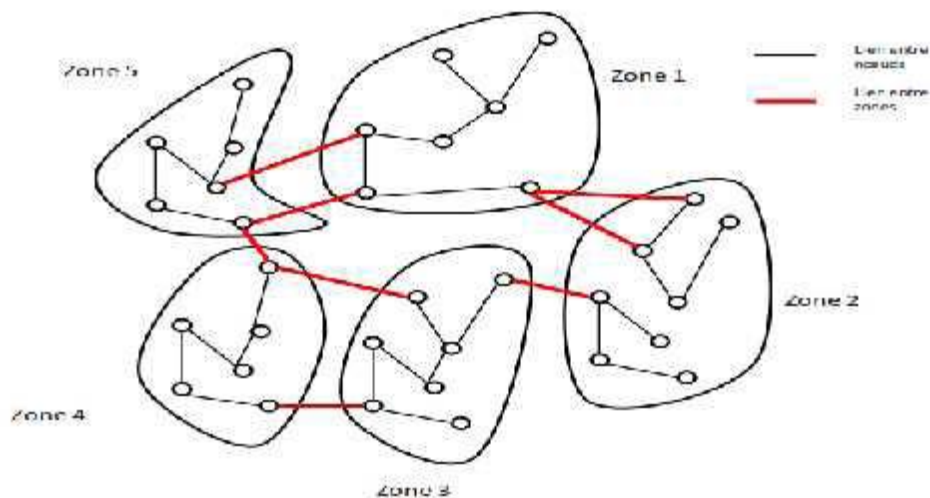


Figure II.4 : Décomposition du réseau en zones dans ZHLS [11]

Dans ce protocole, les paquets qui contiennent les états des liens ou les LSPs (Link State Packet) peuvent être divisés en deux classes : les LSPs orientés nœuds (NodeLSP) et les LSPs orientés zones (ZoneLSP). Pour un nœud donné, un paquet NodeLSP contient l'information d'un nœud voisin tandis qu'un paquet ZoneLSP contient l'information de la zone et il est échangé d'une manière globale. De cette façon, chaque nœud du réseau possède une connaissance complète concernant les nœuds de sa propre zone, et seulement une connaissance partielle du reste du réseau. Cette connaissance partielle est matérialisée par l'état de la connexion des différentes zones du réseau. [11]

a. Construction des tables de routage intra-zone

Chaque nœud diffuse une demande de vérification de liens (Link Request). Les nœuds appartenant à sa portée de communication répondent par un paquet contenant l'ID du nœud et sa zone. Après avoir reçu toutes les réponses, le nœud génère son paquet NodeLSP. Ce paquet NodeLSP contiendra les IDs des nœuds voisins dans sa zone et l'ID de la zone de ses voisins dans les différentes zones. Le paquet NodeLSP d'un nœud est transféré à tous les nœuds de la même zone. Les informations de l'ensemble de ces paquets NodeLSP seront stockées chez chaque nœud de la zone. Les informations des paquets venant des autres zones ne sont pas stockées puisqu'elles ne sont propagées que dans leurs propres zones. De cette façon, un nœud connaît la topologie basée sur le niveau nœud de sa zone (graphe LS). L'algorithme de Dijkstra (ou SPF, Shortest Path Algorithm) est utilisé pour construire la table de routage intra-zone. [11]

b. Construction des tables de routage inter-zone

Certains nœuds peuvent recevoir des réponses sur leurs demandes de vérification de liens (Link Request) des nœuds qui ne sont pas dans sa zone. Ces nœuds sont appelés des nœuds « Gateway ». Après l'échange des paquets NodeLSP dans la phase intra-zone, chaque nœud connaît les zones connectées à sa zone. A ce moment, chaque nœud de la même zone génère le même paquet ZoneLSP. Les nœuds « Gateway » envoient le paquet ZoneLSP à tous les nœuds du réseau. Chaque zone exécute la même procédure. Une liste de paquets ZoneLSP est stockée par chaque nœud du réseau. Par la suite, chaque nœud du réseau connaîtra la topologie du réseau (niveau zone). Une table de routage inter-zone est construite et SPF est utilisé pour trouver le plus court chemin. [11]

Les tables sont mises à jour périodiquement. Les nœuds « Gateway » ne diffuseront pas les paquets ZoneLSP si les nouvelles valeurs sont les mêmes que les anciennes. Le ZoneLSP n'est mis à jour qu'à la création ou lors de la césure d'un lien virtuel entre les zones. Ainsi, si un nœud « Gateway » reçoit deux copies du même ZoneLSP, il ne transfère qu'une seule, ce qui réduit le surcoût dans le réseau. [11]

Par conséquent, l'acheminement des données se fait de deux façons : le routage interzones, et le routage intra-zone. Pour une destination donnée, les données sont envoyées entre les zones en utilisant les identificateurs de zones jusqu'à ce que les données atteignent la zone finale de destination. Par la suite, les paquets de données circulent à l'intérieur de la zone finale, en utilisant l'identificateur du nœud destination. L'adresse < ID zone, ID nœud > ,

est suffisante pour atteindre n'importe quelle destination même si le réseau change de topologie.

c) Le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy)

LEACH est basé sur un algorithme de clustering distribué. C'est un protocole conçu pour le routage dans les réseaux de capteurs homogènes où les capteurs ont les mêmes caractéristiques et les mêmes capacités. Dans LEACH, les cluster-heads sont choisis aléatoirement et communiquent directement avec la station de base ce qui permet de minimiser la consommation et réduire la quantité d'informations envoyées à la station de base. [12]

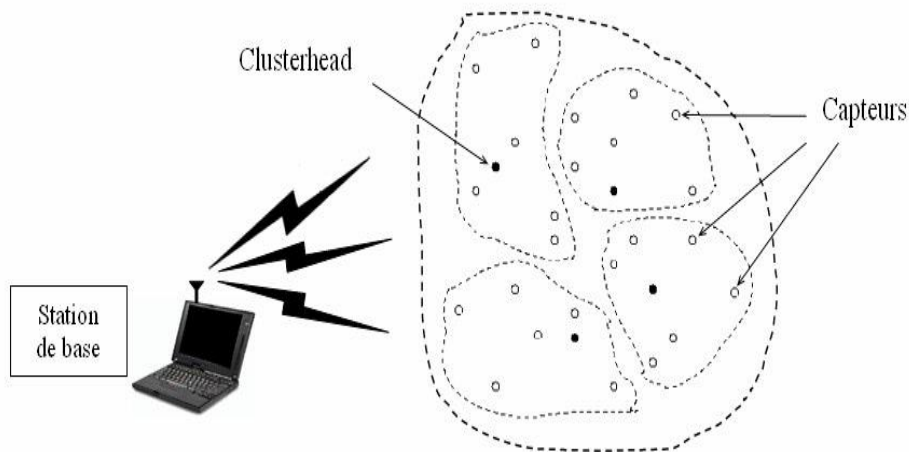


Figure II.5 : Le Clustering dans un réseau de capteurs [12]

LEACH s'exécute en deux grandes phases : la phase de configuration, « set-up phase » et la phase de transmission « steady-phase ».

✍ Phase d'initialisation (phase de configuration)

Comme l'indique la figure I.8, la phase d'initialisation est composée de 3 sous phases : d'annonce, d'organisation des groupes et enfin d'ordonnancement

