

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Tébessa  
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie  
Département : Math et informatique



## MEMOIRE DE MASTER

Spécialité : Informatique  
Option : Réseaux et Sécurité Informatique

### Thème

*Un système basé sur l'intelligence collective  
(l'intelligence computationnelle) pour le  
traitement de problème de clustering dans les  
réseaux de capteurs sans fil*

Présenté par:

*Salhi Ghania*

Devant le jury :

<i>Pr. LAOUAR Mouhammed Ridha</i>	<i>Pr</i>	<i>Université Larbi Tébessi</i>	<i>Président</i>
<i>Dr. SAHRAOUI Abdellatif</i>	<i>MCB</i>	<i>Université Larbi Tébessi</i>	<i>Examineur</i>
<i>Pr. DERDOUR Makhlouf</i>	<i>Pr</i>	<i>Université Larbi Tébessi</i>	<i>Encadreur</i>

2019/2020

## بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

### À MES CHERS PARENTS

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitterai jamais assez.*

*Mon père **ABD ELHAFID** demandé à dieu de bénir et nous rassemble au paradis. Ma mère **NAOUA**. Que Dieu, le Très Haut, lui accorder santé, bonheur et longue vie et faire en sorte que jamais je ne tu déçois.*

*À MES CHERES ET ADORABLES SOEURS QUE J'AIME PROFONDEMENT  
**RIMA, HALIMA, FADDA.***

*À MA CHERE NIECE **GHOUSSOUNE***

*À mes cousins et cousines et ses enfants et tous ceux que j'aime*

*À MES AMIS DE TOUJOURS*

*En souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble. Veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.*

# Remerciements

---

---

*C'est avec un grand plaisir que je réserve ces quelques lignes en signe de gratitude et de profonde reconnaissance à tous ceux qui de près ou de loin, ont contribué à la réalisation et l'aboutissement de ce travail*

*Tout d'abord, je tiens à remercier **Allah** tout puissant, de m'a permis de mener à bien ce mémoire, et de m'a orienté au chemin du savoir.*

*Ensuite, je remercie sincèrement Pr. Mr. **DERDOUR Makhlouf**, Professeur au sein du département mathématiques et informatique de la faculté des sciences exactes et sciences de la nature et de la vie à l'université de Tébessa, pour son encadrement, son assistance, son soutien, sa disponibilité et ses précieux conseils.*

*Mes vifs remerciements s'adressent à Pr. Mr. **LAOUAR Mouhammed Ridha**, Professeur au sein du département mathématiques et informatique de la faculté des sciences exactes et sciences de la nature et de la vie à l'université de Tébessa, et Dr. **SAHRAOUI Abdellatif**, Maitre de conférences au sein du département mathématiques et informatique de la faculté des sciences exactes et sciences de la nature et de la vie à l'université de Tébessa, pour l'honneur qu'il nous a accordé en acceptant de juger notre travail.*

*Enfin, j'adresse mes chaleureux remerciements à mes enseignants pour la qualité de l'enseignement qu'ils ont bien voulu me prodiguer durant mes études afin de me fournir une formation efficiente.*

*Merci à toutes et tous.*

# Résumé

---

---

Les réseaux de capteurs sans fil (RCSFs) représentent une technologie émergente qui vise à offrir des solutions et des capacités innovantes. Leur utilisation continue d'être augmentée dans de nombreux domaines. Cependant, les ressources limitées des nœuds capteurs sont une contrainte importante, principalement en termes d'efficacité énergétique. Pour cette raison, l'un des principaux problèmes dans les RCSFs est de développer un protocole de routage économe en énergie pour augmenter la durée de vie du réseau. L'utilisation de la technique de Clustering est un moyen d'atteindre l'efficacité énergétique. Dans ce mémoire nous proposons une approche basée sur l'intelligence computationnelle pour résoudre le problème de clustering de nœuds de capteurs dans un RCSF dans le but ultime de réduire les dépenses énergétiques pour prolonger la durée de vie du réseau. Dans ce contexte, un protocole de routage de données au sein d'un RCSF a été développé en utilisant l'algorithme génétique. L'analyse des performances et les résultats obtenus montrent que notre protocole GA-LEACHPEGASIS (*Genetic Algorithm LEACH PEGASIS*) proposé optimise de manière efficace la durée de vie du réseau en minimisant la consommation d'énergie par rapport au célèbre protocole LEACH.

**Mots clés :** Réseau de capteurs sans fil, intelligence computationnelle, clustering, efficacité énergétique, durée de vie du réseau.

# Abstract

---

---

Wireless sensor networks (WSNs) represent an emerging technology which aims to offer innovative solutions and capabilities. Their use is continuing to be increased in many areas. However, the limited resources of the sensor nodes is an important constraint, mainly in terms of energy efficiency. For this reason, one of the main issues in the WSNs is developing an energy-efficient routing protocol to improve the network lifetime. One way to achieve energy efficiency would be through the use of a clustering technique. In this work we propose an approach based on computational intelligence to deal with the problem of sensor nodes clustering in a WSN with the ultimate goal of reducing energy costs to extend the network lifetime. In this context, a data routing protocol within a WSN has been developed using one of the computational techniques. The performance analysis and the obtained results show that our proposed protocol GA-LEACHPEGASIS (*Genetic Algorithm LEACH PEGASIS*) efficiently optimizes the network lifetime by minimizing energy consumption compared to the well-known LEACH protocol.

**Keywords:** Wireless sensor network, computational intelligence, clustering, energy efficiency, network lifetime.

# ملخص

تمثل شبكات الاستشعار اللاسلكية (RCSFs) تقنية ناشئة تهدف إلى توفير حلول وقدرات مبتكرة. يستمر استخدامها في زيادة في العديد من المجالات. و مع ذلك فإن الموارد المحدودة لعقد أجهزة الاستشعار تشكل قيدًا بالغ الأهمية، بشكل رئيسي من حيث كفاءة الطاقة. لهذا السبب، أحد المشاكل الرئيسية في شبكات الاستشعار اللاسلكية هي تطوير بروتوكول توجيه فعال مقتصد للطاقة لزيادة عمر الشبكة. يعد استخدام تقنية التجميع إحدى الطرق لتحقيق كفاءة الطاقة. في هذه المذكرة، نقترح نهجًا يعتمد على الذكاء الحسابي لحل مشكلة تجميع عقد الاستشعار في شبكات الاستشعار اللاسلكية بهدف تقليل نفقات الطاقة لإطالة عمر الشبكة. في هذا السياق، تم تطوير بروتوكول توجيه البيانات داخل شبكة الاستشعار اللاسلكية باستخدام إحدى التقنيات الحسابية. يُظهر تحليل الأداء والنتائج التي تم الحصول عليها أن البروتوكول المقترح *GA-LEACHPEGASIS (Genetic Algorithm LEACH PEGASIS)* يعمل على تحسين عمر الشبكة بطريقة فعالة من خلال تقليل استهلاك الطاقة مقارنة ببروتوكول التسلسل الهرمي للتجميع العنقودي التكيفي منخفض الطاقة (LEACH).

**الكلمات المفتاحية:** شبكة الاستشعار اللاسلكية، الذكاء الحسابي، التكتل، كفاءة الطاقة، عمر الشبكة.

# Table des matières

---

---

<b>Remerciements</b>	i
<b>Résumé</b>	ii
<b>Table des matières</b>	v
<b>Liste des figures</b>	ix
<b>Liste des tableaux &amp; liste des algorithmes</b>	xi
<b>Liste des équations &amp; formules</b>	xii
<b>Introduction générale</b>	1
<b>Chapitre 1. Généralités sur les réseaux de capteurs sans fil</b>	3
Introduction	3
1. Présentation des réseaux de capteurs	3
1.1. Définition d'un capteur	3
1.2. Architecture d'un capteur	4
1.2.1. L'unité d'acquisition	4
1.2.2. L'unité de traitement	4
1.2.3. L'unité de transmission ou de communication	4
1.2.4. L'unité d'alimentation ( <i>Batterie</i> )	4
1.3. Définition d'un réseau de capteurs sans fil	5
2. Architecture de communication d'un réseau de capteurs	5
3. Pile protocolaire	6
3.1. Les couches de la pile	7
3.2. Les plans de la pile	8
4. Domaines d'applications des réseaux de capteurs	8
5. Caractéristiques et Limites des réseaux de capteurs	9
5.1. Caractéristiques	9
5.2. Limites	10
6. Consommation d'énergie dans les RCSFs	10
6.1. L'énergie d'un capteur	10
6.2. Modèle de consommation d'énergie	11
6.3. Facteurs de dissipation d'énergie	12

---

6.3.1. Etat du module radio	12
6.3.2. Accès au medium de transmission	13
6.4. Durée de vie d'un réseau de capteurs sans fil	14
7. Techniques de minimisation de la consommation d'énergie	15
7.1. Techniques au niveau de capture	15
7.2. Techniques au niveau de calcul	15
7.3. Techniques au niveau de communication	16
7.3.1. Classification selon la structure de réseau	17
7.3.1.1. Les protocoles à topologie plate	18
7.3.1.2. Les protocoles géographiques	20
7.3.1.3. Les protocoles hiérarchiques	23
7.3.1.4. Les protocoles considérant les qualités de services	24
Conclusion	24
<b>Chapitre 2. Etat de l'art : Le clustering &amp; les techniques computationnelles</b>	<b>25</b>
Introduction	25
1. Vue générale du problème de clustering	26
1.1. Définition d'une partition	26
1.2. Rappels sur la classification	26
1.2.1. La classification supervisée ( <i>ang. classification</i> )	26
1.2.2. La classification non supervisée ( <i>ang. clustering</i> )	27
2. Le clustering et les réseaux de capteurs sans fil	28
3. Les étapes d'un clustering	29
3.1. Sélection/ extraction des caractéristiques	29
3.2. Algorithme de classification automatique	29
3.3. Validation des résultats	30
3.4. Interprétation des résultats	30
4. Application du clustering	30
4.1. Rapprochement des clusters	32
5. Quelques approches classiques du clustering	33
5.1. Clustering hiérarchique	33
5.1.1. Protocole de routage hiérarchique LEACH	35
5.1.1.1. Architecture de communication de LEACH	35
5.1.1.2. Principe de fonctionnement de LEACH	36
5.1.1.3. Avantages et inconvénients de LEACH	38
5.1.2. Protocole de routage PEGASIS	39
5.1.2.1. La construction des chaînes	40
5.1.2.2. Déroulement de l'algorithme	41
5.1.2.3. Avantages et inconvénients du protocole PEGASIS	42
5.2. Le clustering par partitionnement	43

5.2.1. Clustering K-means	43
5.3. Le clustering spectral	43
5.4. Le clustering basé sur la densité	44
5.5. Le clustering basé sur les grilles	44
5.6. Le clustering basé sur les graphes	44
6. Le choix de l'algorithme de clustering	44
7. Vue générale sur l'intelligence computationnelle	45
7.1. Historique	46
7.2. Domaines d'application de l'intelligence computationnelle	47
8. Caractérisation des approches de l'intelligence computationnelle	47
8.1. La logique floue	47
8.2. Réseaux de neurones artificiels	48
8.3. Les algorithmes évolutionnaires	48
8.4. Les systèmes basés sur l'intelligence collective ( <i>Swarm Intelligence</i> )	49
9. Implémentation et les synergies de l'intelligence computationnelle	49
10. Application de l'intelligence computationnelle pour l'optimisation d'énergie dans les réseaux de capteurs sans fil	50
11. Travaux connexes	51
11.1. Travaux au niveau de routage	52
11.1.1. HEEP ( <i>Hybrid Energy Efficiency Protocol</i> )	52
11.1.2. CCBRP ( <i>Chain-Chain Based Routing Protocol</i> )	54
11.1.2.1. Fonctionnement de CCBRP	55
11.1.3. Comparaison des différents protocoles de routage de clustering dans les RCSFs	58
11.2. Travaux au niveau de construction des clusters	58
11.2.1. Approche basée sur les colonies d'Abeilles Artificielles	58
11.2.2. Approche basée sur les algorithmes génétiques	62
Conclusion	63
<b>Chapitre 3. Contribution : l'approche proposée pour le traitement du problème de clustering dans les RCSFs</b>	<b>64</b>
Introduction	64
1. Concepts de base de notre approche	64
1.1. Algorithme génétique	65
1.1.1. Source d'inspiration	65
1.1.2. Présentation d'un algorithme génétique	65
1.1.2.1. Chromosome	65
1.1.2.2. Opérations de l'algorithme génétique	66
1.1.2.3. Fonction de fitness	66
1.1.2.4. Résiliation de l'algorithme génétique	67
1.2. L'imbrication des clusters	67

2. Le principe de l'approche proposée	67
2.1. La description et le déroulement du protocole de routage proposé	69
2.1.1. Formation des clusters et l'élection des CHs par une technique computationnelle	69
2.1.2. Transmission des données	70
2.1.2.1. Construction des chemins	70
2.1.2.1.1. Construction des chaînes	71
2.1.2.2. L'envoi de données	74
Conclusion	75
<b>Chapitre 4. Simulation et analyse des performances</b>	<b>76</b>
Introduction	76
1. A propos de l'environnement MATLAB	76
1.1. Principales fonctionnalités du MATLAB	77
1.2. Choix de l'environnement MATLAB	78
2. Les étapes d'exécution du protocole	78
2.1. Déterminer les paramètres de simulation	78
2.2. Lancer la simulation	78
3. Paramètres de simulation	79
4. Mesure de performance	79
5. Simulation et résultats	80
5.1. L'énergie résiduelle	81
5.2. La durée de vie	82
5.3. Le nombre des CHs	83
5.4. La transmission des données au sink	84
6. Discussion	84
<b>Conclusion</b>	<b>85</b>
<b>Conclusion Générale</b>	<b>86</b>
<b>Références bibliographiques</b>	<b>87</b>

# Liste des figures

---

<b>Figure 1.1.</b> Exemple de capteur intelligent MICA2 de Crossbow	4
<b>Figure 1.2.</b> Composants d'un capteur	5
<b>Figure 1.3.</b> Fonctionnement des réseaux de capteur	6
<b>Figure 1.4.</b> Pile protocolaire	7
<b>Figure 1.5.</b> Applications des réseaux de capteurs	9
<b>Figure 1.6.</b> Energies consommées par un capteur	11
<b>Figure 1.7.</b> Modèle de consommation d'énergie	12
<b>Figure 1.8.</b> Consommation d'énergie en acquisition, traitement et communication	13
<b>Figure 1.9.</b> Emission simultanée d'un message (collision)	13
<b>Figure 1.10.</b> Surécoute dans une transmission	14
<b>Figure 1.11.</b> Techniques de conservation d'énergie	16
<b>Figure 1.12.</b> Classification des protocoles de routage selon la structure de réseau	17
<b>Figure 1.13.</b> Le protocole SPIN	19
<b>Figure 1.14.</b> Fonctionnement du protocole DD	20
<b>Figure 1.15.</b> Transitions des états dans GAF	22
<b>Figure 2.1.</b> Réseau de capteurs sans fils en Cluster	29
<b>Figure 2.2.</b> Les étapes d'un processus de clustering	30
<b>Figure 2.3.</b> Distance entre les deux clusters par lien simple	32
<b>Figure 2.4.</b> Distance entre les deux clusters par lien complet	32
<b>Figure 2.5.</b> Distance entre deux clusters par lien moyen	33
<b>Figure 2.6.</b> Distance entre deux clusters par lien moyen de groupe	33
<b>Figure 2.7.</b> Exemple de taxonomie	34
<b>Figure 2.8.</b> Exemple de dendrogramme	35
<b>Figure 2.9.</b> Architecture de communication du protocole LEACH	36
<b>Figure 2.10.</b> Construction de chaînes en utilisant l'algorithme avide	41
<b>Figure 2.11.</b> Approche de déplacement de jeton	42
<b>Figure 2.12.</b> Clusters de tailles et de densités différentes	45
<b>Figure 2.13.</b> Clusters de formes variées et concentriques	45
<b>Figure 2.14.</b> Position de l'intelligence computationnelle	46