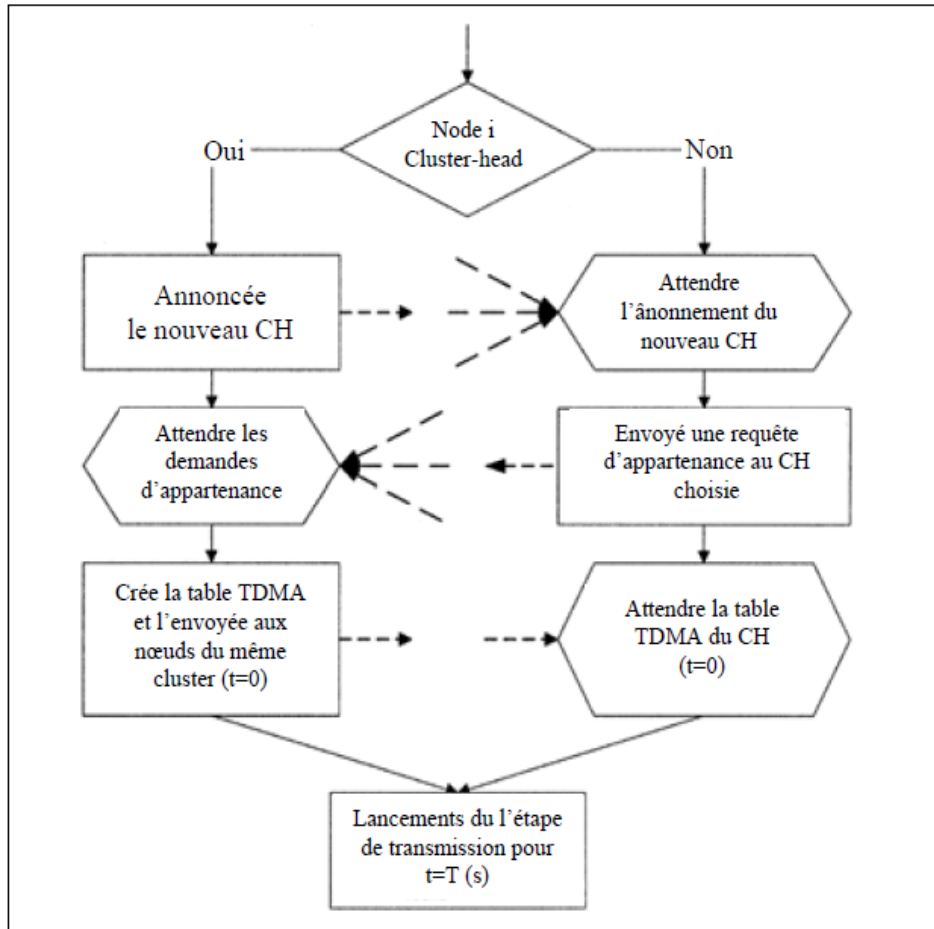


Figure II.6: Opérations de l'étape d'initialisation

Phase d'annonce

Cette phase commence par l'annonce du nouveau round par le nœud puits, et, par la prise de décision locale d'un nœud pour devenir CH avec une certaine probabilité $P_i(t)$ au début du round $r+1$ qui commence à l'instant t . Chaque nœud i génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à $P_i(t)$, le nœud deviendra CH durant le round $r+1$. $P_i(t)$ est calculé en fonction de K et de round r .

$$\text{Nombre (CH)} = \sum_{i=1}^N P_i(t) = K \quad (1.1)$$

Où N est le nombre total de nœuds dans le réseau. Si on a N nœuds et K CH, alors, il faudra N/K rounds durant lesquels un nœud doit être élu seulement une seule fois autant que CH avant que le round soit réinitialisé à 0. Donc la probabilité de devenir CH pour chaque nœud i

est :

$$P_i(t) = \frac{\text{nombre de CH désirés}}{\text{le nombre de noeuds qui n'ont pas encore été élus CH durant les rounds précédents}} \quad (1.2)$$

$$P_i(t) = \begin{cases} \frac{k}{N - k * (r \bmod \frac{N}{k})} & : C_i(t) = 1 \\ 1 & : C_i(t) = 0 \end{cases} \quad (1.3)$$

Où $C_i(t)$ égal à 0 si le nœud i a déjà été CH durant l'un des $(r \bmod N/K)$ rounds précédents, et, il est égal à 1 dans le cas contraire. Donc, seuls les nœuds qui n'ont pas encore été CH, ont vraisemblablement une énergie résiduelle suffisante que les autres et ils pourront être choisis.

Le terme $\sum_{i=1}^N C_i(t)$ représente le nombre total des nœuds éligibles d'être CH à l'instant t . Il est égal à

$$\sum_{i=1}^N C_i(t) = N - k * (r \bmod N/k) \quad (1.4)$$

Utilisant l'équation (1.3) et (1.4), le nombre de CH par round est :

$$_{(CH)} = \sum_{i=1}^N P_i(t) * C_i(t) = \left(K * \left(r \bmod \frac{N}{K} \right) \right) * \left(\frac{K}{N - K * (r \bmod \frac{N}{K})} \right) = K \quad (1.5)$$

La probabilité $P_i(t)$ est basée sur la supposition que tous les nœuds sont initialement homogènes et commencent avec la même quantité résiduelle d'énergie et meurent approximativement en même temps. Cependant, ceci pourrait être le cas juste après le déploiement, mais il n'est pas réellement valable après un certain temps. Alors, si l'énergie des nœuds diffère, il sera plus pratique que la probabilité $P_i(t)$ soit en rapport avec l'énergie restante au niveau de chaque nœud. Cette probabilité sera donc égale à:

$$P_i \text{ (1.6)} = \frac{E_i(t)}{E_{\text{total}}(t)} K$$

Où $E_i(t)$ est l'énergie résiduelle relative à chaque nœud i . Utilisant cette probabilité, le nœud avec une plus grande ressource d'énergie a une plus grande chance de devenir CH. Ainsi, le nombre de nœuds souhaités pour être CH dans chaque round est:

$$\text{Nombre } CH = \sum_{i=1}^N P_i(t) * C_i(t) = \left(\frac{E_1(t)}{E_{\text{total}}(t)} + \frac{E_2(t)}{E_{\text{total}}(t)} + \dots + \frac{E_n(t)}{E_{\text{total}}(t)} \right) \text{ (1.7)}$$

Les équations (1.4) et (1.6) seront égales si les nœuds commencent avec la même énergie. De plus, en utilisant l'équation (1.6), les nœuds requièrent des informations sur toute l'énergie disponible dans le réseau.

✍ Phase d'organisation de groups

Après qu'un nœud soit élu CH, il doit informer les autres nœuds non-CH de son nouveau rang dans le round courant. Pour cela, un message d'avertissement ADV contenant l'identificateur du CH est diffusé à tous les nœuds non-CH en utilisant le protocole MAC

CSMA pour éviter les collisions entre les CH. La diffusion permet de s'assurer que tous les nœuds non-CH ont reçu le message. Par ailleurs, elle permet de garantir que les nœuds appartiennent au CH qui requière le minimum d'énergie pour la communication. La décision est basée donc sur l'amplitude du signal reçu; le CH ayant le signal le plus fort (i.e. le plus proche) sera choisi. En cas d'égalité des signaux, les nœuds non-CH choisissent aléatoirement leur CH. [13]

✍ Phase d'ordonnement

Après la formation des groupes chaque CH agit comme un centre de commande local pour coordonner les transmissions des données dans sa grappe. Il crée un modèle de communication en se basant sur TDMA, ensuite il le transmet aux nœuds de sa grappe (cluster). Etant donné que chaque nœud connaît d'avance le slot de temps qu'il va occuper, cela permet alors au nœud de passer à l'état "endormi" durant les slots inactifs. Ainsi, la perte d'énergie due aux états de sur écoute (overhearing) et d'écoute passive (idle) est évitée.

Chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA, il le transmet aux nœuds appartenant à son cluster afin de l'utiliser pour leurs transmissions, ceci afin de minimiser les interférences entre les messages des CHs les plus proches. [13]

✍ phase de transmission

La phase de transmission les nœuds envoient leurs données au cluster-head du cluster auquel ils appartiennent, ensuite le cluster-head agrège toutes les données et envoie le paquet de données à la station de base. Au début de la « steady phase », un ordonnancement TDMA (Time Division Multiple Access) permet d'assigner à chaque nœud un slot de temps pour la transmission de ses données. Dans le souci d'économiser l'énergie, les nœuds simples, qui ne sont pas cluster-head sont actifs uniquement durant leur temps de transmission, le reste du temps ils mettent leur radio en veille. [13]

Après un intervalle de temps donné prédéterminé, le réseau va passer à un nouveau round. Ce processus est répété jusqu'à ce que tous les nœuds du réseau seront élus CH, une seule fois, tout au long des rounds précédents. Dans ce cas, le round est réinitialisé à 0.

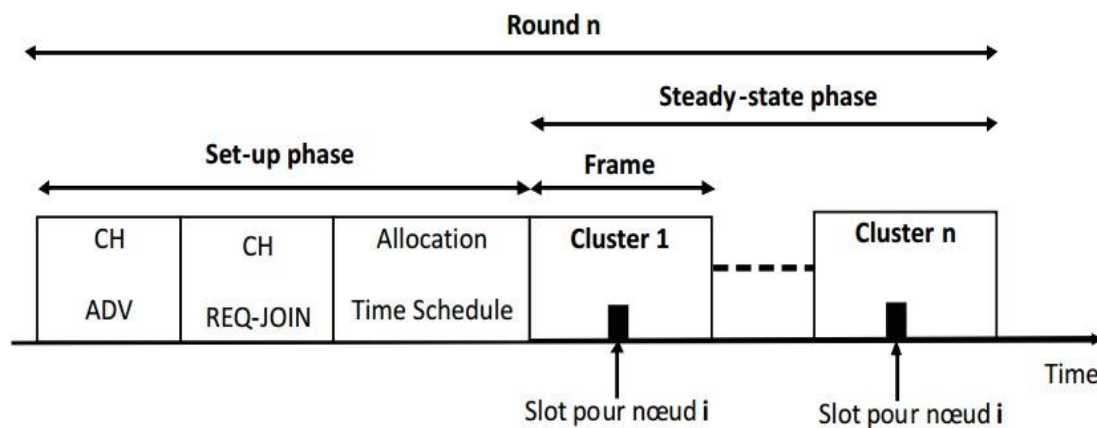


Figure II.7: Répartition du temps et différentes phases pour chaque round [13]

d) 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. [14]

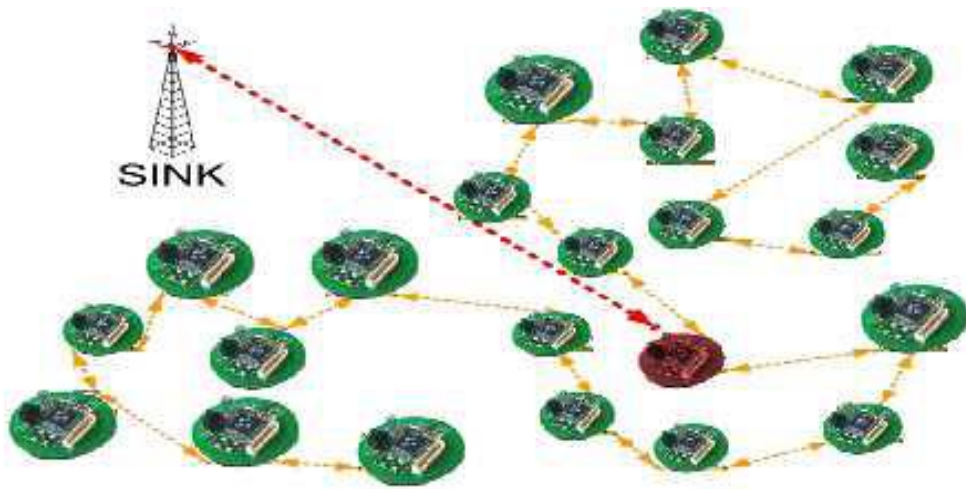


Figure II.8: Formation des chaînes gourmandes dans PEGASIS [14]

Algorithme de la chaîne Greedy

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

La fusion de données à chaque nœud (sauf nœuds d'extrémité)

- ✍ Un seul message est transmis à chaque nœud
- Calcul Delay: N unités pour un nœud réseau N
- ✍ Transmission séquentielle est supposée
- Nœud 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. 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. [15]

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.

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

Manjeshwar et Agrawal [16] 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.

f) 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 [17]. 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, 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.

g) HEED (Hybrid, Energy-Efficient, Distributed approach)

Younes et Fahmy [18] 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 du cluster heads 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 :

(1.8)

$$P_{ch} = C_{prob} (E_n/E_{total})$$

Où E_n est l'énergie restante du nœud n , E_{total} est l'énergie initiale dans le réseau et C_{prob} est 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 intracluster et les clusters générés ne sont pas équilibrés en taille. [19]

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). [20]

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 :

a) 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.

b) 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 termes 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.

c) 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. [20]

II.3. Conclusion

Dans les réseaux de capteurs 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 (FireFly Algorithm)

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. [19]

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. [19]

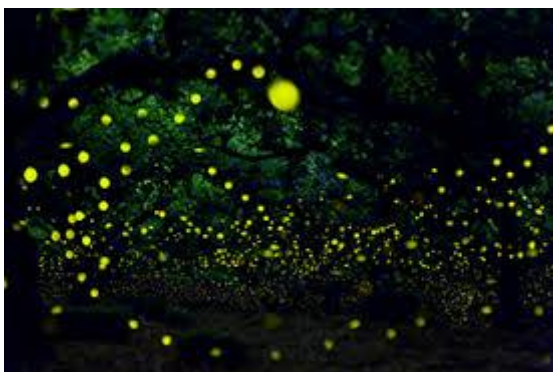


Figure III.1 : Les Lucioles

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 unisexe, 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). [20]
 - 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. [36]
 - 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.

Chapitre II : Algorithme des Lucioles (Firefly)

Les principales étapes de l'algorithme Firefly sont données Ci-dessous :

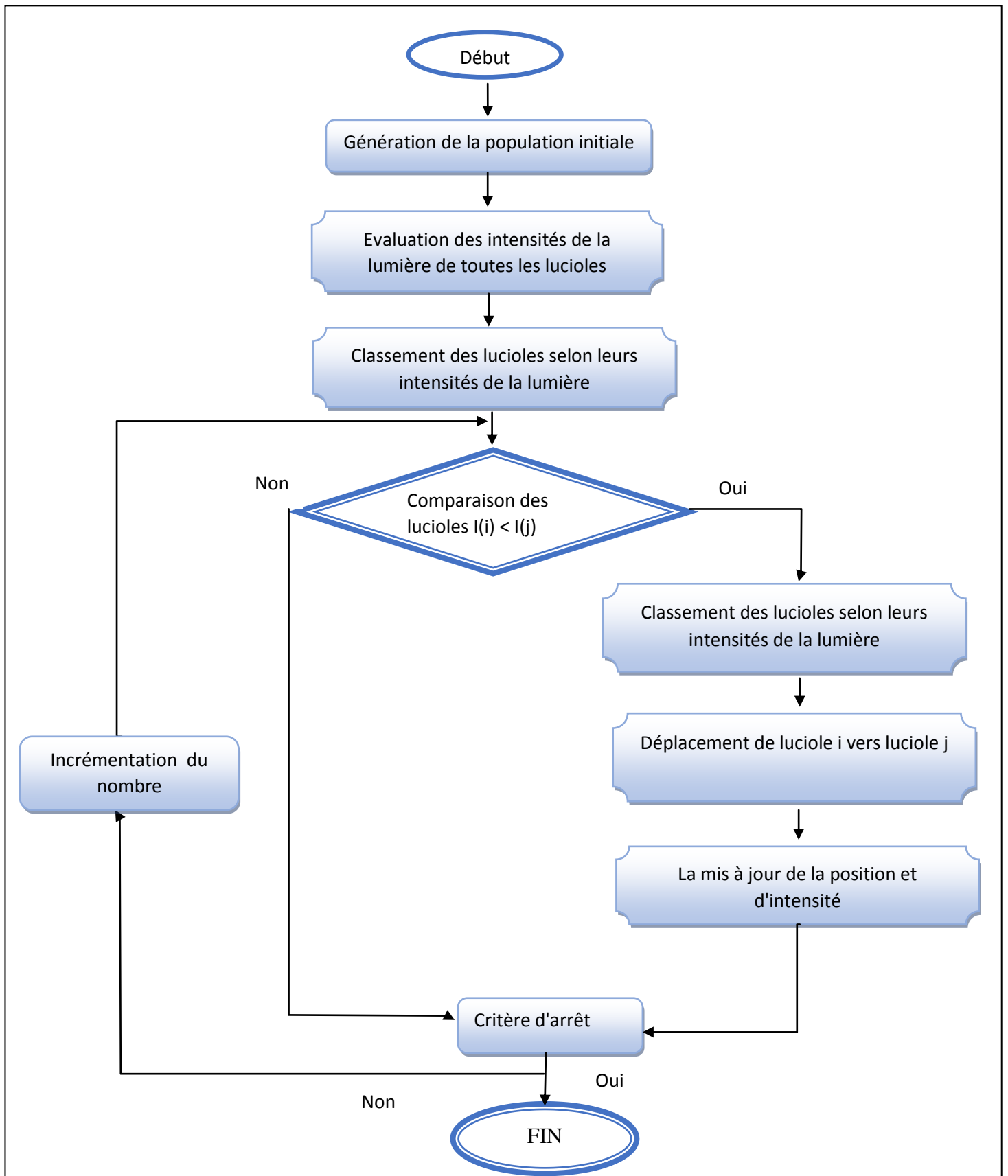


Figure III.2: l'organigramme d'algorithme de luciole

2.3. Description de l'organigramme

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. [21]

2.3.2. Fonction d'évaluation

La fonction d'évaluation (fitness), attribut à chaque luciole une valeur numérique qui représente un coût de performance, elle est utilisée pour coder la luminosité des Fireflies. Grâce 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. [21]

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

a) 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.

b) 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.