---

<b>Figure 2.15.</b> 9-bit Représentation binaire des chromosomes	50
<b>Figure 2.16.</b> L'organisation des noeuds avec l'approche chains clustering	53
<b>Figure 2.17.</b> 100 noeuds capteurs divisés en 10 chaînes	55
<b>Figure 2.18.</b> La transmission des données pour CCBRP	57
<b>Figure 2.19.</b> Les phases de l'algorithme ABC	59
<b>Figure 3.1.</b> Un chromosome d'un RCSF	66
<b>Figure 3.2.</b> L'organisation des clusters d'un RCSF selon l'approche proposée	68
<b>Figure 3.3.</b> Schéma de construction des chemins	71
<b>Figure 3.4.</b> Les étapes d'exécution avec l'approche dynamique	72
<b>Figure 3.5.</b> Les étapes d'exécution avec l'approche statique	73
<b>Figure 3.6.</b> Exemple d'un tableau des chaînes indexées	73
<b>Figure 3.7.</b> Schéma de transmission de données	74
<b>Figure 4.1.</b> Les paramètres de simulation	78
<b>Figure 4.2.</b> Durée de vie du réseau avec LEACH classique	80
<b>Figure 4.3.</b> Durée de vie du réseau avec LEACH développé	80
<b>Figure 4.4.</b> Le déploiement aléatoire des nœuds capteurs	81
<b>Figure 4.5.</b> La consommation d'énergie versus les ronds (temps d'exécution)	81
<b>Figure 4.6.</b> Nombre de nœuds vivants	82
<b>Figure 4.7.</b> Nombre de nœuds morts	82
<b>Figure 4.8.</b> Les ronds relatifs au 1 <sup>er</sup> nœud mort, 10 <sup>ème</sup> nœud mort et dernier nœud mort	83
<b>Figure 4.9.</b> Nombre de CHs par unité de temps	84
<b>Figure 4.10.</b> Nombre de paquets reçus par la station de base	84

# Liste des tableaux & liste des algorithmes

---

---

## Liste des tableaux

<b>Tableau 2.1.</b> Exemple d'application de l'apprentissage supervisé	27
<b>Tableau 2.2.</b> Exemple d'application d'apprentissage non supervisé	28
<b>Tableau 2.3.</b> Comparaison des différents protocoles de routages de clustering dans les RCSFs	58
<b>Tableau 4.1.</b> Les paramètres de simulation	79

## Liste des algorithmes

<b>Algorithme 2.1.</b> Pseudo-code pour la 1ere phase de CCBRP	56
<b>Algorithme 2.2.</b> Pseudo-code pour la 2eme phase de CCBRP	56
<b>Algorithme 2.3.</b> Algorithme basé sur ABC pour le Clustering d'un RCSF	61
<b>Algorithme 3.1.</b> Les étapes de l'approche proposée pour la formation des clusters et l'élection des CHs	69
<b>Algorithme 3.2.</b> Algorithme de construction de chaînes	72

# Liste des équations

---

---

<b>Equation 1.1.</b> L'énergie consommée par un capteur _____	11
<b>Equation 1.2.</b> L'énergie consommée par l'unité de communication _____	11
<b>Equation 1.3.</b> L'énergie de transmission _____	11
<b>Equation 1.4.</b> L'énergie de réception _____	11
<b>Equation 2.1.</b> Le nombre de CHs dans un RCSF avec le protocole LEACH _____	37
<b>Equation 2.2.</b> La probabilité de chaque nœud pour devenir CH _____	37
<b>Equation 2.3.</b> Autre méthode pour calculer la probabilité de chaque nœud pour devenir CH _____	37
<b>Equation 2.4.</b> La fonction de fitness de l'approche basée sur les AGs _____	50
<b>Equation 2.5.</b> Le coût énergétique de la phase d'initialisation _____	54
<b>Equation 2.6.</b> L'énergie totale des réseaux _____	54
<b>Equation 2.7.</b> Le total approximatif des tours _____	54
<b>Equation 2.8.</b> L'énergie totale dissipée dans le réseau pendant une rond _____	54
<b>Equation 2.9.</b> La solution de chaque abeille artificielle _____	59
<b>Equation 2.10.</b> La fonction de fitness de l'algorithme ABC _____	60
<b>Equation 2.11.</b> Le pourcentage correspondant à chaque valeur de fitness _____	61
<b>Equation 2.12.</b> La fonction de fitness de l'approche basée sur les AGs _____	62
<b>Equation 3.1.</b> Le coût énergétique de la phase de clustering _____	70

# Introduction Générale

*" La réussite est la place qu'on occupe dans les journaux.*

*La réussite est l'insolence d'un jour. "*

***Dave BARRY***

---

## **1. Contexte et problématique**

Les réseaux de capteurs sans fil se composent d'un ensemble de capteurs capables de communiquer entre eux et de fournir des informations par la combinaison des mesures prises par les différents capteurs. Ces RCSFs sont utilisés dans plusieurs domaines tels que : le domaine militaire, médical, industriel, transport, et ils sont utilisés aussi dans les applications de surveillance de l'environnement (monitoring)...etc. Bien que ces réseaux soient utiles et intéressants à mettre en œuvre, ils admettent plusieurs inconvénients qui représentent des défis pour les chercheurs dans ce domaine. De par la nature-même des RCSF, beaucoup de défis imposés par leur structures et leur déploiement sont à relever en revanche. Les problèmes sont causés principalement par les échecs de communication, les contraintes de stockage et de calcul, les ressources énergétiques limitées et les déploiements et localisations sous optimaux.

Parmi les défis essentiels associés à ces réseaux nous pouvons citer: la consommation d'énergie et la durée de vie puisque les capteurs sont généralement dotés de batteries non rechargeables et leur remplacement est quasiment impossible quand ils sont déployés dans des zones hostiles. A cet effet, dans le cadre de ce projet de fin d'études, nous proposons une nouvelle approche de routage qui permet de relever ces défis. Cette approche proposée est basée sur le clustering en utilisant l'un des algorithmes de l'intelligence computationnelle pour organiser le réseau et un protocole de routage pour acheminer les données prises par les capteurs à la station de base. Ceci est dans le but de minimiser la consommation d'énergie et par conséquent augmenter la durée de vie du réseau.

## **2. Objectif et motivations**

Notre travail de recherche consiste à investiguer le potentiel d'application des algorithmes d'intelligence computationnelle dans le domaine des réseaux de capteurs sans fil et à proposer une nouvelle approche apportant une contribution dans le routage à base de Clustering.

La contribution réalisée au cours de ce mémoire est dédiée à l'exploration du domaine des RCSFs. Dans ce travail, nous proposons une nouvelle approche basée sur l'intelligence computationnelle pour traiter le problème de clustering afin de proposer un meilleur protocole permettant une consommation d'énergie optimale et une prolongation de la durée de vie de réseau de capteurs.

## **3. Organisation du manuscrit**

Ce mémoire est constitué de quatre (4) chapitres :

1. Le premier chapitre est un chapitre introductif commence par l'introduction des concepts de base des réseaux de capteurs sans fil, suivi par les différents protocoles de routages et techniques d'économie d'énergie utilisées ces dernières années pour allonger significativement la durée de vie d'un réseau de capteurs sans fil.
2. Le Chapitre 2 présente un état de l'art sur le clustering et l'intelligence computationnelle, suivi par les travaux connexes appliqués dans les RCSFs pour traiter le problème de clustering.
3. Dans le Chapitre 3 nous présentons en détail notre contribution qui consiste à appliquer un algorithme d'intelligence computationnelle afin de concevoir un protocole de routage capable de traiter le problème de clustering et optimiser la consommation d'énergie dans les réseaux de capteurs sans fil.
4. Finalement, dans le dernier chapitre, nous présentons un aperçu sur les outils utilisés pour l'implémentation ainsi que les comparaisons et les résultats obtenus.

Nous concluons ce manuscrit par une conclusion générale concernant la contribution proposée ainsi que les perspectives envisagées en vue d'ouvrir de nouvelles directions.

# Chapitre 1

## Généralités sur les réseaux de capteurs sans fil

*"Toute réussite est relative. Vous pouvez améliorer votre réussite*

*relative en dénigrant les compétences et les résultats de votre entourage."*

**Scott ADAMS**

---

### Introduction

Les dernières décennies ont été marquées par le développement rapide de la technologie de transmission sans fil et tout particulièrement par l'apparition des réseaux de capteurs sans fil (RCSFs). Ces réseaux se composent de micros composants nommés « capteurs » peu coûteux, capables de collecter des informations pour les traiter ensuite les transmettre à l'aide d'une communication sans fil à une station de base. Par conséquent, les réseaux de capteurs ont révolutionné le paradigme de collecte et de traitement d'informations dans divers environnements, ils sont au centre de diverses applications dans plusieurs secteurs tel que la surveillance de zones sensibles pour les militaires, la détection de feu de forêt, le diagnostic médical...etc. [1].

Dans ce chapitre, nous allons présenter un ensemble de généralités sur les réseaux de capteurs sans fil, leurs architectures, leurs caractéristiques, leurs limites, ainsi que leurs domaines d'application. Nous discuterons la pile protocolaire, et nous terminerons par la consommation d'énergie comme un grand défi des réseaux de capteurs sans fil et les différentes techniques de conservation d'énergie dans ces réseaux, ainsi que les différentes approches de routage dans les RCSFs en articulant les avantages et les inconvénients de chacune.

### 1. Présentation des réseaux de capteurs

#### 1.1. Définition d'un capteur

Un capteur est un dispositif électronique qui transforme l'état d'une grandeur physique observée (température, lumière, pression, etc.) en une grandeur utilisable. Il constitue l'élément de base des systèmes d'acquisition de données [2]



**Figure 1.1.** Exemple de capteur intelligent MICA2 de Crossbow[2].

## 1.2. Architecture d'un capteur

L'infrastructure d'un capteur est concrétisée autour de quatre principales unités comme le montre la figure 1.1.

### 1.2.1. L'unité d'acquisition

Cette unité englobe deux sous-unités : le capteur et le convertisseur analogique-numérique (ADC). Le capteur obtient des mesures numériques sur des phénomènes observés et les transforme en signaux analogiques. L'ADC convertit ces signaux analogiques en signaux numériques puis il les transmet à l'unité de traitement [3].

### 1.2.2. L'unité de traitement

Cette unité constitue l'élément le plus important du capteur. Elle est composée d'un processeur et d'une unité de stockage. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition et stocke les données collectées [4].

L'unité de traitement possède deux interfaces : une interface avec l'unité d'acquisition et une autre avec l'unité de transmission. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de transmission [5].

### 1.2.3. L'unité de transmission ou de communication

L'unité de transmission fournit un moyen de communication entre les nœuds de réseau; elle assure l'échange de donnée (émissions et réceptions) entre ces nœuds via un support de communication radio [4].

### 1.2.4. L'unité d'alimentation (Batterie)

C'est l'unité qui alimente les trois unités citées précédemment, elle fournit l'énergie nécessaire à leur fonctionnement [2].

Il existe d'autres composants complémentaires qui peuvent être reliés aux capteurs comme les systèmes de localisation GPS (*Global Position System*) et les mobilisateurs qui permettent le déplacement des capteurs, etc. [5].

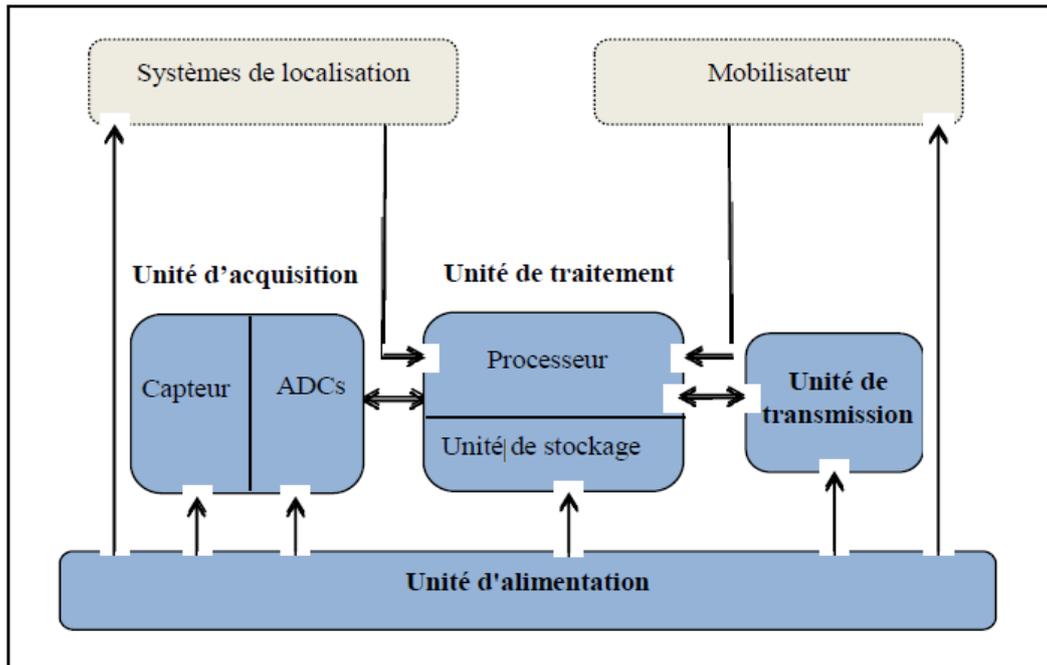


Figure 1.2. Composants d'un capteur [5].

### 1.3. Définition d'un réseau de capteurs sans fil

Un réseau de capteurs sans fil est un type spécial de réseaux ad hoc avec un grand nombre de nœuds qui sont des micro-capteurs capables de recevoir et de transmettre des données environnementales d'une manière autonome sans intervention humaine. La position de ces nœuds n'est pas obligatoirement prédéterminée, ils peuvent être aléatoirement dispersés dans une zone géographique appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté (par exemple : lâchée de capteurs sur un volcan pour étudier les phénomènes volcanologiques et leurs évolutions).

Le réseau possède en général un nœud particulier, la base (ou sink), connectée avec les autres nœuds par un réseau filaire et reliée à une alimentation électrique [6].

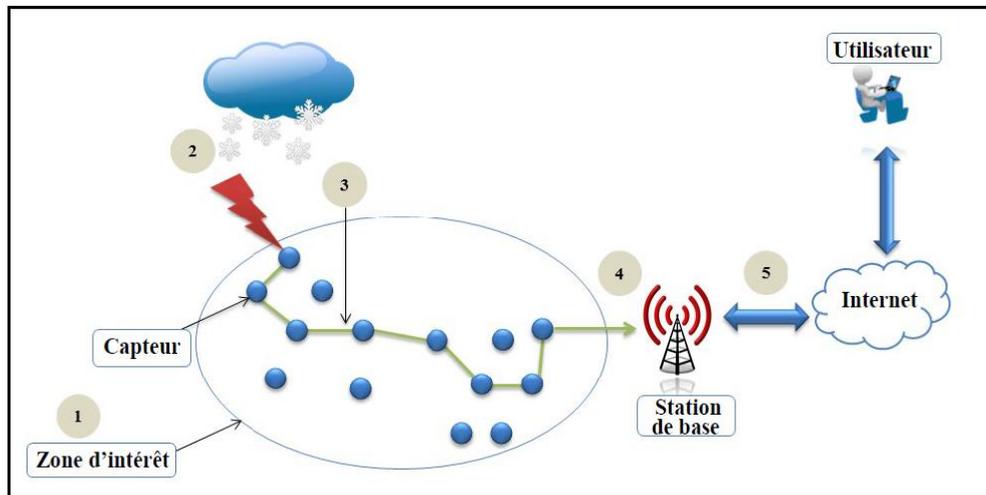
## 2. Architecture de communication d'un réseau de capteurs

L'architecture des réseaux de capteurs fournit un prototype nécessaire à la compréhension des réseaux de capteurs et des relations entre les éléments utilisés. Cette architecture se base sur l'interaction entre les trois éléments suivant [7] :

- **Nœud capteur**
- **Zone d'intérêt** : Appelée aussi « champ de captage », c'est une zone géographique où on place les capteurs du réseau pour superviser ou surveiller des phénomènes divers.
- **Stations de base (Sink)** : Nommé aussi nœuds-puits où l'écoulement des données se termine. Ces nœuds-puits sont des nœuds spéciaux qui possèdent plus de

ressources matérielles et permettent de collecter et stocker les informations sorties des capteurs.

L'interaction entre ces éléments suit une séquence d'opérations telles que le déploiement, la détection d'évènement et la transmission des données. Ces opérations doivent être exécutées dans l'ordre comme dans la figure 1.3. [8].

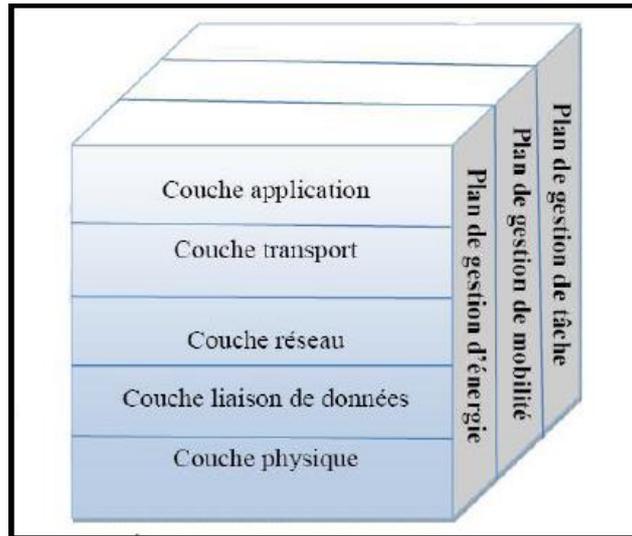


**Figure 1.3.** Fonctionnement des réseaux de capteur [8].

1. Les capteurs sont déployés dans la zone d'intérêt de façon aléatoire ou fixe.
2. Lorsqu'un capteur détecte un évènement il le traite localement pour acquérir des données concernant cet évènement.
3. La transmission des données acquis par le capteur est réalisée par le biais d'une communication multi-sauts où un chemin est créé pendant le passage des messages par plusieurs nœuds.
4. La transmission des données se déroule jusqu'à ce qu'elles atteignent la station de base où l'écoulement des données se termine.
5. La station de base est branchée à une machine puissante via internet pour utiliser les données récoltées.

### 3. Pile protocolaire

La pile protocolaire mise en œuvre pour l'ensemble de protocoles de communication du réseau de capteurs [9] est illustrée par la figure 1.4.



**Figure 1.4.** Pile protocolaire [9].

La pile protocolaire comprend 5 couches qui possèdent les mêmes fonctions que celles du modèle OSI (*Open Systems Interconnection*). Chaque couche de la pile 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. Elle comprend ainsi 3 plans pour la gestion d'énergie, la gestion de la mobilité ainsi que la gestion des tâches.

### 3.1. Les couches de la pile

La pile protocolaire pour les réseaux de capteurs est organisée en couches superposées l'une sur l'autre. Chaque couche assure une partie des fonctionnalités nécessaire à la communication entre les entités du réseau. Les cinq couches du RCSF sont [9] :

- **La couche application** : Elle constitue les différentes applications qui peuvent être implémentées sur un réseau. Il s'agit du niveau le plus proche des utilisateurs géré directement par le logiciel.
- **La couche transport** : Elle permet de diviser les données issues de la couche application en paquets pour les délivrer, ainsi elle réordonne et rassemble les paquets venus de la couche réseau avant de les envoyer à la couche application.
- **La couche réseau** : Elle permet de trouver un chemin et une transmission fiable des données fournies par la couche transport.
- **La couche liaison de données** : Elle prend soin de multiplexage du flux de données, contrôler l'accès aux MAC (Media Access Control), établir les liaisons point à point et multi-point dans le réseau et assurer la détection et la réception des trames de données.

- **La couche physique** : Elle permet d'assurer la génération et modulation du signal radio, Estimation de la qualité de liens (signal) et la sélection des fréquences porteuses des données au niveau bit.

### 3.2. Les plans de la pile

Les plans de gestion sont nécessaires afin que les nœuds capteurs puissent fonctionner ensemble de manière efficace et prolonger la durée de vie du réseau. Les trois plans du RCSF sont [9]:

- **Le plan de gestion de l'énergie** : Il permet de préserver l'énergie en contrôlant l'utilisation de la batterie. Par exemple, un capteur se met en veille après la réception d'un message à partir d'un voisin dans le but d'éviter la duplication des messages déjà reçus. Dans le cas où le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage.
- **Le plan de gestion de la mobilité** : Il assure le routage des données dans un réseau de capteurs mobile. Il est capable d'enregistrer les mouvements du nœud afin de l'aider à se localiser.
- **Le plan de gestion des tâches** : Comme il n'est pas nécessaire que tous les nœuds de réseau travaillent avec le même rythme, certains nœuds exécutent la tâche de captage plus que d'autres selon leur niveau d'énergie. Le plan de gestion des tâches permet d'ordonner les différentes tâches de captage de données dans une zone spécifique afin d'assurer un travail coopératif et efficace au niveau de consommation d'énergie.

## 4. Domaines d'applications des réseaux de capteurs

Les avantages qui caractérisent les réseaux de capteurs les permettent de s'adapter à plusieurs domaines d'applications. Parmi ces domaines où les réseaux de capteurs se révèlent très utiles et peuvent offrir de meilleures contributions, on peut noter le domaine militaire, environnemental, domestique, sanitaire et de la sécurité, etc. [2][6].

- **Applications militaires** : On peut penser à un réseau de capteurs déployé sur un endroit stratégique ou d'accès difficile, afin de surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations).
- **Applications domestiques** : En plaçant, sur le plafond ou dans le mur, des capteurs, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes.

- **Applications environnementales** : Les réseaux de capteurs sont beaucoup appliqués dans ce domaine pour détecter des incendies, surveiller des catastrophes naturelles, détecter des pollutions et suivre des écosystèmes.
- **Applications agricoles** : 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.
- **Applications médicales** : Les réseaux de capteurs ont aussi des développements dans le domaine de diagnostic médical. Par exemple, des micro-caméras sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures.
- **Applications de transport** : Il est possible d'intégrer des nœuds capteurs au 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.

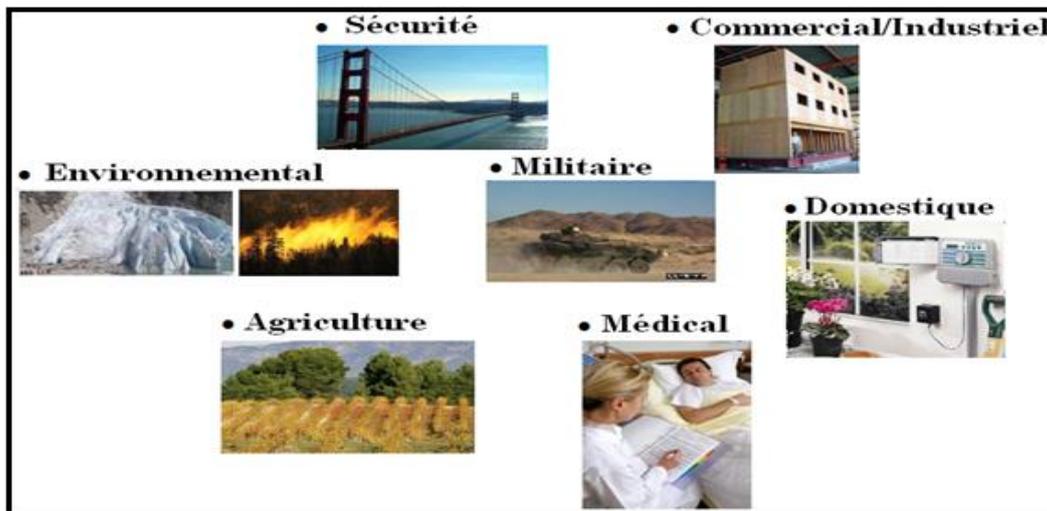


Figure 1.5. Applications des réseaux de capteurs [6].

## 5. Caractéristiques et Limites des réseaux de capteurs

Les caractéristiques et limites des réseaux de capteurs sans fil sont nombreux on peut citer les suivants [6]:

### 5.1. Caractéristiques

- **Absence d'infrastructure** : En général, les réseaux ad hoc, se distinguent des autres réseaux mobiles par la propriété d'absence d'infrastructure préexistante et de tout genre d'administration centralisée. Les hôtes mobiles sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.
- **Taille** : Un grand nombre de nœuds dispersés aléatoirement (des réseaux de 10000 nœuds peuvent être envisagés).

- **Contrainte d'énergie** : Les nœuds capteurs sont placés dans des surfaces distantes dans plusieurs applications, le service du nœud peut ne pas être possible, dans ce cas la durée de vie du nœud peut être déterminée par la durée de vie de la batterie, ce qui exige la minimisation des dépenses d'énergies.
- **Topologie dynamique** : Les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi, la topologie du réseau fréquemment changeante.
- **Auto organisation du réseau** : Ceci peut être nécessaire dans plusieurs cas. Par exemple, un réseau comportant un grand nombre de nœuds placés dans des endroits hostiles où la configuration manuelle n'est pas faisable, doit être capable de s'auto-organiser. Un autre cas est celui où un nœud est inséré ou retiré (à cause d'un manque d'énergie ou de destruction physique), ainsi le réseau doit être capable de se reconfigurer pour continuer sa fonction.
- **Sécurité physique limitée** : Les RCSFs mobiles sont plus touchés par les paramètres de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.

## 5.2. Limites

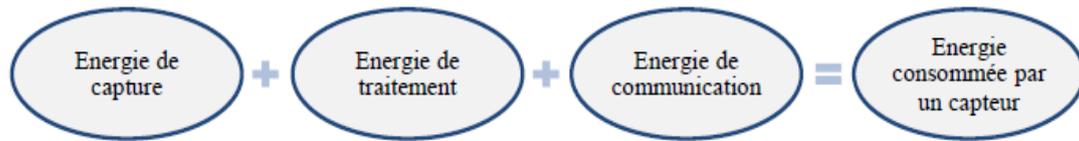
- Les ressources de calcul et de mémoire des nœuds sont relativement faibles, par exemple les nœuds de capteur de type "mote" sont composés d'un microcontrôleur 8-bits 4MHz, 40 KOctets de mémoire et une radio avec un débit d'environ 10 kbps.
- Non seulement les capacités des nœuds sont faibles, mais en plus ils opèrent sur des piles et par conséquent ont une durée de vie limitée.
- L'énergie limitée des capteurs est probablement la caractéristique la plus pénalisante, le plus grand des défis dans le domaine des réseaux de capteurs reste de concevoir des protocoles, entre autre de sécurité, qui minimisent l'énergie afin de maximiser la durée de vie du réseau. En d'autres mots, l'énergie est sans aucun doute la ressource qui convient pour gérer avec la plus grande attention.

## 6. Consommation d'énergie dans les RCSFs

Pour maximiser la durée de vie du réseau, l'efficacité énergétique est une forte exigence. Donc, prolonger la durée de vie de chaque nœud capteur devient un objectif en soi, afin de garantir une meilleure durée de vie globale au réseau.

### 6.1. L'énergie d'un capteur

L'énergie consommée par un nœud capteur est due essentiellement aux opérations suivantes : la capture, le traitement et la communication de données [10].



**Figure 1.6.** Energies consommées par un capteur [10].

- **Energie de capture :** En général, l'énergie de capture représente un faible pourcentage de l'énergie totale consommé par un capteur. Il est dissipée pour accomplir certain nombre des tâches tel que, l'échantillonnage, traitement de signal, conversion analogique/numérique et activation de la sonde du capture.
- **Energie de traitement :** L'énergie de traitement est un peu plus que l'énergie de capture. Cependant, elle est faible par rapport à celle nécessaire pour la communication. Elle se divise en deux parties :
  - 1) L'énergie de commutation qui est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel).
  - 2) L'énergie de fuite qui correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement.
- **Energie de communication :** Représente la portion la plus grande de l'énergie consommée par un nœud capteur. L'énergie de communication se décline en deux parties : l'énergie de réception et l'énergie de l'émission. Cette énergie est influencée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio.

## 6.2. Modèle de consommation d'énergie

Formellement, l'énergie ' $E_c$ ' consommé par un capteur est défini dans [11] par l'équation (1):

$$E_c = E_{c/capture} + E_{c/traitement} + E_{c/communication} \quad (1.1)$$

Dans lequel:

- $E_{c/capture}$ : l'énergie consommée par l'unité de capture.
- $E_{c/traitement}$ : l'énergie consommée par l'unité de traitement.
- $E_{c/communication}$ : l'énergie consommée par l'unité de communication.

$E_{c/communication}$  représente la somme des deux valeurs: ' $E_{TX}$ ' qui est l'énergie de transmission et ' $E_{RX}$ ' qui est l'énergie de réception, comme le montre l'équation (2):

$$E_{c/communication} = E_{TX} + E_{RX} \quad (1.2)$$

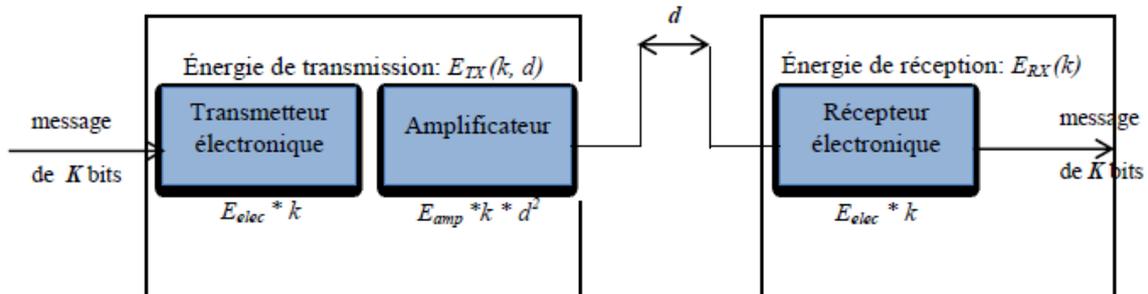
Où:

$$E_{TX}(k, d) = (E_{elec} * k) + (E_{amp} * k * d^2) \quad (1.3)$$

$$E_{RX}(k) = E_{elec} * k \quad (1.4)$$

Et:

- $k$ : la taille des données en bits.
- $d$ : distance entre le nœud émetteur et le nœud récepteur.
- $E_{elec}$ : l'énergie de transmission électronique.
- $E_{amp}$ : l'énergie d'amplification.



**Figure 1.7.** Modèle de consommation d'énergie [11].

À travers ce modèle de consommation d'énergie, nous notons que la distance géographique ' $d$ ' entre l'émetteur et le récepteur et la taille des données fournies ' $k$ ' sont des facteurs dominants dans la consommation d'énergie.

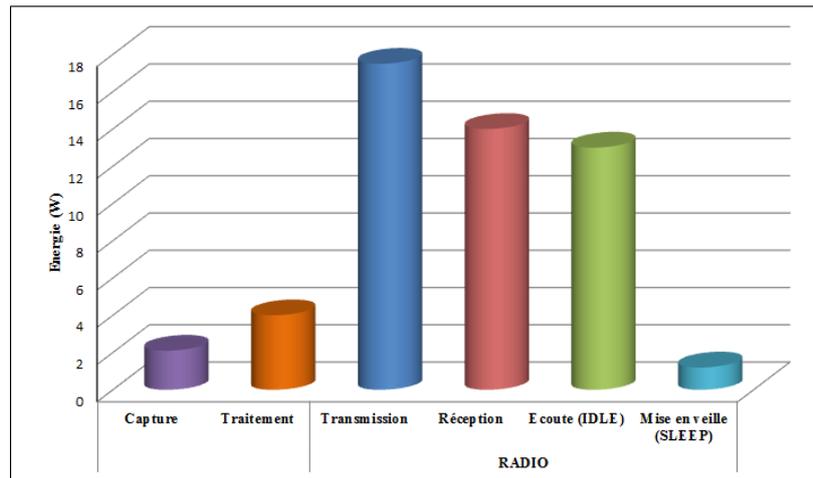
### 6.3. Facteurs de dissipation d'énergie

Pour assurer l'efficacité énergétique les solutions envisagées doivent réduire ou éliminer les facteurs de dissipation d'énergie qui sont présentées ci-dessous.

#### 6.3.1. Etat du module radio

Le module radio est le composant du nœud capteur qui consomme un pourcentage élevé d'énergie que les autres composants, puisque c'est lui qui assure la communication entre les capteurs. Ce module passe principalement par les quatre états suivants [12]:

- Etat transmission : La radio transmet un paquet.
- Etat réception : La radio reçoit un paquet.
- Etat sommeil (mise en veille, SLEEP): La radio est mise hors tension.
- Etat actif (écoute, IDLE): Dans cet état la radio est allumée, mais elle n'est pas employée (ni entrain de recevoir, ni de transmettre). Cela provoque une perte de l'énergie suite à l'écoute inutile du canal de transmission. Pour éviter ce problème, il faut basculer les nœuds dans le mode sommeil le plus longtemps possible.



**Figure 1.8.** Consommation d'énergie en acquisition, traitement et communication [13].

### 6.3.2. Accès au médium de transmission

La sous couche MAC assure l'accès au support de transmission, la fiabilité de transmission, le contrôle de flux, la détection d'erreur et la retransmission des paquets. Parce que les nœuds partagent le même médium de transmission, la sous-couche MAC joue un rôle important pour la coordination entre les nœuds et la minimisation de la consommation d'énergie [14]. Dans ce qui suit, nous citons les principales causes de consommation d'énergie au niveau de la couche MAC.