2.3.3. 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.

2.3.4. 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. [21]

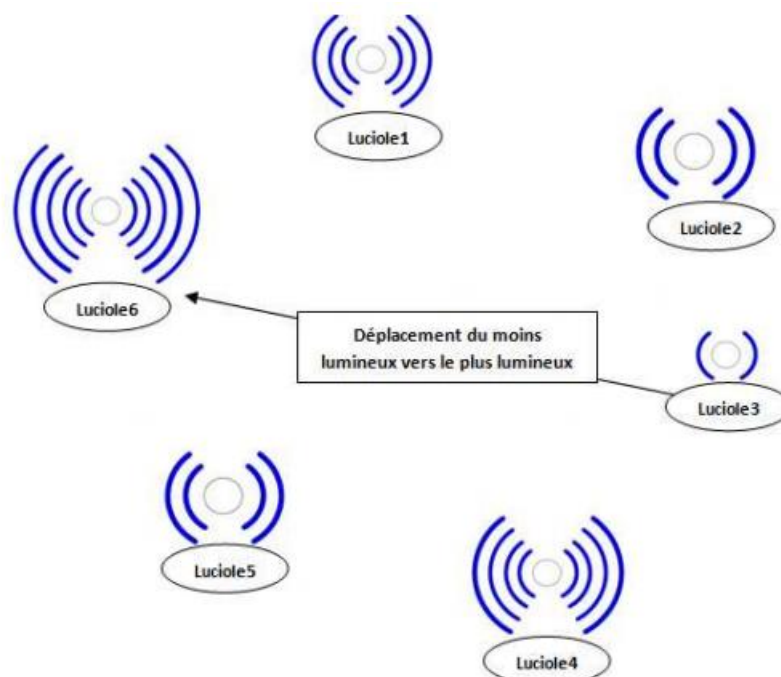


Figure III.3: Déplacement des lucioles

2.3.5. 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 MaxGénération : le maximum nombre de génération
- 4 γ : 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 γ

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.3: l'algorithme des Lucioles

2.4. 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. [22]

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 [22], 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. [22]

✓ L'intensité et L'attractivité

a. 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-She Yang suppose que l'attraction d'une luciole est déterminée par sa luminosité qui à son tour est associée à la fonction d'objectif. [23]

Dans la forme la plus simple, l'intensité lumineuse $I(r)$ varie en fonction de la loi du carré inverse $I(r) = I_0 / r^2$ où I_0 est l'intensité à la source, ou dans un milieu donné avec une lumière fixe γ coefficient d'absorption, l'intensité lumineuse varie avec la distance r , qui est $I(r) = I_0 e^{-\gamma r}$, Afin d'éviter la singularité à $r = 0$ dans l'expression $I(r) = I_0 / r^2$, l'effet combiné de la fois le loi carrée inverse et l'absorption peuvent être estimés à l'aide de la forme suivante :

$$I(r) = I_0 e^{-\gamma r^2} \quad (1.1)$$

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

b. 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 β d'une luciole peut être définie comme :

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{1.2}$$

Où β_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 β_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 : [40]

$$\beta(r) = \beta_0 e^{-\gamma r^2}$$

✓ **Coefficient d'absorption**

Le coefficient d'absorption γ 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. [23]

✓ **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 :

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \tag{1.3}$$

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 :

$$r_{ij} = \sqrt{(y_i - y_j)^2 + (x_i - x_j)^2} \quad (1.4)$$

✓ **Mouvement**

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

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha (\text{rand} - 1/2) \quad (1.5)$$

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 α est un paramètre de répartition aléatoire, tandis que rand est un générateur de nombre aléatoire distribué de façon uniforme dans l'espace $[0,1]$. [23]

✓ **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.3. Conclusion

L'algorithme des lucioles "Firefly Algorithm" permet de fournir rapidement des solutions qui sont proches 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 à 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 allons présenter dans les chapitres précédents l'une des contraintes majeures dans la conception des RCSF : l'énergie. Le déploiement des RCSFs dans des zones non surveillées et dont l'intervention humaine est minimale ou quasiment impossible, exige des capteurs un effort important pour survivre, et les ressources des nœuds de capteurs sont assez limitées.

Dans ce dernier chapitre nous allons évaluer le nouveau protocole AFABC qui est une version modifiée de l'algorithme des lucioles (appelé Firefly algorithm), pour bien minimiser la distance totale de transmission dans le réseau à l'aide du clustering ce qui réduirait aussi le nombre de transmissions, 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, nous allons le comparer avec un autre protocole de routage.

IV.2. Description générale de l'algorithme proposé

IV.2.1 Algorithme AFABC (Adaptive Firefly Algorithm Base Clustering)

Notre protocole est basé sur un algorithme de luciole, qui peut être utilisé pour déterminer la hiérarchie des clusters pour envoyer des données à la station de base. Cette section traite l'algorithme de luciole adapté pour la construction de ce nouveau protocole:

IV.2.1.1 Formation des CHs

✓ Algorithme 1R-CH (1 Round-Clustering Hiérarchique)

Initialement, les CHs sont choisis selon un facteur énergétique (F_T) calculé comme suit :

$$F_T = \frac{E}{E_t} \quad (3.1)$$

E : Énergie initiale ;

E_t : Énergie consommée pour transmettre un paquet vers la station de base.

Chaque nœud élu pour devenir CH, cherche parmi ses voisins, s'il y a un nœud CH, si c'est le cas, celui qui a le plus grand facteur F_T devient le CH de ce voisinage, sinon ce nœud est élu CH. Si le nœud élu n'a aucun voisin alors il devient un CH.

Algorithme 1R-CH

N: nombre des nœuds, $X_i(i=1 \dots N)$

NBV: le nombre des voisins, $V_j(1 \dots NBS)$

E: énergie initial

E_T : énergie de transmission ver le station de base

Début

Pour chaque nœud N faire

1. calculer le factor énergétique des donnees vers le SB:

$$F_T = \frac{E}{E_T}$$

3. Calculer le nombre de ces voisins

3.1 Si $NBV = 0$ Alors

$X_i = CH$

Sinon

3.2 Si $V_j(X_i)$ est un CH

$V_j(X_i) = CH$

Sénon

3.2 $X_i = CH$

Fin si

Fin si

Fin pour

Fin.

Figure IV.1: formation des CHs avec le protocole 1R-CH

Dans les autres rond on déterminer la CHs à l'aide d'équation suivant :

$$x_i = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha * (rand - 1/2) \quad (3.2)$$

Où x_i est la position du CH et X_j est l'emplacement du nœud voisine et que la coordonnée x est pris en considération pour le calcul de l'intensité de référence, r_{ij} est la distance entre un groupe CH et un nœud voisin, calculée selon l'équation de la distance euclidienne et ϵ est $(rand-0.5)$. β , γ et α sont les paramètres qui sont rand réglable. [24]

IV 2.1.1.2 Algorithme AFABC

Algorithme AFABC

γ : Coefficient d'absorption

r_{ij} : La distance entre CH et le nœud simple

I_0 : L'intensité initial

$max_generation$: Le maximum génération

Debut

1. Génération de population initiale

IR-CH ()

Répéter jus qu'à ce qu'a ($t+1 < max_generation$)

Pour $i=1$ à CH do

Pour $j=1$ à NBV do

2 $x_i = x_i + \beta e^{-\gamma r^2} (x_j - x_i) + (rand - 1/2)$

3. Calculer l'intensité $I(x_i) = I_0 e^{-\gamma x_i^2}$

Si $I_i < I_j$

4. $j = CH$

Sinon

$i = CH$

Fin si

Fin pour

Fin pour

Calculer la fonction objective

Fin

Visualiser le résultat

Figure IV.2 : Algorithme AFABC

IV 2.1.2. Fonction objective

L'énergie utilisée pour l'agrégation des données ainsi que la distance totale de transmission, sont les principaux facteurs qu'on cherche à minimiser, cette fonction utilisée pour déterminer pour chaque itération la meilleure configuration dans le réseau.

$$\text{Fonction objective} = \beta \times f1 + (1 - \beta) \times f2 \quad (3.3)$$

$$f1 = \max_{k=1,2..k} \{ \sum \forall N_i \in C_{p,k} d(N_i, CH_{p,k}) / C_{p,k} \} \quad (3.4)$$

$$f2 = \frac{\sum_{i=1}^N E(N_i)}{\sum_{k=1..K} E(CH_{p,k})} \quad (3.5)$$

Ou $f1$: est la moyenne maximale euclidienne distance de nœuds à leurs CH de cluster associés.