- **La collision**

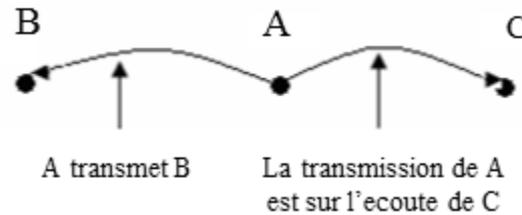
La collision est l'une des sources de perte d'énergie les plus importantes. Les nœuds capteurs ont en général une seule antenne radio et partagent le même canal de transmission. Par conséquent, dans le cas de transmission simultanée des données provenant de plusieurs capteurs une collision se produit. Ces données transmises deviennent inexploitables et doivent être abandonnées [15].



**Figure 1.9.** Emission simultanée d'un message (collision) [15].

- **La surécoute**

Quand un nœud reçoit des paquets qui ne lui sont pas destinés, le phénomène de surécoute (*overhearing*) se produit. La surécoute conduit à une perte d'énergie additionnelle à cause de l'implication des autres capteurs dans la réception des données [15].



**Figure 1.10.** Surécoute dans une transmission [15].

- **La surcharge (overhead)**

On a toujours besoin de l'échange de paquets de contrôle pour assurer le fonctionnement de protocoles de la couche MAC,. Ces paquets assurent les différentes fonctionnalités : signalisation, connectivité, établissement de plan d'accès et évitement de collisions. On parle alors d'overhead des paquets de contrôle qui nécessite une énergie additionnelle [11].

- **La surémission (overemitting)**

Quand un nœud capteur envoie les données à un destinataire qui n'est pas prêt à les recevoir, le phénomène de surémission (*overemitting*) se produit. En effet, les messages envoyés sont considérés inutiles et consomment une énergie additionnelle [15].

- **L'écoute à vide (idle listening)**

Quand la radio écoute le canal pour recevoir d'éventuelles données, l'écoute à vide ou (*idle listening*) se produit. Son coût énergétique est particulièrement élevé et le coût exact de l'écoute à vide dépend du matériel radio et du mode de fonctionnement [15].

- **La taille des paquets**

La taille des messages échangés dans le réseau a un effet sur la consommation d'énergie des nœuds émetteurs et récepteurs. En effet l'augmentation de la taille des paquets produit plus d'énergie. Donc une grande puissance de transmission est nécessaire pour des paquets de grande taille [16].

#### 6.4. Durée de vie d'un réseau de capteurs sans fil

La durée de vie est une métrique d'évaluation de performance très importante dans les RCSFs. Il existe plusieurs définitions de ce paramètre de mesure. Toutefois, notons que sa définition n'est pas toujours triviale.

Dans [17] Chen et Zhao, conçoivent la durée de vie d'un réseau de capteurs sans fil comme étant une mesure étroitement liée à la vie des nœuds du réseau. Quant à Esseghir et al. [18], ils définissent la durée de vie comme étant une autre mesure qui est liée à l'accessibilité des nœuds dans le réseau...etc.

Dans notre cas d'étude, On peut considérer la durée de vie d'un RCSF comme étant liée à l'activité nodale. Donc la durée de vie du RCSF est définie comme étant la durée pendant laquelle le réseau assure toutes les fonctionnalités du réseau (couverture, connectivité...etc.).

## 7. Techniques de minimisation de la consommation d'énergie

Après la description des principales causes de consommation d'énergie dans les RCSFs, nous présentons dans ce qui suit les différentes techniques utilisées pour minimiser cette consommation. Ces techniques sont appliquées soit au niveau de capture, au niveau de calcul ou au niveau de communication.

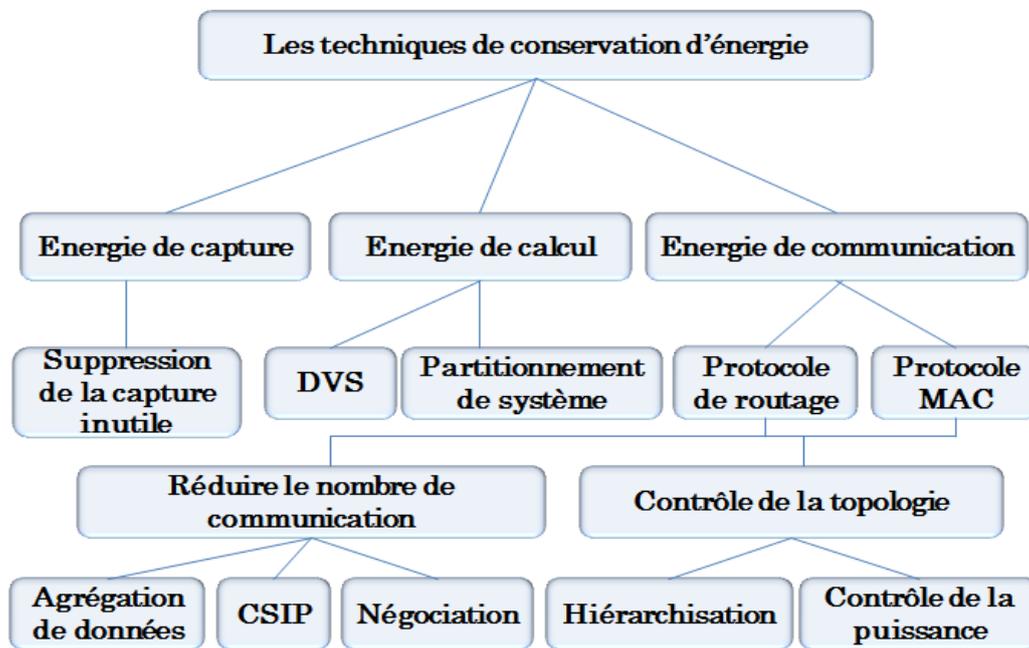


Figure 1.11. Techniques de conservation d'énergie.

### 7.1. Techniques au niveau de capture

La seule solution apportée pour la minimisation de la consommation d'énergie au niveau de la capture consiste à réduire les durées de captures [19].

### 7.2. Technique au niveau de calcul

L'énergie de calcul peut être optimisée en utilisant deux techniques [19]:

- L'approche DVS (*Dynamique Voltage Scaling*), qui consiste à ajuster de façon adaptative la tension d'alimentation et la fréquence de microprocesseur pour économiser la puissance de calcul sans dégradation des performances.
- L'approche de partitionnement de système, qui consiste à transférer un calcul prohibitif en temps de calcul vers une station de base qui n'a pas de contraintes énergétiques et qui possède une grande capacité de calcul.

### 7.3. Technique au niveau de communication

La minimisation de la consommation d'énergie pendant la communication est étroitement liée aux protocoles développés pour la couche réseau et la couche MAC. Ces protocoles se basent sur plusieurs techniques : agrégation de données, négociation et CSIP (*Collaborative Signal and Information Processing*). Cette dernière technique est une discipline qui combine plusieurs domaines: la communication et le calcul à basse puissance, algorithmes distribués et tolérance aux fautes, traitement de signal, Systèmes adaptatifs et théorie de fusion des capteurs et des décisions. Ces techniques ont pour but de réduire le nombre d'émission/ réception des messages [19].

Par contre, le contrôle de la topologie permet l'ajustement de la puissance de transmission et le regroupement des nœuds capteurs (hiérarchisation).

- Le contrôle de la puissance de transmission n'a pas seulement un effet sur la durée de vie de la batterie d'un nœud capteur, mais aussi sur la capacité de charge du trafic qui est caractérisée par le nombre de paquets transmis avec succès vers une destination. En outre, il influe sur la connectivité et la gestion de la densité (le nombre de nœuds voisins). Ainsi, il peut conserver l'énergie à deux niveaux : explicitement par l'application de puissances faibles d'émissions et implicitement en réduisant la contention avec d'autres nœuds transmetteurs. Le module de contrôle de la puissance est souvent intégré dans les protocoles soit de la couche réseau soit de la couche MAC [19].
- La hiérarchisation consiste à organiser le réseau en structure à plusieurs niveaux. C'est le cas, par exemple, des algorithmes de groupement (*clustering*), qui organisent le réseau en groupes (*clusters*) avec des chefs de groupe (*cluster head*) et des nœuds membres [19].

Une panoplie de protocoles a été proposée dans la littérature. La grande partie de ces protocoles est destinée à la couche réseau et MAC.

Le principe de fonctionnement de chaque protocole de routage dans les RSCFs diffère suivant la philosophie de l'approche à laquelle il appartient. Ces approches peuvent être distinguées suivant : la topologie du réseau, les opérations supportées et la destination des paquets transmis [20].

La topologie du réseau de capteurs sans fil, subdivise les solutions du routage en trois catégories principales: les protocoles centrés sur les données d'architecture plate, les protocoles hiérarchiques, et les protocoles de routage géographique.

En plus, si les opérations supportées sont présent en charge, les solutions peuvent être classée en: basée multi-chemins (multipath-based), basée requêtes (query-based),

basée négociation (negociation-based), basée sur les qualités du service (Qos-based) et basée sur la cohérence du traitement des données (coherent-based) .

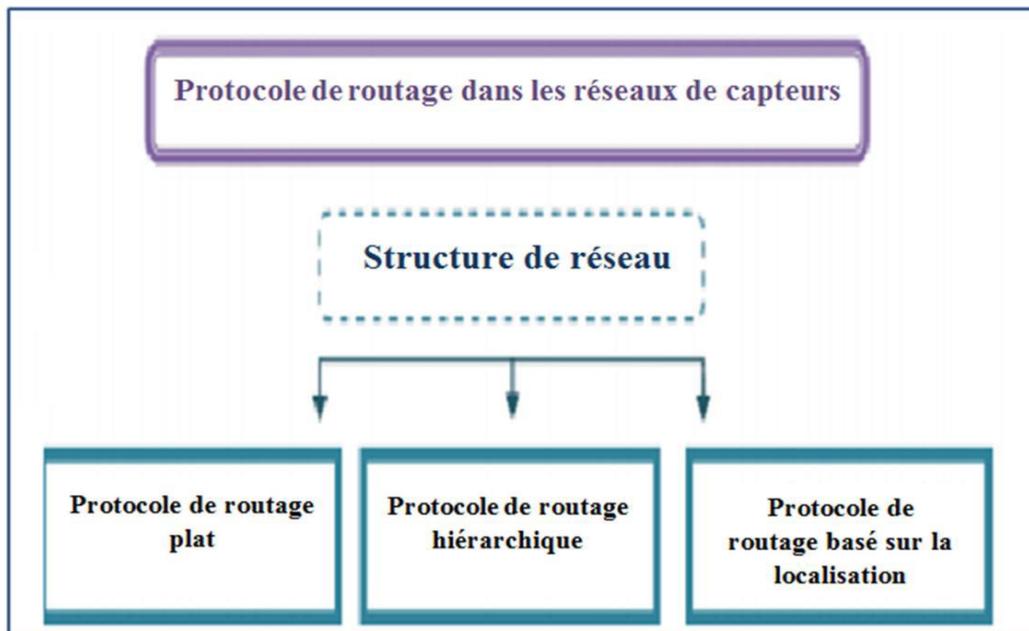
Un traitement non-cohérent de données signifie que chaque nœud agrège ses données avant la transmission, par contre, un traitement cohérent signifie que les données sont transmises vers des nœuds spécifique dits « Aggregator » pour les opérations d'agrégation [21].

Le dernier critère de classification, qui est la destination des paquets transmis, distingue les solutions proposées en [22]: communication « Unicast », où la destination du message est un et un seul nœud, communication « Multicast », où les destinataires du message sont un sous ensemble définie du réseau et la communication en « Broadcast » où le message est destiné à l'ensemble du réseau.

Parmi les classifications mentionnées ci-dessus, celle selon la topologie est la plus répondue en littérature (enrichie par une quatrième classe basée sur QoS qui prend en charge quelques qualités de communication en plus du routage de données) ; c'est cette classification que nous allons utiliser pour présenter les protocoles existants pour les RCSFs.

### 7.3.1. Classification selon la structure du réseau

Comme c'est illustré dans la figure suivante, les protocoles de routage basés sur la structure du réseau peuvent être classifiés en trois catégories : protocoles plats, protocoles hiérarchiques et protocoles basés sur la localisation.



**Figure 1.12.** Classification des protocoles de routage selon la structure de réseau [23].

### 7.3.1.1. Les protocoles à topologie plate

Dans une topologie plate, tous les nœuds possèdent le même rôle. Les nœuds sont semblables en termes de ressources.

Les protocoles à topologie plate, aussi dit, protocoles data-centric sont les premiers à être utilisés pour le routage sur les RCSFs; ils ont un principe simple, c'est la BS qui envoie des requêtes vers des zones spécifiques du réseau et attend l'arrivée des réponses des nœuds visés. Puisque les données sont demandées par les requêtes, un nommage basé sur les attributs est nécessaire pour permettre la correspondance entre requête envoyée par la BS et les données recensées au niveau des nœuds [24].

Cependant, les nœuds proches de la BS participent plus que les autres aux tâches de routage. De plus, ces réseaux présentent une faible scalabilité dû au fonctionnement identique des nœuds et d'une manière distribuée nécessitant ainsi un grand nombre de messages de contrôle [25].

Les réseaux plats sont caractérisés par : la simplicité des protocoles de routage, un coût de maintien réduit, une grande tolérance aux pannes ainsi qu'une habilité à construire de nouveaux chemins suite aux changements de topologie.

Deux exemples phares de cette classe de protocoles sont SPIN et DD.

#### ✓ Le protocole SPIN (*Sensor Protocols for Information via Negotiation*)

C'est un protocole de routage Data-centric qui utilise des techniques de négociation afin d'éliminer les problèmes de redondance de données dans le routage. Il est basé sur l'idée que les nœuds capteurs opèrent plus efficacement et conservent l'énergie en envoyant des données qui décrivent les données des capteurs au lieu d'envoyer les données entières, à moins que les données entières ne soient explicitement demandées. Cela permet de pallier au problème d'inondation [25].

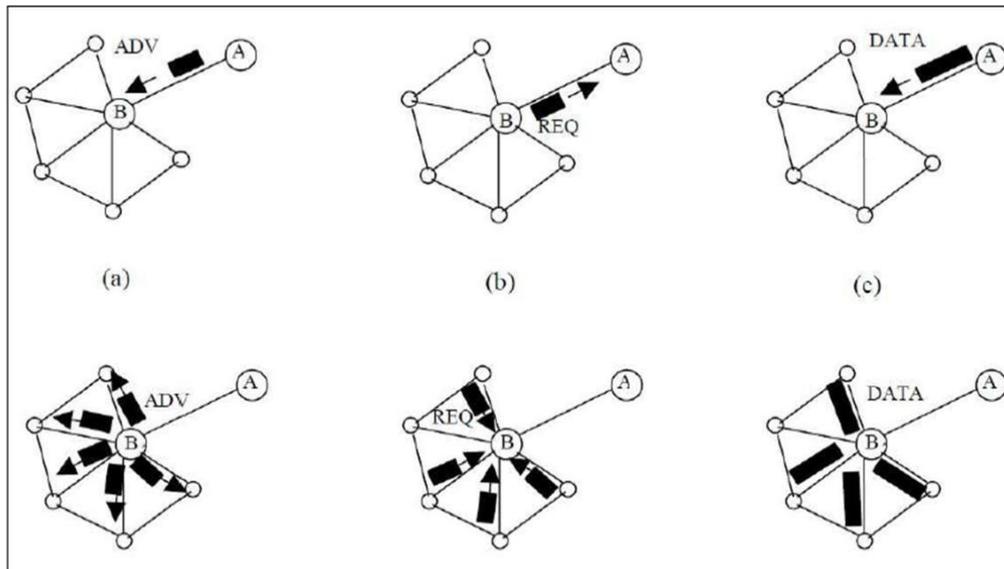
L'idée consiste à nommer les données en utilisant des descripteurs de haut niveau ou des méta-données. Avant la transmission, les méta-données sont échangées entre les capteurs par un mécanisme de publicité de données. Chaque nœud recevant de nouvelles données, les annonce à ses voisins et ceux intéressés récupèrent les données en envoyant une requête [26].

Les communications dans SPIN se font en trois étapes (voir figure 1.13) avec trois types de messages: 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. Or, 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 [27].



**Figure 1.13.** Le protocole SPIN [23].

L'avantage principal de SPIN est l'élimination des envois redondants des données en utilisant la négociation. En plus, les changements de la topologie n'affectent pas les performances du protocole réseau puisque chaque nœud ne se base que sur les informations locales pour la prise de décision; il n'a besoin de communiquer des informations de contrôle qu'avec ses voisins directs.