$C_{p,k}$: Le nombre des nœuds qui appartiennent à C_k des particules p .

$f2$: est la fonction qui est le rapport entre l'énergie totale initiale de tous les nœuds.

β : est un utilisateur défini constant pour peser la contribution de chacun des sous-objectif

Après la formation des clusters, le réseau entre dans la phase de l'état d'équilibre, où les nœuds commencent effectivement à transmettre leurs valeurs détectées 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.

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

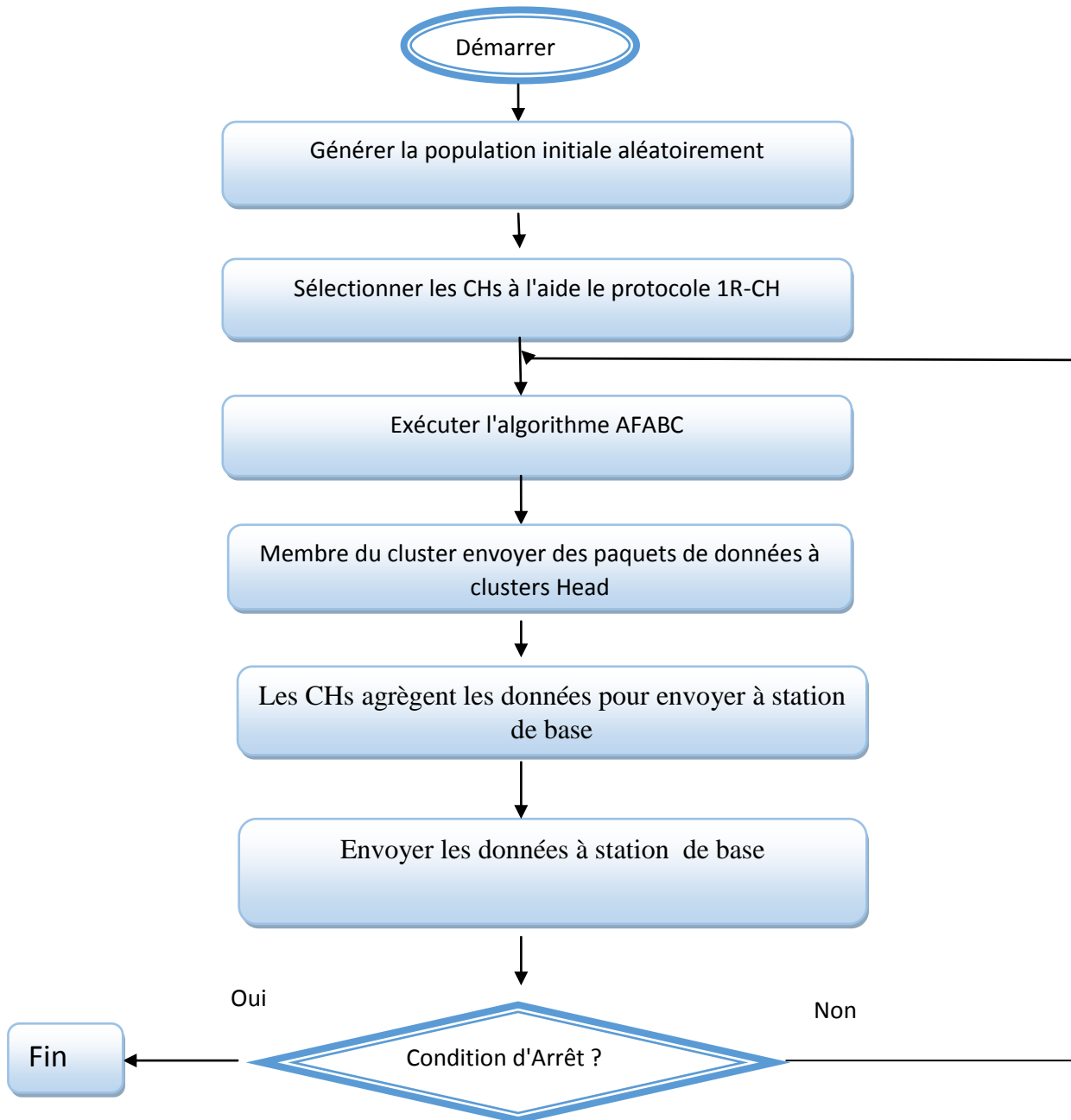


Figure IV.3 : l'organigramme de protocole AFABC

IV 3. Choix du langage et de l'environnement d'implémentation

Nous avons choisi comme langage d'implémentation pour notre protocole, on a utilisé le MATLAB (MATrix LABoratory), 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. [25]

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 (current folder). d. En bas à droite l'historique des commandes tapées (command history). [26]

IV .4 . Les étapes d'exécution de l'application

IV 4.1 Lancement d'application

Premièrement présentée l'interface d'application et après avoir appuyé sur le bouton « Enter » pour accéder à la partie de simulation.

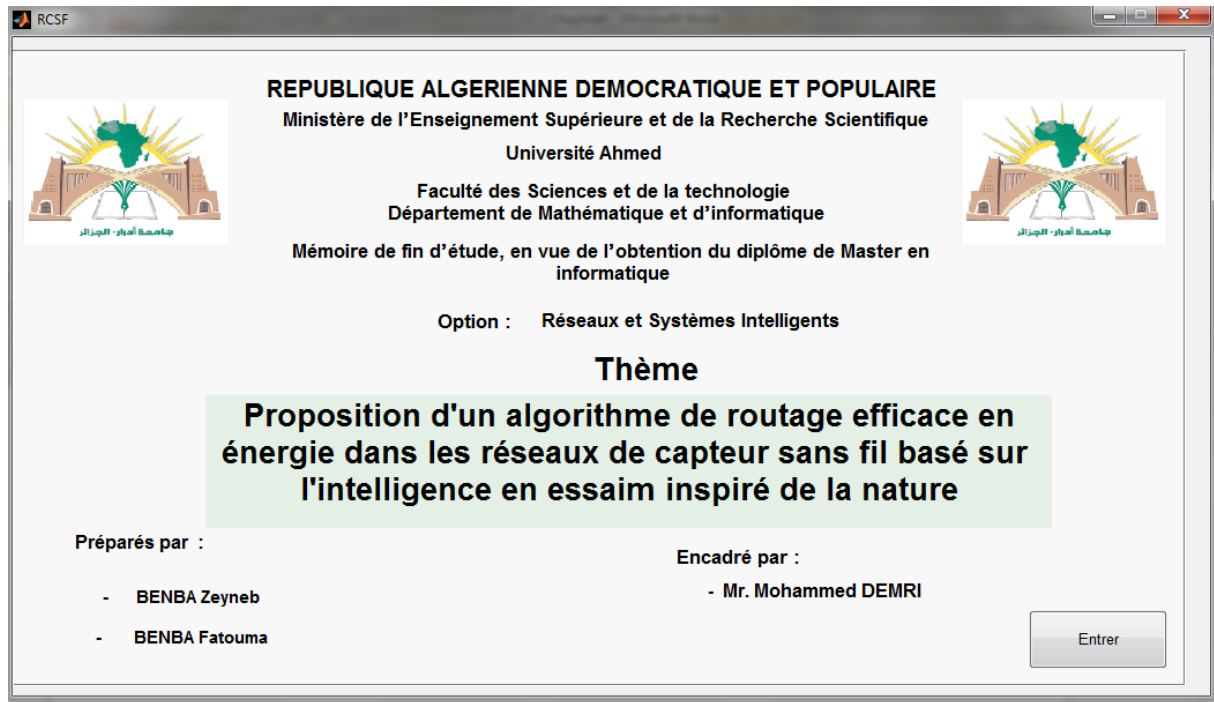


Figure IV.4 : L'interface initiale d'application

IV 4.2 Description et paramètres de simulation

les différents paramètres nécessaires pour la mise en œuvre 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 à bien adapter la configuration le réseau et l'algorithme proposé par rapport à l'objectif de notre travail.

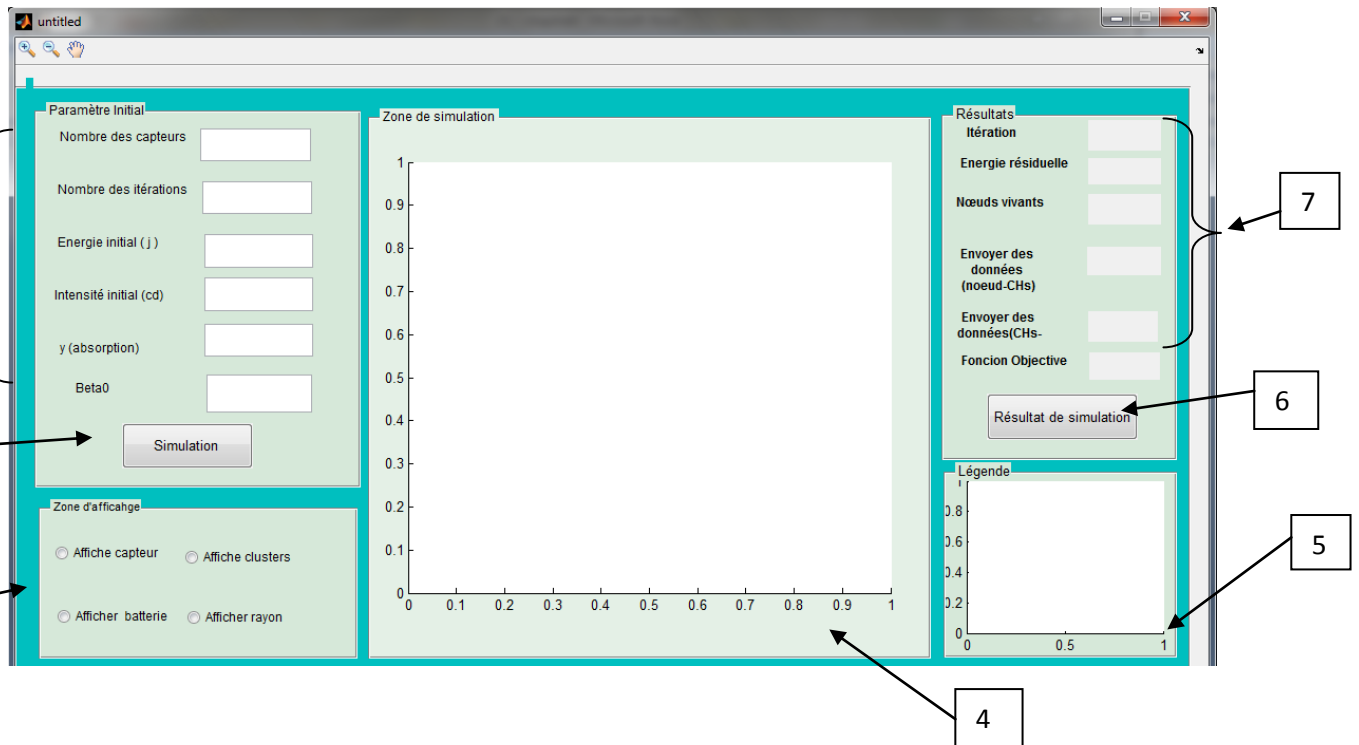


Figure IV.5: L'interface Principale de simulation

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

4 .Paramètres de simulation

4.1 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. La figure 1 présente un modèle d'antenne et les règles de consommation d'énergie associées. [27]

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

$$E_{Tx}(k, d) = E_{Tx}(l) + E_{Tx_amp}(k, d) \quad (3.6)$$

$$E_{Tx}(k, d) = \begin{cases} k.E_{elec}(k, d) + k.\mathcal{E}_{friss}.d^2 \\ k.E_{elec}(k, d) + k.\mathcal{E}_{two_ray_amp}.d^4 \end{cases} \quad (3.7)$$

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

$$E_{Rx}(k) = E_{Rx_elec}(k) = k.E_{elec} \quad (3.8)$$

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;

ϵ_{amp} : Facteur d'amplification;

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

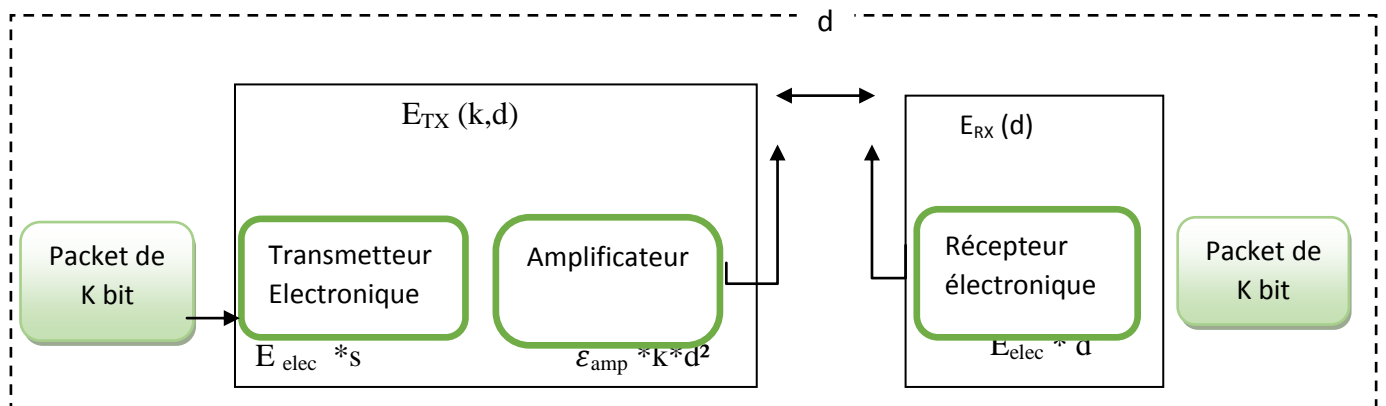


Figure IV.6: Modèle de consommation d'énergie pour la communication

- **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 [28]. 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 :

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

4.2 Paramètres d'algorithmes des lucioles

la population initiale présente l'ensemble des lucioles, la simulation de dans le réseau est réalisé par l'algorithme AFABC, alors il faut adapter les paramètres suivants :

- Le nombre de nœuds ;
- Le nombre d'itération ;
- Energie initiale de chaque nœud;
- L'intensité initiale;
- L'attraction initiale;
- coefficient d'absorption;

4.3 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 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.

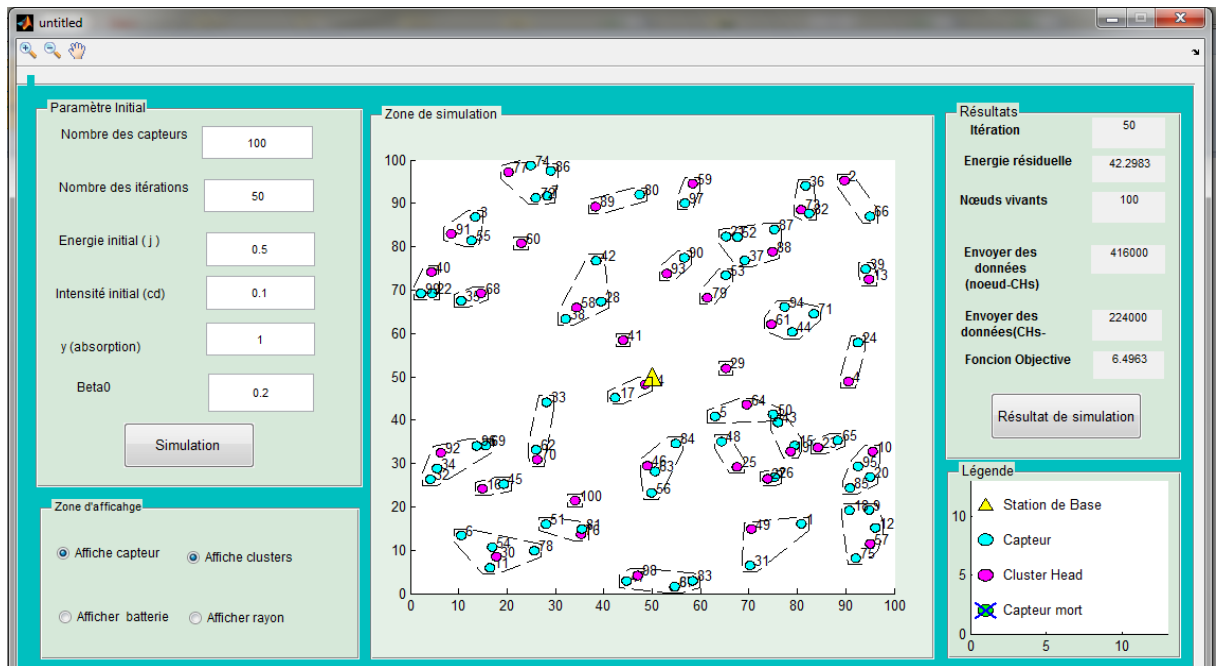


Figure IV.6 : La simulation

Ensuite, observer les résultats de la simulation . après avoir appuyé sur le bouton « Résultats de simulation».