Néanmoins, cet avantage peut générer un inconvénient majeur qui est le non garanti de la livraison des données dans le cas où le nœud ayant des données sollicitées est injoignable [28].

### ✓ Le protocole DD (*Direct Diffusion*)

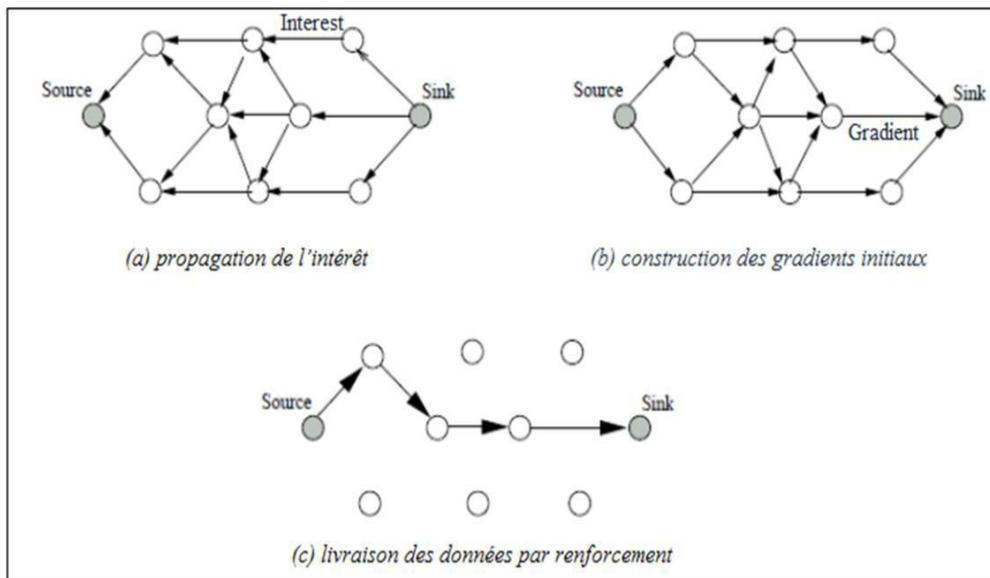
C'est un protocole de propagation de données, permettant d'utiliser plusieurs chemins pour le routage d'information. La diffusion dirigée suggère l'utilisation d'un schéma de nommage sous forme de pair « Attribut-Valeur » pour les requêtes et les données. Le principe de fonctionnement de ce protocole est le suivant:

Le nœud Sink commence à envoyer une requête d'intérêt « Interest » vers la destination à travers ses nœuds voisins pour démarrer une application bien déterminée, et chaque nœud recevant l'intérêt crée un gradient vers le nœud source de cet intérêt. A

savoir que chaque nœud est défini à l'aide d'une liste de paires attribut-valeur comme le nom des objets, l'intervalle, la durée, la zone géographique, etc...et qu'un gradient est un lien réponse de la part du voisin recevant l'intérêt et qui est caractérisé par le débit, la durée et la date d'expiration de données.

De cette manière, plusieurs routes reliant le sink à la source de données sont réalisées; puis la meilleure route sera renforcée pour éviter la redondance. Le sink renvoie le message d'intérêt initial à travers la route choisie. Un intervalle plus petit renforce donc le nœud source sur ce chemin pour envoyer des données plus fréquemment [25].

La figure 1.14 illustre les phases de fonctionnement de ce protocole.



**Figure 1.14.** Fonctionnement du protocole DD [23].

Si la route choisie échoue, une nouvelle ou une alternative doit être identifiée, puisque les données sont demandées par des requêtes.

En outre, cette méthode est coûteuse en termes de consommation d'énergie et ne représente pas un bon modèle pour les RCSFs.

### 7.3.1.2. Les protocoles géographiques

Ces protocoles qui sont appelés aussi les protocoles de routage basés sur la localisation utilisent les informations d'emplacement pour guider la découverte de routage et la transmission des données. Ils permettent la transmission directionnelle de l'information en évitant l'inondation d'information dans l'ensemble du réseau [28].

Dans ce type de protocoles, chaque nœud du réseau connaît sa position et celle de ses voisins. Le positionnement du nœud peut être obtenu en utilisant un système de géo-positionnement tel que le GPS ou bien via des algorithmes de positionnement relatif [28]. Il peut donc calculer sa distance et celle de ses voisins à la destination et envoyer l'information à son voisin qui le rapproche le plus de la destination finale.

Une estimation de la consommation d'énergie est réalisée au préalable pour désigner le chemin le plus rentable énergétiquement en obligeant les nœuds, qui ne sont pas sur le chemin du routage choisi, à entrer en mode veille.

Avec la topologie réseau basée sur des informations de localisation de nœuds, la gestion du réseau devient simple, mais le fait que chaque nœud doit connaître les emplacements des autres nœuds reste l'inconvénient majeur de ces protocoles [29].

Nous allons présenter, dans ce qui va suivre, deux solutions phares de ce type de routage, qui sont : le protocole GAF et le protocole GEAR.

### ✓ **Le protocole GAF (*Geographic adaptive fidelity*)**

C'est un protocole de routage basé sur la localisation des nœuds. Il est conçu principalement pour les réseaux mobiles ad hoc, mais peut être applicable aux réseaux de capteurs.

Ce protocole découpe le réseau en plusieurs zones virtuelles; chaque nœud est affecté à une zone donnée selon ses coordonnées géographiques (en utilisant le GPS).

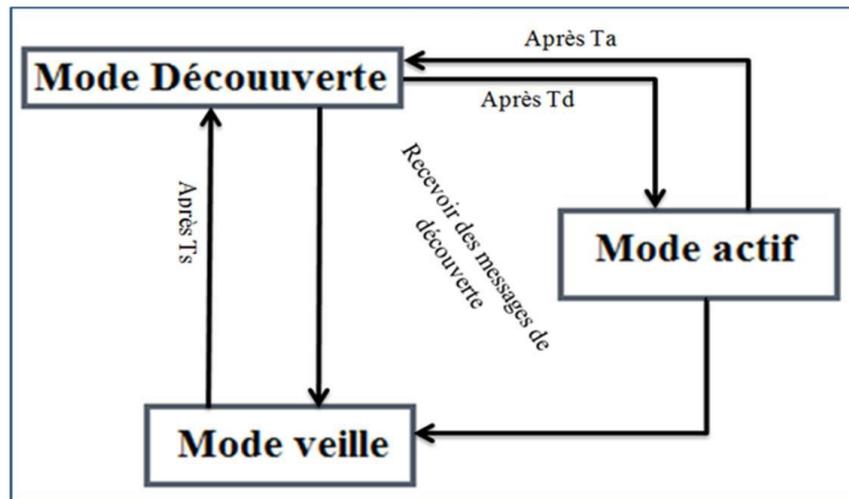
Les nœuds appartenant à la même zone sont considéré comme équivalents en coût de routage; ainsi, on choisit seulement un nœud de chaque zone pour router les données et les autres sont misent en mode veille pour conserver de l'énergie. De cette façon, la durée de vie du réseau est augmentée en fonction du nombre total des nœuds inactifs [23].

Dans GAF, le nœud peut se retrouver dans l'un des trois cas possibles:

- **Mode découvert:** pour déterminer ses voisins de zone.
- **Mode actif:** s'il participe dans le routage de données.
- **Mode endormi:** s'il ne participe pas au routage.

Les transitions d'état de GAF sont représentées sur la figure 1.15.

Quel nœud va dormir et pendant combien de temps sont des paramètres qui dépendent de l'application et en conséquence les paramètres liés sont ajustés pendant le processus de routage [23].



**Figure 1.15.** Transitions des états dans GAF [23].

En effet, un seul nœud dans chaque grille reste à l'état actif en faisant passer les autres nœuds de la grille à l'état de veille pour une certaine période de temps tout en assurant la fidélité du routage.

Cependant, dans certains environnements où les nœuds sont fortement mobiles, la fidélité du routage pourrait être réduite si un nœud actif quitte la grille. Ainsi, le nombre de données perdues sera important. Pour supporter la mobilité, chaque nœud estime et diffuse le temps nécessaire pour quitter sa zone à ses voisins. Sur la base de ce temps ils estiment leurs temps de réveil pour choisir celui qui va prendre le relais parmi eux.

Nous pouvons dire que GAF est un protocole géographique et hiérarchique en même temps, car, à chaque instant chaque zone est représentée par un seul nœud pour router les données qui transitent via cette zone; à la différence, que ce représentant n'accomplit aucune tâche de fusion ou d'agrégation de données. En d'autres termes, les nœuds d'une zone constituent un groupe défini par leur positionnement dans une même zone délimitée physiquement.

Il faut noter que GAF ne prend pas en compte l'énergie du nœud lors du choix du représentant, ce qui peut générer des trous dans le réseau une fois que ce dernier épuise son énergie [30].

### ✓ Le protocole GEAR (*Geographic and energy-aware routing*)

Ce protocole est considéré comme une amélioration du protocole DD en se basant sur la géolocalisation et l'énergie des nœuds voisins.

Il découpe le réseau en régions dont l'idée principale est de limiter le nombre et la diffusion des intérêts vers des zones spécifiques au lieu de l'ensemble du réseau. De cette manière, une conservation d'énergie est obtenue par rapport à DD [28].

Avec le protocole GEAR, chaque nœud maintient le coût pour atteindre la destination en passant par ses voisins. Ce coût est divisé en deux parties : un coût estimé et un coût d'apprentissage. Le coût estimé est une combinaison de l'énergie résiduelle et la distance vers la destination. Le coût d'apprentissage consiste en un réajustement du premier coût en prenant en compte le routage à travers les trous qui peuvent apparaître sur le réseau. Un trou apparaît lorsqu'un nœud n'a aucun voisin sur un chemin vers le destinataire. S'il n'y a pas de trous, le coût estimé est égal au coût d'apprentissage. Le coût d'apprentissage est propagé dans le sens inverse chaque fois qu'un paquet est reçu par le destinataire pour ajuster le coût de la route pour les paquets qui vont suivre.

L'acheminement des paquets suit les étapes suivantes [25]:

- Acheminer le paquet jusqu'à la région, en envoyant le paquet au nœud le plus proche de la région cible parmi ses voisins et ayant le niveau d'énergie résiduelle le plus élevé.
- Acheminer le paquet dans la région de destination par une sorte de diffusion si le nombre de nœuds n'est pas élevé, sinon la région est découpée en sous-régions et le paquet est transmis individuellement à chaque sous-région.

### 7.3.1.3. Les protocoles hiérarchiques

Le routage hiérarchique est considéré comme étant l'approche la plus favorable en termes de simplicité et d'efficacité énergétique.

Les protocoles hiérarchiques comme nous avons dit précédemment restructurent le réseau global en groupes dits clusters, chacun d'eux est constitué d'un chef appelé CH et de ses membres. Suivant l'application, les membres peuvent être des voisins directs du chef ou pas dont ils acheminent leurs messages à leur chef, lequel les achemine ensuite dans le réseau tout entier via d'autres CHs jusqu'à la BS.

Le point fort de ce type de protocoles est l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis au sink, ce qui implique une meilleure économie d'énergie.

Le problème qui peut se poser dans cette topologie est la surcharge des CHs qui induit un déséquilibre de la consommation d'énergie dans le réseau. Pour remédier à ce problème, CHs peuvent être des capteurs spécifiques avec plus de ressources énergétiques et plus de capacités de traitement ou bien ils peuvent être élus dynamiquement et ainsi garantir un équilibre de la consommation d'énergie et augmenter la tolérance aux pannes [25].

Vu l'importance de ce type de protocoles dans la suite de notre travail, nous avons décidé de lui consacrer une grande partie dans le chapitre suivant pour mieux expliquer cette nouvelle approche de conception.

#### **7.3.1.4. Les protocoles considérant les qualités de services**

Dans cette catégorie, le protocole essaye de trouver un compromis entre la consommation de l'énergie et un ou plusieurs qualités de services lors de la livraison de données; ces qualités de services peuvent être le délai, la bande passante, ...etc. Un exemple de ce type de routage est le protocole SPEED. C'est un protocole géographique conçu pour les communications en temps réel sur les RCSFs. SPEED améliore le protocole géographique GEAR en prenant en compte le délai de livraison de données. Dans ce but, il estime le délai sur chaque saut du chemin de livraison en calculant le temps d'aller-retour d'un paquet (en retranchant le temps de traitement pour le nœud destinataire). Le prochain saut est choisi parmi les nœuds proches du destinataire garantissant le meilleur délai [23].

L'avantage majeur de ce type de protocoles reste la prise en compte des délais de transmissions qui les rend très recommandés à des applications de surveillance (centres nucléaires, applications militaires, etc.). Or, l'approche doit prendre en considération la contrainte d'énergie en parallèle avec les critères des qualités de services [23].

### **Conclusion**

Nous avons présenté dans ce chapitre, des généralités sur les réseaux de capteurs sans fil. Nous allons discuter également les principales caractéristiques et limites qui influencent la conception des réseaux de capteurs sans fil. Par la suite, nous décrirons la problématique de la consommation d'énergie dans les RCSFs. Ensuite, nous présenterons une vue globale des techniques proposées dans la littérature pour la gestion de la consommation de l'énergie. Finalement on a donné une vue globale sur les techniques de routage dans les réseaux de capteurs sans fil et qui ont comme un objectif commun la minimisation de la consommation d'énergie et donc le prolongement de la durée de vie du réseau. La description que nous avons faite sur ces réseaux fournira au lecteur les bases nécessaires à la compréhension de tous les concepts liés à ce type de réseaux.

## Chapitre 2

# Etat de l'art : Le clustering & les techniques computationnelles

*" Les scientifiques sont des gens affligés*

*par le besoin maladif de tout expliquer."*

**Grégoire LACROIX**

---

### Introduction

Depuis l'apparition de l'informatique, l'ensemble de données stockées sous forme numérique ne cesse de croître de plus en plus rapidement partout dans le monde. Il existe dès lors un grand intérêt à développer des techniques permettant d'utiliser aux mieux tous ces stocks d'informations telles que le clustering, afin d'en extraire un maximum de connaissances utiles que représentent dans notre cas d'étude les informations transmises par les nœuds capteurs d'un RCSF.

Le but de ce chapitre est de présenter une vue générale du problème du clustering au sein des RCSFs, en introduisant les notions et les concepts de base sur lesquels s'appuiera la suite de ce mémoire. Nous commençons ce chapitre par rappeler quelques concepts et définitions essentiels pour comprendre les différentes méthodes et outils de classification automatique (clustering). Nous présentons par la suite les étapes de base de la procédure clustering ainsi que ses différentes applications possibles. Puis, nous discutons les différentes notions qui sont utilisées pour définir toute méthode de clustering.

L'utilisation de clustering sur les RCSFs propose quelques problèmes tels que la consommation d'énergie et le retard lors des opérations de transmissions. L'intelligence computationnelle est un domaine très intéressant englobant des techniques efficaces qui sont établies pour résoudre une grande gamme de ces problèmes complexes d'optimisation. La plupart de ces techniques sont inspirées des systèmes naturels.

Dans la suite de ce chapitre nous présentons une initiation à l'intelligence computationnelle puis sa définition. Ensuite, nous présentons un aperçu de ses

différentes techniques et méthodes les plus pondérées et leur concept de base ainsi que les différentes combinaisons synergiques. Nous terminons ce chapitre par la citation de quelques travaux dans la résolution de problème de la consommation d'énergie dans les RCSFs lors de l'utilisation de ce type de classification.

## 1. Vue générale du problème de clustering

### 1.1. Définition d'une partition

Étant donnée un ensemble fini d'objets noté  $I$ , on appelle partition de  $I$  toute famille de parties  $P$  non vides disjoints deux à deux dont l'union forme l'ensemble  $I$ .

$P = \{C_i, i \in I\}$ , tel que  $C_i$  : partie de  $I$  (ou une classe) possédant les propriétés suivantes [31]:

1.  $\forall i \in I, C_i \neq \emptyset$
2.  $\forall i \in I, \forall j \in I, i \neq j \rightarrow C_i \cap C_j = \emptyset$
3.  $\bigcup_{i \in I} C_i = I$

### 1.2. Rappels sur la classification

Les méthodes de classification de manière générale ont pour but d'identifier les classes auxquelles appartiennent des objets à partir de certains traits descriptifs. Cependant, il existe deux variantes de classification: la classification dite supervisée et la classification dite non supervisée.

Dans l'approche de la classification supervisée, les classes existent a priori. On dispose en donnée du problème d'un ensemble de classes et d'objets, chacun d'eux étant déjà placé dans une classe qui lui convient au mieux. Le but de cette démarche est de pouvoir allouer une classe à un nouvel objet, en restant le plus cohérent possible avec la structure initiale en classes.

Dans une classification non supervisée, les classes sont encore inexistantes. Pour cette démarche, on dispose donc au départ d'un ensemble d'objets. L'idée consiste à découper l'ensemble des objets en groupes (clusters) de telle sorte que les caractéristiques des objets dans un même cluster soient similaires et les caractéristiques des objets dans des clusters différents soient distinctes [32]. Dans la suite de ce chapitre nous allons nous intéresser exclusivement à l'approche non supervisée, encore nommée *clustering* ou *segmentation*.

#### 1.2.1. La classification supervisée (ang. classification)

Si les classes possibles sont connues et si les exemples sont fournis avec l'étiquette de leur classe on parle de classification supervisée (ang. classification), L'objectif est alors d'apprendre à l'aide d'un modèle d'apprentissage à partir de l'ensemble des exemples (appelé ensemble d'apprentissage) des règles qui permettent de prédire la classe des

nouveaux exemples ce qui revient à découvrir la structure des classes afin de pouvoir généraliser cette structure sur un ensemble de données plus large.

Un exemple d'application de la classification supervisée concernant les voitures peut être tiré de [33], ou il peut s'agir par exemple de déterminer si une nouvelle voiture rencontrée fait partie de la classe des citadines, des voitures intermédiaires ou des voitures confortables, en se basant sur des caractéristiques, et sur la classe connue des voitures déjà rencontrées (exemples d'apprentissage). Le tableau 2.1 présente le problème dans ce cadre. À chaque exemple est associée la classe de voiture à laquelle il appartient. L'objectif est alors d'être capable d'estimer la classe la plus appropriée à tout nouvel exemple rencontré (par exemple la voiture 7 dans le tableau 2.1) [31].

identifiant	carburant	cylindres	longueur	puissance	Classe
1	gpl	8	186	6000	confort
2	Essence	4	170	5800	intermédiaire
3	diesel	6	172	5500	intermédiaire
4	diesel	4	156	5200	citadine
5	Essence	12	190	5500	confort
6	Essence	4	175	5800	intermédiaire
7	diesel	6	170	6000	?

**Tableau 2.1.** Exemple d'application de l'apprentissage supervisé [31].

### 1.2.2. La classification non supervisée (ang. clustering)

On parle alors de classification non supervisée appelée aussi "classification automatique", "clustering" ou encore "regroupement", si seulement les exemples, sans étiquettes, sont disponibles, et si les classes et leurs nombres sont inconnus. La classification non supervisée (ang. clustering) consiste à diviser un ensemble d'exemples en sous-ensembles, appelés classes (clusters), tels que les objets d'une classe sont similaires et que les objets de classes différentes sont différents, afin d'en comprendre la structure [34]. Autrement dit, il s'agit à ce niveau de rechercher la distribution sous-jacente des exemples dans leurs espaces de description. Comme le montre le tableau 2.2, il n'y a aucune information de classe associée aux exemples dans ce cas [31].

identifiant	carburant	cylindres	longueur	puissance	Classe
1	gpl	8	186	6000	?
2	Essence	4	170	5800	?
3	diesel	6	172	5500	?
4	diesel	4	156	5200	?
5	Essence	12	190	5500	?
6	Essence	4	175	5800	?
7	diesel	6	170	6000	?

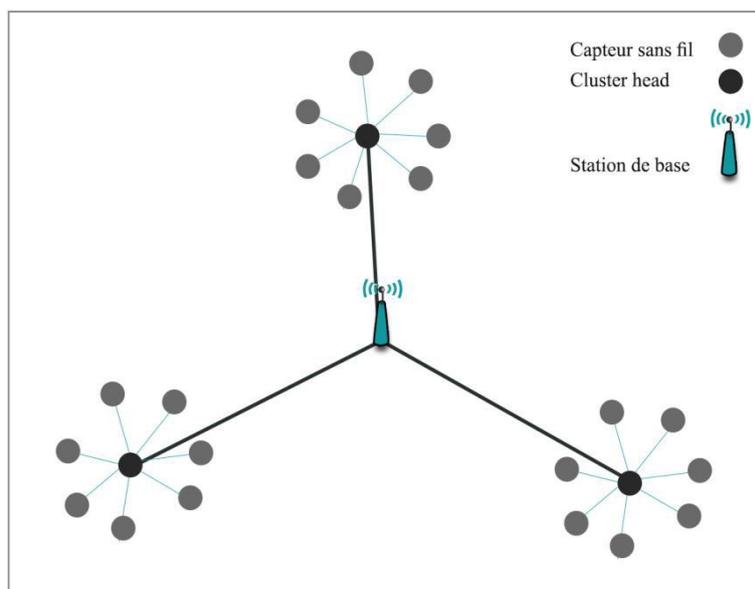
**Tableau 2.2.** Exemple d'application d'apprentissage non supervisé [31].

## 2. Le clustering et le les réseaux de capteurs sans fil

Il est parfois préférable d'avoir des algorithmes de routage qui prennent en compte la structure logique du réseau, plus précisément lorsque le nombre de nœuds dans le réseau est très important. Une des approches possibles est l'utilisation du Clustering. La mise en Cluster des nœuds d'un RCSF fait référence au processus qui permet de partitionner ces nœuds de manière à obtenir un nombre de grappes (Cluster) indépendantes. Chaque Cluster est constitué d'un sous-ensemble de nœuds de capteurs et possède une tête de Cluster (CH). Le rôle de ce dernier est double; Tout d'abord, un CH collecte les données des nœuds de son Cluster et, deuxièmement, il envoie les données reçues au récepteur, comme illustré dans la figure 2.1.

Le but ultime de la mise en Cluster des nœuds de capteurs est de réduire les coûts de communication. En effet, ceci pourrait être réalisé car les nœuds de capteurs doivent communiquer avec leur CH uniquement sans avoir à arbitrer directement sur le nœud puits qui peut être éloigné. Par ailleurs, seuls quelques nœuds doivent communiquer avec la station de base.

Cependant, la mise en Cluster des nœuds d'un RCSF est une tâche ardue en raison de la nature dynamique de ce réseau qui entraîne de fréquents changements dans les CH. Cela rend le problème plus complexe. De plus, un tel comportement rend très difficile l'élaboration d'un modèle mathématique [35].



**Figure 2.1.** Réseau de capteurs sans fils en Cluster [35].

### 3. Les étapes d'un clustering

Les étapes de base du processus de classification automatique ou clustering sont présentées dans la figure 2.2 et peuvent être récapitulées comme suit [31]:

#### 3.1. Sélection/extraction des caractéristiques

La sélection des caractéristiques est le processus d'identification d'un sous ensemble optimal de caractéristiques d'origine pertinentes pour un critère fixé auparavant pour les utiliser dans le regroupement. La sélection de ce sous ensemble de caractéristiques permet d'éliminer les informations non pertinentes et redondantes selon le critère utilisé. Tandis que l'extraction des caractéristiques vise à l'utilisation d'une ou plusieurs transformations des caractéristiques d'entrées pour produire de nouvelles caractéristiques saillantes. Chacune de ces techniques ou les deux peuvent être utilisées pour obtenir un ensemble approprié de caractéristiques à utiliser dans le clustering.

#### 3.2. Algorithme de classification automatique

L'étape de regroupement ou classification peut être effectuée de plusieurs façons. La classification de données (ou clustering) peut être dure (une partition de données en groupes) ou floue (où chaque modèle a un degré d'appartenance à chacun des clusters de sortie). Ainsi, notre objectif à travers cette étape est de choisir l'algorithme de clustering le plus approprié pour le regroupement de l'ensemble de données où chaque algorithme de clustering est caractérisé principalement par une mesure de proximité et un critère de regroupement.

- **La mesure de proximité :** est une mesure qui quantifie à quel degré deux points de données quelconque sont "similaire" (vecteurs des caractéristiques). Dans la plupart

des cas nous devons nous assurer que tous les variables choisis contribuent également au calcul de la mesure de proximité.

- **Critère de regroupement** : peut être exprimé par une fonction de coût ou d'autre type de règles. il est nécessaire de prendre en compte le type des clusters attendus par le regroupement de l'ensemble de données. Ainsi, nous pouvons définir un "bon" critère de regroupement, menant à un partitionnement qui représente le mieux que possible l'ensemble de données.

### 3.3. Validation des résultats

Les algorithmes de regroupement permettent d'extraire des clusters qui ne sont pas connus à priori. En outre, différentes approches conduisent généralement à différents groupes et même pour le même algorithme. Par conséquent une classification finale d'un ensemble de données exige un certain genre d'évaluation dans la plupart des applications. L'exactitude des résultats obtenus par les algorithmes de regroupement est vérifiée en utilisant des techniques et des critères bien appropriés.

### 3.4. Interprétation des résultats

Le but ultime du regroupement est de fournir aux utilisateurs un aperçu significatif des données d'origine, afin qu'ils puissent résoudre efficacement les problèmes rencontrés.

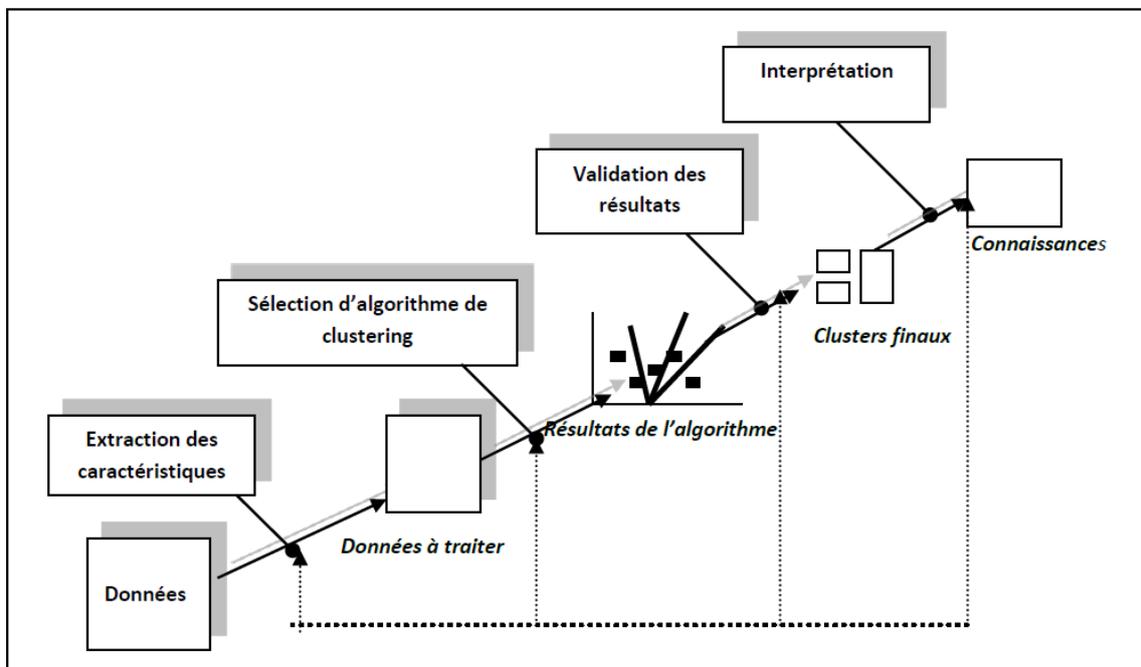


Figure 2.2. Les étapes d'un processus de clustering [31].

## 4. Application du clustering

La classification non supervisée a été utilisée dans plusieurs domaines, allant de l'ingénierie (apprentissage automatique, intelligence artificielle, reconnaissance des

formes, génie mécanique, génie électrique), l'informatique (Webmining, base de données spatiale, la collection de documents textuels, la segmentation des images), sciences médicales et vie (génétique, la biologie, la microbiologie, la paléontologie, la psychiatrie, clinique, pathologie), sciences de la terre (Géographie, géologie, télédétection), sciences sociales (sociologie, la psychologie, l'archéologie, l'éducation), et l'économie (marketing, commerce). Cette diversité reflète la position importante du regroupement dans la recherche scientifique. D'autre part, cette diversité peut être une source de confusion, en raison des terminologies et objectifs différents. Les algorithmes de clustering ont été développés pour résoudre des problèmes particuliers, dans des domaines spécifiques, et ils sont généralement basés sur des hypothèses et des suppositions sur l'ensemble de données à traiter. Ces suppositions affectent inévitablement les performances de ces algorithmes dans d'autres problèmes qui ne satisfont pas ces hypothèses. Par exemple, l'algorithme K-means basé sur la distance euclidienne et, par conséquent, il tend à générer des clusters hyper sphériques. Mais si les clusters réels sont dans une autre forme géométrique, K-means peut ne pas être efficaces, ce qui conduit à chercher d'autres algorithmes de clustering. Dans ce qui suit, nous décrivons quelques domaines d'applications où le clustering a été utilisé comme une étape essentielle [31]:

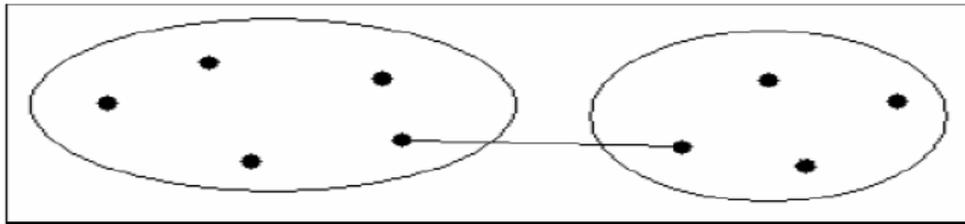
- La segmentation,
- La reconnaissance de formes et,
- Le datamining.
- **La segmentation** : La segmentation d'une base de données a pour objectif principal de réduire la taille des données afin de faciliter leur traitement. On parle dans ce cas de compression de données. Elle est parfois utilisée pour discrétiser une base de données, c'est à dire transformer la description complexe des objets par un unique attribut caractérisant leurs appartenance à une classe identifiée automatiquement.
- **Le datamining** : Vise à traiter les bases de données volumineuses qui imposent au clustering des exigences computationnelles supplémentaires. Ces défis ont conduit à l'émergence de puissantes méthodes de classification non supervisée largement utilisées afin de faciliter le regroupement de grands ensembles de données caractérisés par de nombreux attributs de différents types. La naissance du clustering dans le data mining est due principalement aux développements intenses dans les domaines de recherche d'informations et de Texte Mining, les applications de bases de données spatiales par exemple les données astronomiques, l'analyse de données, les applications Web, l'analyse de l'ADN en bioinformatique, et bien d'autres applications spécifiques.

- **La reconnaissance de formes :** Dans de nombreuses applications de reconnaissance de formes, il est extrêmement difficile ou coûteux, voire impossible, d'étiqueter de manière fiable un échantillon d'apprentissage avec sa véritable catégorie. Les applications typiques incluent la reconnaissance vocale et de caractère. Les algorithmes d'apprentissage de classification non supervisée ont été également utilisés pour la segmentation des images et vision par ordinateur. L'analyse typologique est une technique très importante et très utile. La vitesse, la fiabilité et la consistance avec laquelle un algorithme de clustering peut organiser de grandes quantités de données constituent de fortes raisons écrasantes de l'utiliser dans des applications de reconnaissance de formes.

#### 4.1. Rapprochement des clusters

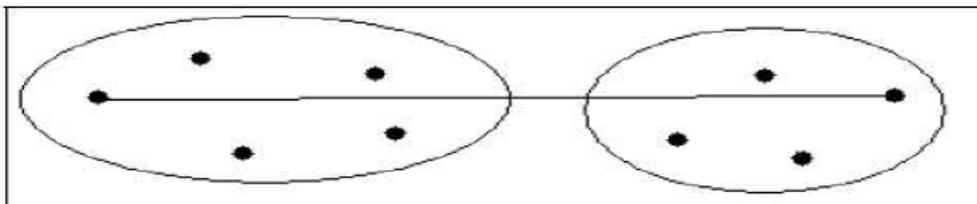
L'un des problèmes rencontrés au niveau du clustering est le choix d'une technique de rapprochement des clusters. Le rapprochement entre deux clusters consiste à déterminer quels concepts prendre dans chacun des clusters pour définir la mesure de distance. Cependant, ce rapprochement se fait sur la base de mesures de distances calculées à partir des valeurs prises par les concepts du cluster. Plusieurs techniques de rapprochement entre clusters existent, parmi celles-ci nous pouvons citer [32]:

- **Lien simple (single link) :** il définit la distance entre deux clusters comme étant la plus petite distance entre toutes les paires de concepts des deux clusters.