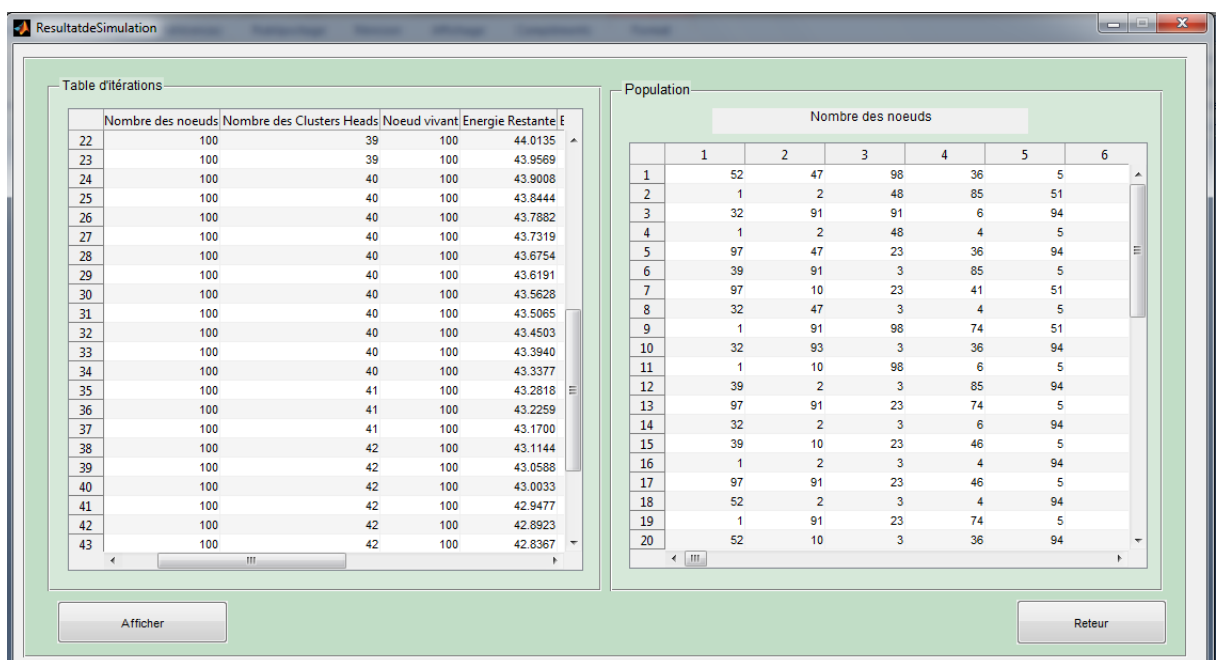


Figure IV.7 : Résultats de simulation

IV.5. Simulation et résultat

Dans cette section, nous allons présenter les résultats de simulation que nous avons obtenus. Ces résultats valident 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 Fire-LEACH. [29]

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 de la littérature, à savoir LEACH et FIRE-LEACH [29] 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

Zone de couverture	100*100 m²
Nombre de nœuds	100
L'énergie initiale des noeuds	0.5 J
Langur de paquet	6400
Eelec	50 nJ/bit
Efs	10 pJ/bit/m ²
Emp	0.0013 pJ/bit/m ²
Eda	5 pJ/bit/sig
Type de distribution	aléatoire

Tableau IV.1: Les paramètres de simulation

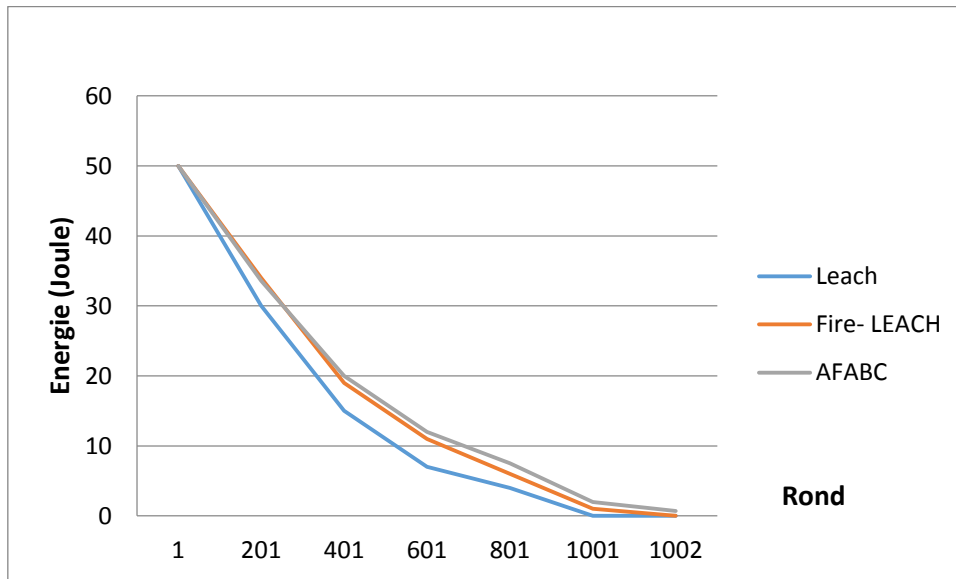
Intensité initiale des lucioles I_0	0.5
Coefficient d'absorption	1
β	0.2
α	paramètre de répartition aléatoire

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 Fire-LEACH, LEACH et le protocole AFABC par rapport chaque rond.

Rond	Leach	Fire- LEACH	AFABC
1	50	50	50
201	30	34	33,6
401	15	19	20
601	7	11	12
801	4	6	7,5
1001	0	1	2
1201	0	0	0,7

Le tableau IV.3: la consommation d'énergie après 1201 chaque rond.



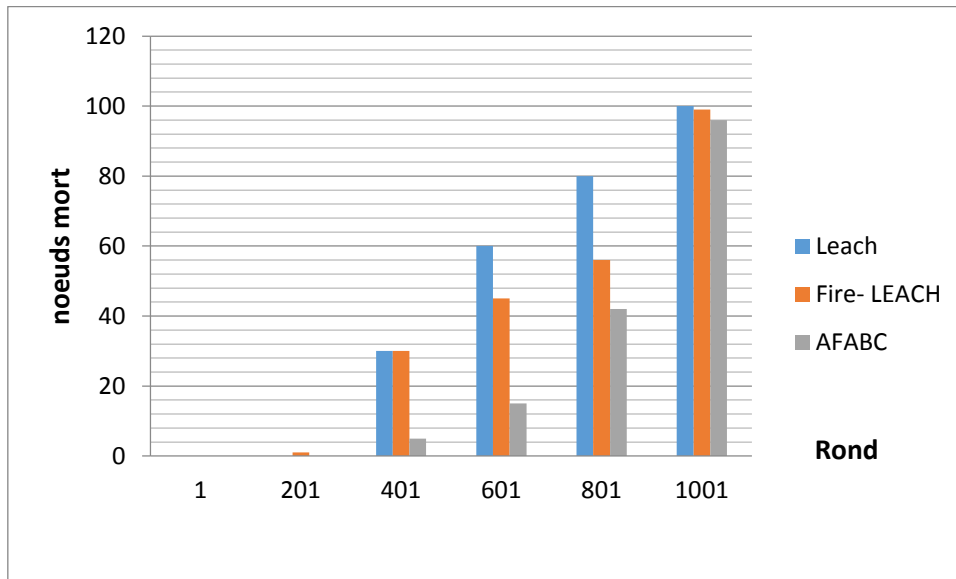
La figure IV.3: la consommation d'énergie après chaque rond.