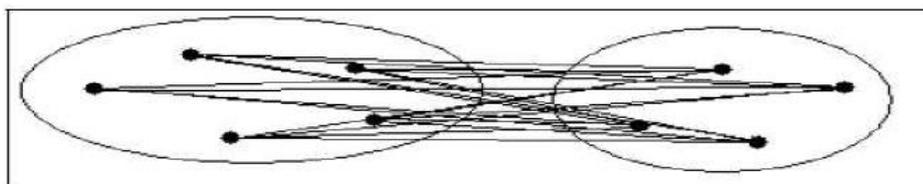
**Figure 2.3.** Distance entre les deux clusters par lien simple [32].

- **Lien complet (complet link) :** il définit la distance entre deux clusters comme étant la plus grande distance entre toutes les paires de concepts des deux clusters.



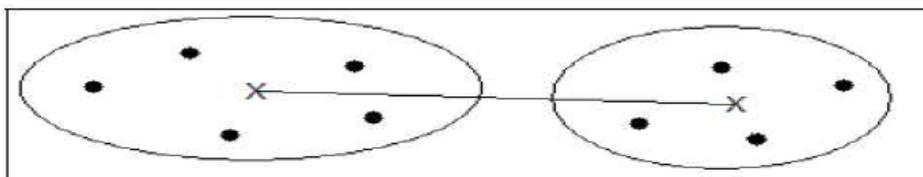
**Figure 2.4.** Distance entre les deux clusters par lien complet [32].

- **Lien moyen (average link) :** il définit la distance entre deux clusters en faisant intervenir tous les objets présents dans ces clusters.



**Figure 2.5.** Distance entre deux clusters par lien moyen [32].

- **Lien moyen de groupe (Group average link) :** il définit la distance entre deux clusters comme étant la distance entre les centres des clusters.



**Figure 2.6.** Distance entre deux clusters par lien moyen de groupe [32].

## 5. Quelques approches classiques du clustering

Le clustering se catégorise en plusieurs familles de méthodes selon la stratégie suivie pour construire les clusters. Parmi ces méthodes on trouve :

### 5.1. Clustering hiérarchique

La méthode hiérarchique génère une partition de l'espace de données, mais aussi une succession de partitions de l'espace des données. Celles-ci sont souvent représentées sous la forme d'un dendrogramme. Un dendrogramme est un arbre de partitions successives de l'espace de données. Selon que l'on parcourt le dendrogramme de haut en bas ou de bas en haut, la méthode sera dite *descendante* (division) ou *ascendante* (agglomération).

Le principe de la méthode hiérarchique descendante consiste à commencer avec un groupe contenant tous les nœuds. Puis de procéder à une division du groupe en plus petits groupes jusqu'à ce que chaque groupe ne contienne plus qu'un exemple.

Dans la méthode hiérarchique ascendante, on commence avec autant de clusters qu'il y a de nœuds. Ensuite, à chaque étape, on regroupe les deux clusters qui sont jugés les plus similaires pour terminer avec un seul grand cluster englobant toutes les données.

Nous avons utilisé cette dernière, qui suit l'algorithme suivant [32]:

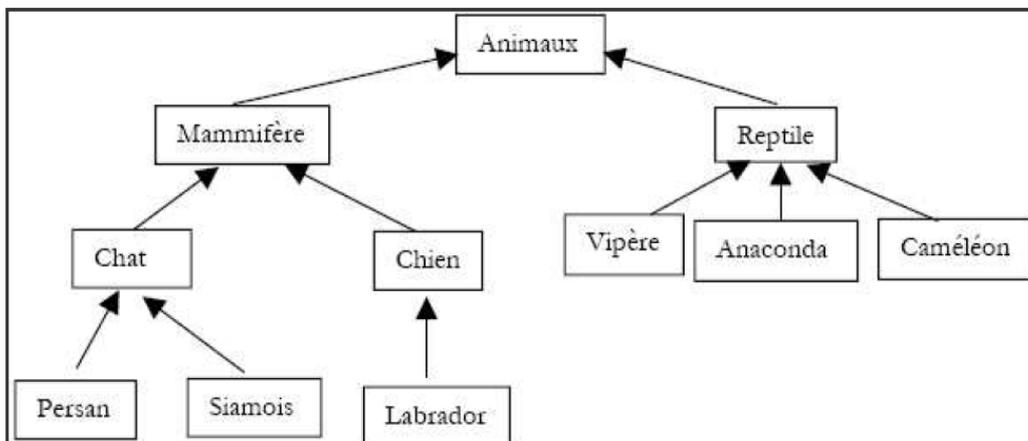
- Chaque nœud est placé dans son propre cluster.
- Calcul de la matrice de distance  $M$  entre chaque couple de clusters.
- Répéter :
  - ✓ Sélectionner dans  $M$  les deux clusters les plus proches.
  - ✓ Fusionner les clusters par un cluster plus général.

- ✓ Mettre à jour M en calculant la distance entre le cluster créé et les autres clusters existants (techniques de rapprochements entre deux clusters)
- Jusqu'à K (avec K le nombre de regroupements à effectuer).

Afin de déterminer le nombre optimal de clusters, on coupe l'arbre (dendrogramme) au moment où l'on commence à rassembler des clusters dits similaires. Comme la distance entre clusters varie en fonction de l'ontologie utilisée, dans le cas de dendrogramme suivant de représentant celui de la taxonomie de la figure 2.7, nous avons fixé la distance minimale entre cluster, à trois afin d'obtenir le nombre optimal de clusters. La coupe est présentée en traits pointillés sur la figure 2.8. Dans notre exemple, nous avons utilisé un paramètre en entrée qui représente le nombre de nouveaux clusters créés. Dans le nombre de groupe de concepts est égal à la différence entre le nombre de concepts de l'ontologie et le nombre de nouveaux clusters créés.

Pour le problème de technique de rapprochement entre les clusters, nous avons utilisé le principe du lien simple décrit dans la section précédente.

De manière générale, la méthode hiérarchique ascendante est très intuitive et facile à implémenter mais coûteuse à partir du moment où les clusters sont comparés deux à deux à chaque étape ( $n(n-1)/2$  calculs effectués à chaque étape avec n le nombre de clusters) [32].



**Figure 2.7.** Exemple de taxonomie [32].

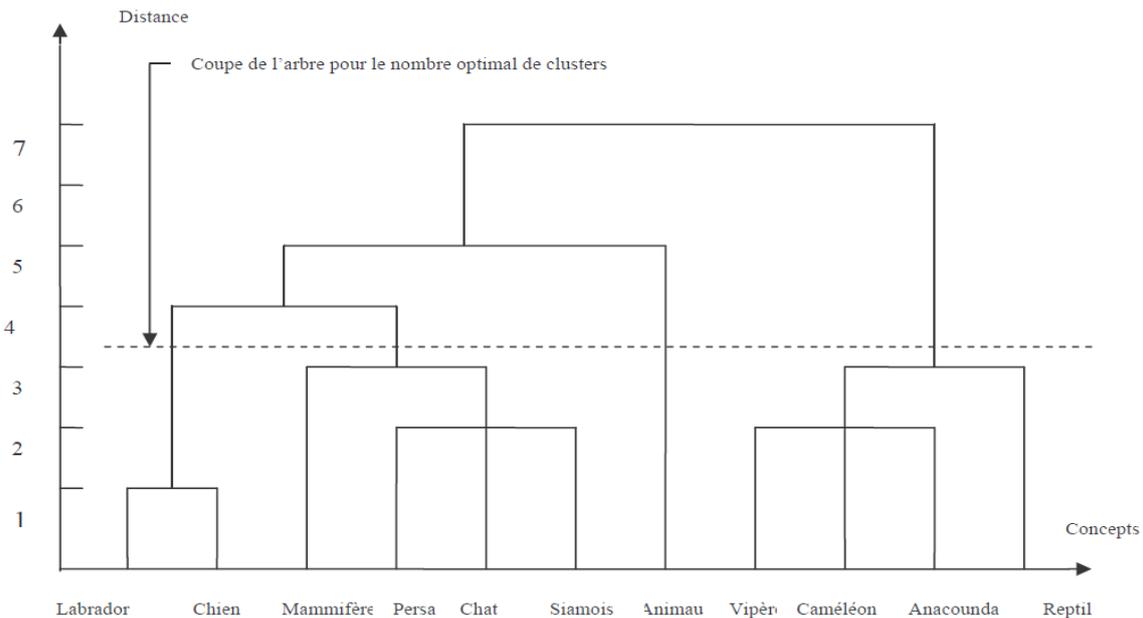


Figure 2.8. Exemple de dendrogramme [32].

### 5.1.1. Protocole de routage hiérarchique LEACH

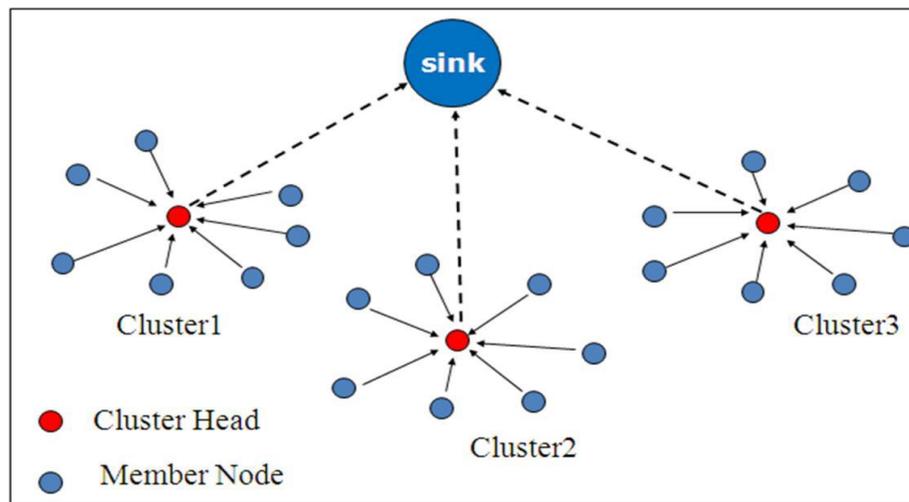
LEACH « *Low-Energy Adaptive Clustering Hierarchy* » est considéré comme étant le premier protocole de routage hiérarchique basé sur les clusters qui était proposé par Heinzelman et al. comme étant l'un des algorithmes de routage hiérarchique le plus populaire pour les réseaux de capteurs sans fil.

L'idée est de former des clusters de nœuds capteurs basés sur les zones où il y a un fort signal reçu, puis utiliser des CHs locaux pour atteindre la destination en optimisant la consommation d'énergie [29].

#### 5.1.1.1. Architecture de communication de LEACH

L'architecture de communication de LEACH consiste, de façon similaire aux réseaux cellulaires, à former des cellules basées sur l'amplitude du signal, et utiliser les têtes de cellules comme routeurs vers le nœud puits. Ces cellules sont appelées clusters, quant aux têtes, Les CHs sont choisis de façon aléatoire et périodique parmi les nœuds formant le cluster, en fonction de l'état de sa batterie. Puis, ils sont utilisés comme relais pour atteindre le puits suivant un algorithme qui utilise la rotation randomisée des têtes de groupe pour distribuer équitablement la charge d'énergie entre les nœuds de réseau [36].

Un nœud capteur décide quel cluster rejoindre en se basant sur la puissance des signaux reçus. A la formation des groupes. Comme montre la figure 2.9, tous les nœuds ordinaires transmettent leurs données à leur CH qui les agrège et transmettent, à leur tour, à la BS selon une communication unicast (à un seul saut).



**Figure 2.9.** Architecture de communication du protocole LEACH [23].

Les CHs ont pour mission d'assurer les fonctions les plus coûteuses en énergie, à savoir la communication avec le nœud puits qui est supposé éloigné, ainsi que tous les traitements de données (agrégation, fusion et transmission de données) afin de réduire la quantité des données transmises. Ce dispositif permet d'économiser l'énergie puisque les transmissions sont uniquement assurées par les CHs plutôt que par tous les nœuds du réseau. Par conséquent, LEACH réalise une réduction significative de la dissipation d'énergie [37].

### 5.1.1.2. Principe de fonctionnement de LEACH

Le protocole LEACH prend pour hypothèse l'égalité des énergies résiduelles des capteurs lors du démarrage de fonctionnement du réseau. La vie du réseau est alors segmentée en rounds caractérisés par un choix de CH.

Or, Chaque round est constitué de deux phases : la phase set-up qui est une phase d'initialisation et la phase steady-state, une phase de transmission [23].

#### ➤ La phase d'initialisation

La phase set-up est composée, à son tour, de trois sous-phases: d'annonce, d'organisation des groupes et enfin d'ordonnement :

- **Phase d'annonce :** Avant de lancer cette phase, nous désirons avoir un certain nombre de CH. Ce nombre, que l'on note  $K$ , est fixe et il est inchangé durant tous les rounds. Nous estimons que le pourcentage optimal du nombre de CHs désirés devrait être de 5% à 15% du nombre total de nœuds [36]. Si ce pourcentage n'est pas respecté, cela mènera à une grande dissipation d'énergie dans le réseau. En effet, si le nombre de CHs est très élevé, nous posséderons un nombre important des nœuds CHs qui se consacrent aux tâches très coûteuses en ressources énergétiques. Or, nous aurons une dissipation d'énergie considérable dans le

réseau. De plus, si le nombre de CHs est très petit, ces derniers vont gérer des groupes de grandes tailles. En conséquence, ces CHs s'épuiseront rapidement à cause de travail important qui leur est demandés.

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  $r$  [38]:

$$\text{Nombre CH} = \sum_{i=1}^N P_i(t) = K \quad (2.1)$$

Où  $N$  est le nombre total de nœuds dans le réseau. Si on a  $N$  nœuds et  $K$  CHs, 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 chaque nœud  $i$  pour devenir CH est:

$$P_i(t) = \frac{\text{le nombre de CH désirés}}{\text{le nombre de noeuds qui n'ont pas encore été élus CH durant les } r \text{ rounds précédents}} \quad (2.2)$$

$$P_i(t) = \begin{cases} \frac{K}{N - K * (r \bmod \frac{N}{K})} & C_i(t) = 1 \\ 1 & C_i(t) = 0 \end{cases} \quad (2.3)$$

Où  $C_i(t)$  égal à 0 si le nœud  $i$  a déjà été CH durant l'un des 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.

- **Phase d'organisation des clusters :** Après qu'un nœud soit élu CH, il doit informer les autres nœuds capteurs 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 autres nœuds en utilisant le protocole MAC CSMA « *Carrier Sense Multiple Access* » pour éviter les collisions entre les CHs. La diffusion permet de s'assurer que tous les nœuds ont reçu le message. Par ailleurs, elle permet de garantir que les nœuds appartiennent au CH qui requiert 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, c'est-à-dire le plus proche, sera choisi. En cas d'égalité des signaux, les nœuds ordinaires choisissent aléatoirement leur CH [38].

Chaque membre informe son CH de sa décision. Une fois que le CH ait reçu la demande, il lui envoie un message d'acquittement « Join- REQ ».

- **Phase d'ordonnement** : Après la formation des groupes, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données au sein de son groupe. Il crée un ordonnanceur TDMA « *Time Division Multiple Access* » et assigne à chaque nœud membre un slot de temps durant lequel il peut transmettre ses données. L'ensemble des slots assignés aux nœuds d'un groupe est appelé frame. La durée de chaque frame diffère selon le nombre de membres du groupe. Par ailleurs, afin de minimiser les interférences entre les transmissions dans des groupes adjacents, chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA « *Code Division Multiple Access* ». Il le transmet par la suite à ses membres afin de l'utiliser pour leurs transmissions [39].

#### ➤ La phase de transmission

Dans cette deuxième phase, le transfert de données vers la BS aura lieu. En utilisant l'ordonnanceur TDMA, les membres émettent leurs données captées pendant leurs propres slots. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leur énergie. Ces données sont ensuite agrégées par les CH qui les fusionnent et les compressent, et, envoient le résultat final au Sink.

Après un certain temps 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 [23].

#### 5.1.1.3. Avantages et inconvénients de LEACH

Bien que LEACH puisse augmenter la durée de vie du réseau en manipulant ses ressources tout en respectant plusieurs contraintes telle que la consommation d'énergie, il présente certaines limitations. Dans ce qui suit, nous citons quelques avantages et inconvénients du protocole LEACH [23]:

#### ➤ Avantages

Le protocole LEACH présente les avantages suivants:

- **Algorithme distribué** : l'auto-configuration des clusters se fait indépendamment de la BS.
- **Rotation des rôles de chefs de groupes**: les CHs sont choisis de façon aléatoire et périodique parmi les nœuds formant le cluster ce qui empêche la forte consommation d'énergie pour la transmission des données.