On remarque que pour les trois protocoles (LEACH, FIRE-LEACH et AFABC) la valeur d'énergie consommée est diminuée avec l'augmentation des nombre des ronds du réseau. Après 1201 rond le protocole LEACH et FIRE-LEACH, L'énergie est consommée entièrement, par contre dans le protocole AFABC le réseau elle reste vivant.

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

	Leach	Fire- LEACH	AFABC
1	0	0	0
201	0	1	0
401	30	30	5
601	60	45	15
801	80	56	42
1001	100	99	96

Le tableau IV.4: nombre de nœuds morts après chaque rond

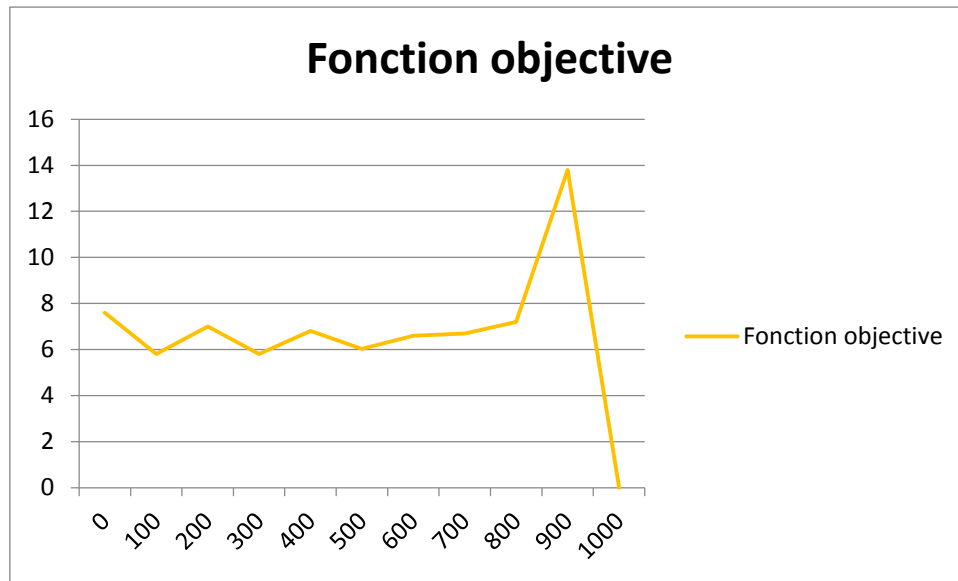


La Figure IV.4: nombre des nœuds mort.

Le tableau IV.5 représente les valeurs de notre fonction d'affinité ou moment de fonctionnement de réseau.

Ronde	Fonction objective
0	7,6
100	5,8
200	7
300	5,8
400	6,8
500	6,03
600	6,6
700	6,7
800	7,2
900	13,8
1000	0

Le tableau IV.5 représente les valeurs de fonction objective.



La Figure IV.5: valeurs de fonction objective après chaque rond

IV.7. Conclusion

Dans ce chapitre, nous avons exposé les résultats de la simulation de notre protocole AFABC, effectuée sur le simulateur MATLAB13. Les résultats des simulations effectuées montrent clairement que le protocole AFABC permet de meilleures prestations par rapport au protocole Fire-LEACH et LEACH, en ce qui concerne le nombre, la taille et la distribution sur le réseau des clusters générés. Le protocole AFABC permet également une augmentation du gain d'énergie de l'énergie, qui prolonge la durée de vie.

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 AFABC. 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 AFABC avec le simulateur MATLAB 13 et comparé avec deux autres protocoles de la littérature à savoir LEACH et Fire-LEACH. 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...);
- ✓ Une hybridation avec l'algorithme CR (cuckoo search) qui a été développé par Yang ;
- ✓ L'évitement toutes sorte d'envois à des longues distances (d'un CHs vers la station de base) en intégrant une technique de routage multi-sauts (d'un CH vers ses voisins jusqu'à la station de base)
- ✓

Références Bibliographiques

- [1]. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. "Wireless sensor networks : a survey. Computer Networks (Elsevier) ", vol.38, no.4, March 2000, pp.393-422.
- [2]. Yazeed Al-Obaisat, Robin Braun "On Wireless Sensor Networks: Architectures, Protocols, Applications, and Management" Institute of Information and Communication Technologies University of Technology, Sydney, Australia.
- [3]. Yasser Romdhane « Evaluations des performances des protocoles S-MAC et directed diffusion dans les réseaux des capteurs » rapport de projet de fin d'étude, école supérieure des communications de Tunis, 2007.
- [4]. Yacine CHALLAL « réseau de capteur sans fil », support de cours, 17/11/2008.
- [5]. HILL (J.-L.), « l'architecture système pour les réseaux de capteurs sans fil », thèse à l'Université de Californie à Berkeley, 2003.
- [6]. Brown, M. J. Users Guide Developed for the JBREWS Project. Los Alamos National Laboratory of California University. 1999. Technical report LA-UR-99-4676.
- [7]. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks. IEEE Communications Magazine, vol 40, pp. 102-114, August, 2002.
- [8]. Michael Fitzgerald. Technnology Review : Tracking a Shopper's Habits. Technology Review. [En ligne] 04 August 2008.
- [9]. Anis Laouiti, Cédric Adjih. Mesures de performances du protocole OLSR. Projet Hipercom. 2003. Rapport technique.
- [10].] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. 5th annual ACM/IEEE international conference on Mobile computing and networking . August 1999, pp. 174 - 185.
- [11]. Hamma, T. Katoh, T. Bista, B.B. Takata, T. An Efficient ZHLS Routing Protocol for Mobile Ad Hoc Networks. 17th International Conference on Database and Expert Systems Applications. 2006, pp. 66-70.
- [12]. V. Kawadia and P. R. Kumar, "Power Control and Clustering in Ad Hoc Networks", IEEE INFOCOM. (2003).

Références Bibliographiques

- [13]. T. Kwon and M. Gerla, "Clustering with Power Control", In Proceedings MILCOM '99, volume 2. (1999).
- [14]. Lindsey, S. Raghavendra, C.S. PEGASIS: Power-efficient gathering in sensor information systems. IEEE Aerospace Conference Proceedings. 2002, Vol. 3, pp. 3-1130.
- [15]. Lindsay, S., Raghavendra, C. S., and Sivalingam, K. M. Data Gathering in Sensor Networks using the Energy Delay Metric. [éd.] IEEE Computer Society. 15th international Parallel & Distributed Processing Symposium. April 23 - 27, 2001 2001, p. 188.
- [16]. 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,
- [17]. 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.
- [18]. 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.
- [19]. 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.
- [20]. Yang X.S, "Nature-Inspired Metaheuristic Algorithms". Luniver Press, UK. 2008.
- [21]. Romana CAPOR-HROSIK, Adis ALIHODZIC, Milan TUBA, Mirjana VUKOVIC, Milenko PIKULA, "Firefly Algorithm for Constrained Optimization Problems". ISBN: 978-960-474-330-8.
- [22]. Xin-She Yang, "Nature-Inspired Metaheuristic Algorithms", 2nd Edition Copyright © 2010 Luniver Press.
- [23]. Xin-She Yang Department of Engineering, University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK “ Firefly Algorithms for Multimodal Optimization ” 2010, pp. 4–6.
- [24]. Xin –She Yang, “ Nature- Inspired Metaheuristic Algorithms: Second Edition”, Luniver Press, UK, 2008.

Références Bibliographiques

[25]. Cleve Moler October 4, 2011 " Experiments with MATLAB" , Electronic edition published by MathWorks, Inc, <http://www.mathworks.com/moler>.

[26]. M. Gerritsen " A brief introduction to MATLAB",p 2 , septembre 2006

[27]. G.J. Pottie, W.J. Kaiser, "Wireless integrated network sensors", Communications of the ACM 43 (5), pp. 551– 558. (2000)

[28]. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks, IEEE Communications Magazine, Vol. 40, No. 8, pp. 102-114, (2002).

[29]. E Sandeep Kumar. Fire-LEACH: A Novel Clustering Protocol for Wireless Sensor Networks based on Firey Algorithm. International Journal of Computer Science Theory and Application, ORB Academic Publisher, 2014, 1 (1), pp.15 - 17.