- **Faible énergie pour l'accès au média:** Le mécanisme de groupes permet aux nœuds d'effectuer des communications sur des petites distances avec leurs CHs afin d'optimiser l'utilisation du média de communication en la faisant gérer localement par un CH pour minimiser les interférences et les collisions.
- **Agrégations des données:** Les CHs compressent les données arrivant de leurs membres, et envoient un paquet d'agrégation au nœud puits afin de réduire la quantité d'informations qui doit lui être transmise. Cela permet de réduire la complexité des algorithmes de routage, de simplifier la gestion du réseau, d'optimiser les dépenses d'énergie et enfin de rendre le réseau plus évolutif.

### ➤ Inconvénients

- **Absence des CHs:** On pourra ne pas avoir des CHs durant un round si les nombres aléatoires générés par tous les nœuds du réseau sont supérieurs à la probabilité  $P_i(t)$ .
- **La distance entre le CH et les autres nœuds:** Les nœuds les plus éloignés du CH meurent rapidement par rapport aux plus proches.
- **Diminution de l'énergie des nœuds:** cette diminution est due à l'utilisation d'une communication à un seul saut au lieu d'une communication multi-sauts.
- **La rotation des CHs:** c'est une méthode qui n'est pas efficace pour de grandes structures de réseaux à cause de la surcharge d'annonces engendrées par le changement des CHs, et qui réduit le gain d'énergie initial.
- **La sécurisation:** le protocole LEACH n'est pas sécurisé. Aucun mécanisme de sécurité n'est intégré dans ce protocole. Ainsi, il est très vulnérable même aux simples attaques. Donc, un attaquant peut facilement monopoliser le réseau et induit à son dysfonctionnement.

### 5.1.2. Protocole de routage PEGASIS

Le protocole PEGASIS « *Power-Efficient Gathering in Sensor Information Systems* », proposé par Lindsey et Raghavendra en 2002, est un protocole basé sur les chaînes. L'idée de base du protocole est que, dans le but de prolonger la durée de vie du réseau, les nœuds vont être organisés de telle sorte à ce qu'ils forment une chaîne, n'auront ainsi besoin de communiquer qu'avec seulement leurs voisins les plus proches.

Pour localiser le voisin le plus proche, chaque nœud utilise la force du signal pour mesurer la distance vers tous les nœuds voisins, et ajuster par la suite la force du signal de telle sorte que seul un nœud peut être entendu. La forme agrégée des données sera envoyée à la BS par n'importe quel nœud dans la chaîne et les nœuds dans cette dernière vont se prendre en relais dans la transmission à la BS.

Donc, pour communiquer avec le puits, le processus est organisé en rounds; au cours de chaque round un seul nœud, appelé le leader, est autorisé à communiquer avec le puits directement et qui est un nœud représentatif d'une chaîne déjà présélectionné. Ce privilège est accordé à l'ensemble des nœuds du réseau à tour de rôle. Les données collectées sont transmises d'un nœud à un autre qui les agrège jusqu'à ce qu'elles arrivent au leader qui les transmet à la BS [36].

Par conséquent, PEGASIS a deux principaux objectifs. D'abord, augmenter la durée de vie de chaque nœud en employant des techniques de collaboration et augmenter par conséquent la durée de vie du réseau. En second lieu, permettre seulement la coordination locale entre les nœuds voisins de sorte que la largeur de bande consommée dans la communication soit réduite [23].

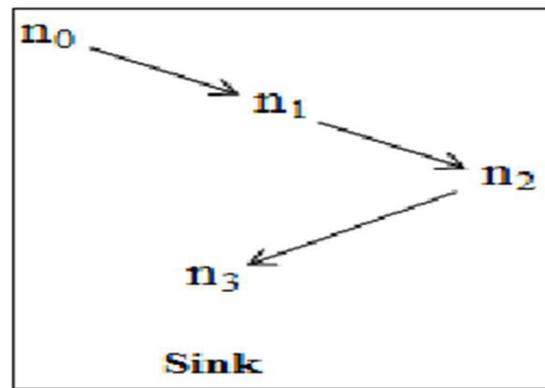
### 5.1.2.1. La construction des chaînes

Les nœuds vont être organisés de sorte qu'ils forment une chaîne, qui peut être soit calculée d'une façon centralisée par la BS et émise à tous les nœuds, ou accomplie par les nœuds capteurs eux-mêmes en employant un algorithme avide (Greedy algorithm). Si la chaîne est calculée par les nœuds capteurs, ils peuvent d'abord obtenir toutes les données sur l'emplacement des nœuds capteurs et calculent localement la chaîne en utilisant le même algorithme avide. Puisque tous les nœuds ont les mêmes données d'emplacement et exécutent le même algorithme, ils vont tous produire le même résultat.

Pour construire la chaîne, PEGASIS commence avec le nœud le plus éloigné de la BS. Le voisin le plus proche de ce nœud sera le nœud suivant dans la chaîne. Les voisins successifs sont sélectionnés de cette manière parmi les nœuds non visités afin de former la chaîne de nœuds [23].

L'algorithme commence par le nœud le plus lointain pour s'assurer que les nœuds les plus loin de la BS ont des voisins proches à mesure que, dans l'algorithme avide, les distances voisines augmenteront graduellement puisque des nœuds déjà présents sur la chaîne ne peuvent pas être revisités [23].

La figure 2.10 montre la construction de chaîne en utilisant l'algorithme avide où le nœud  $n_0$  se joint au nœud  $n_1$  qui se joint à son tour au nœud  $n_2$ , et le nœud  $n_2$  se joint au nœud  $n_3$ . Quand un nœud meurt, la chaîne est reconstruite de la même manière pour dévier le nœud mort.



**Figure 2.10.** Construction de chaînes en utilisant l'algorithme avide [23].

### 5.1.2.2. Déroutement de l'algorithme

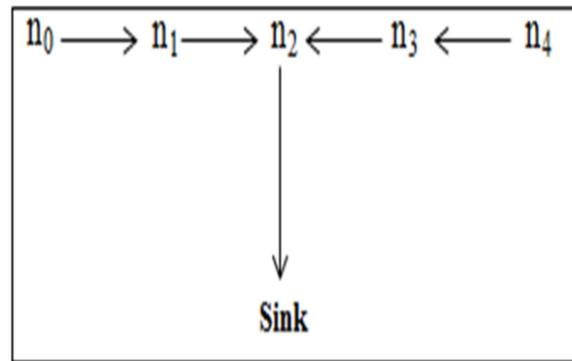
Pour collecter les données des nœuds capteurs dans chaque cycle, chaque nœud reçoit les données d'un voisin, les fusionne avec les siennes, et transmet à un autre voisin dans la chaîne. À noter que ce nœud, noté  $i$ , serait à une position aléatoire sur la chaîne que nous l'appelons  $j$ . Les nœuds se relient dans la transmission à la BS et PEGASIS va utiliser le nombre  $i \bmod N$  sachant que  $N$  représente le nombre de nœuds afin de transmettre à la BS dans le cycle  $i$ . Ainsi, le leader dans chaque cycle de communication sera à une position aléatoire sur la chaîne [23].

Chaque cycle de collecte de données peut être lancé par la BS avec un signal de balise qui synchronisera tous les nœuds capteurs. Puisque tous les nœuds connaissent leurs positions sur la chaîne, PEGASIS peut employer une approche de slot de temps, TDMA, pour la transmission des données. Dans le  $i^{\text{ème}}$  cycle de collecte de données, le nœud  $(i-1)$  sera leader.

Le nœud  $n_0$  transmettra ses données au nœud  $n_1$  dans le premier slot,  $n_1$  fusionne et transmet les données dans le deuxième slot, et ainsi de suite jusqu'à ce que le nœud leader soit atteint. Dans les slots suivants, les transmissions de données ont lieu depuis le nœud  $n_{n-1}$  et se déplacent vers le nœud leader de l'extrémité de la chaîne. Finalement, dans le  $n^{\text{ième}}$  slot, le leader transmet les données à la BS.

Alternativement, dans un cycle donné, PEGASIS peut utiliser une approche de déplacement de jeton à contrôle simple lancée par le leader pour commencer la transmission des données des extrémités de la chaîne. Le coût est très petit du fait que la taille du jeton est très petite. La figure 2.11 montre un exemple de déplacement de jeton.

Le nœud  $n_2$  est le leader et va passer le jeton le long de la chaîne commençant au nœud  $n_0$ . Le nœud  $n_0$  passera ses données au nœud  $n_2$ . Après que le nœud  $n_2$  ait reçu les données du nœud  $n_1$ , il passera le jeton au nœud  $n_4$ , et le nœud  $n_4$  passera ses données au nœud  $n_2$  avec fusion des données le long de la chaîne [23].



**Figure 2.11.** Approche de déplacement de jeton [23].

PEGASIS exécute la fusion de données à chaque nœud excepté les nœuds de fin de chaîne. Chaque nœud va fusionner les données de ses voisins avec les siennes afin de générer un paquet simple et les transmet par la suite à son autre voisin (s'il en a deux). Dans l'exemple précédent, Le nœud  $n_0$  transfère ses données au nœud  $n_1$ , qui agrège ses propres données avec celles du nœud  $n_0$ , puis les communique au nœud leader  $n_2$ . Le nœud  $n_2$  passe le jeton au nœud  $n_4$  qui transmet ses données au nœud  $n_3$ . Ce dernier agrège éventuellement ses données avec celles du nœud  $n_4$  et les transmet à son tour au nœud  $n_2$ , qui attend pour recevoir des données de ses voisins et puis les agrège avec ses propres données et transmet un message unique à la BS. Ainsi, dans PEGASIS, chaque nœud va recevoir et transmettre un paquet de données dans chaque cycle et sera le leader une fois chaque  $N$  cycles. En addition, les nœuds reçoivent et transmettent des paquets de contrôle de jeton très petits [23].

### 5.1.2.3. Avantages et inconvénients du protocole PEGASIS

Bien que le protocole PEGASIS engendre beaucoup d'avantages en ce qu'il offre comme bonne manipulation de ressources du réseau en respectant plusieurs contraintes telle que la consommation d'énergie, un nombre d'inconvénients restent plus ou moins apparents. Dans la suite, nous mentionnons quelques avantages et inconvénients de ce protocole [23] :

#### ➤ Avantages

- Utilisation d'agrégation des données qui minimise le nombre des transmissions et qui conserve l'énergie.
- Elimination de la phase de la construction des clusters pour chaque round qui génère une surconsommation d'énergie importante.

#### ➤ Inconvénients

- Bien que l'overhead du clustering soit évité, PEGASIS exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données.

- Ce protocole atteint rapidement ses limites de fonctionnement dans le cadre des réseaux fortement denses.
- 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.

## 5.2. Le clustering par partitionnement

Cette méthode consiste à diviser de manière optimale l'ensemble des instances en un groupe fini de groupes ( $K$ ). L'objectif est ici de minimiser une mesure de la dissemblance intra-groupe pour  $k$  groupes. Le problème étant lié à l'optimisation d'une combinatoire, la solution trouvée sera rarement l'optimum global mais plutôt un des nombreux optimums locaux. Parmi ces méthodes, on trouve : les  $K$ -moyennes, les  $K$  médoïdes ou le partitionnement autour des médoïdes (PAM) et la Carte Auto-Organisatrice [40].

### 5.2.1. Clustering K-Means

La méthode de  $K$ -Means ou algorithme de regroupement par centre mobile permet d'effectuer une classification d'un ensemble de données en  $K$  clusters. Un cluster regroupe plusieurs concepts similaires. Chaque cluster est décrit par son centre. Les centres des clusters sont mobiles au cours de l'exécution de l'algorithme.

L'algorithme peut se présenter comme suit [32]:

- ✓ Le nombre de clusters, le paramètre  $k$ , est fourni au départ.
- ✓ Un ensemble de  $K$  centres est choisi aléatoirement dans l'ensemble des données.
- ✓ Les  $K$  clusters sont formés en regroupant dans chaque centre l'ensemble de données plus proches du centre courant que de tout autre centre.
- ✓ Le centre de chaque cluster est calculé et devient le nouveau centre.
- ✓ L'algorithme boucle alors sur l'étape précédente : les données sont réaffectées en fonction de ces nouveaux centres et la condition d'arrêt est que les centres deviennent immobiles.

De manière générale, l'algorithme de clustering du  $K$ -Means est efficace, mais présente quelques faiblesses comme le nombre de clusters  $K$  qui doit être fixé a priori, ce qui oblige à initialiser les centres des clusters. L'initialisation des centres de clusters conditionne le résultat final (des initialisations différentes peuvent mener à des clusters différents, il peut arriver qu'un cluster ne conditionne que son centre) [32].

## 5.3. Le clustering spectral

Il est considéré également comme un clustering de partitionnement. Par rapport à des algorithmes classiques comme celui des  $K$ -moyennes, cette technique offre l'avantage de

classer des ensembles de données de structure « non-globulaire » dans un espace de représentation adéquat [40].

#### 5.4. Le clustering basé sur la densité

Il consiste à identifier dans l'espace de description des objets les régions de forte densité, entourées par des régions de faible densité pour former les clusters [40].

#### 5.5. Le clustering basé sur les grilles

Il consiste à partitionner l'espace en différentes cellules à l'aide d'une grille, puis à identifier les ensembles de cellules denses connectées pour la formation des clusters. Les méthodes appartenant à cette catégorie nécessitent deux paramètres à savoir : la taille de la grille et la densité minimum déterminant si une cellule de la grille est considérée comme dense ou non [40].

#### 5.6. Le clustering basé sur les graphes

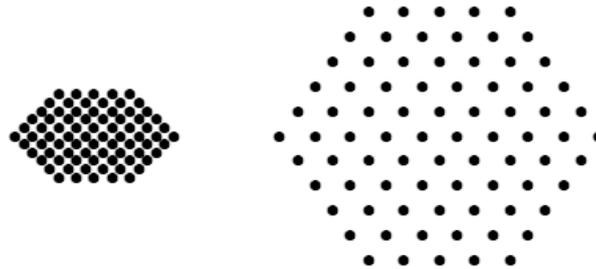
Il considère les clusters comme étant des ensembles de nœuds connectés dans un graphe. L'objectif est donc de former le graphe qui connecte les ensembles entre eux de telle manière que la somme des valeurs des arcs correspondant aux distances entre les exemples soit minimale [40].

### 6. Le choix de l'algorithme de clustering

Le choix de l'algorithme de clustering dépend en général de la nature des variables (quantitative et qualitative) dans les données et des clusters attendus (nombre, forme, densité, etc.). Généralement, les critères de décision peuvent être [40]:

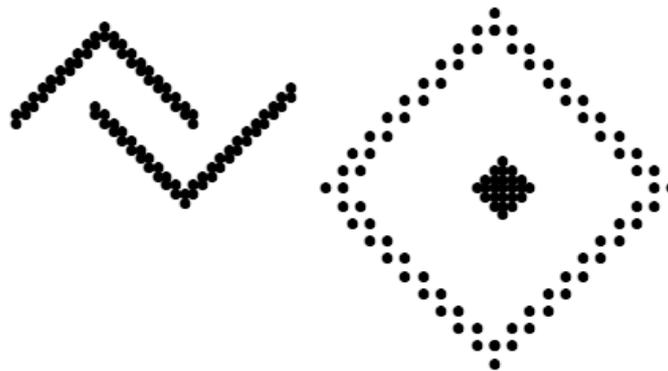
- **Les connaissances a priori** définissent l'ensemble d'informations concernant le nombre de clusters désiré, la distance minimale entre les clusters disjoints, etc.
- **La présentation des résultats** définit le type de la sortie de l'algorithme (une hiérarchie de clusters ou une partition de l'ensemble des exemples).
- **La complexité** représente le temps de calcul nécessaire à la résolution d'un problème. Il est un critère important à prendre en compte lors du choix de l'algorithme. En particulier, il est admis que la complexité algorithmique doit être linéaire en fonction du nombre d'exemples dans le cas des bases de données volumineuses.
- **Déterministe** définit la capacité des algorithmes à fournir les mêmes résultats (sans aucun changement) en utilisant les mêmes données en entrée.
- **Incrémental** définit la manière dont les données sont intégrées dans l'algorithme. Dans ce cas, les données sont intégrées au fur et à mesure de leur arrivée.
- **Prise en compte du contexte** définit la capacité de l'algorithme à prendre en compte ou non la problématique du contexte.

- **La tolérance au bruit** définit la capacité de l'algorithme à gérer ou non le bruit qui peut exister dans les données.
- **La tolérance aux clusters de tailles variées** définit la capacité de l'algorithme à détecter des clusters ayant des tailles différentes (Figure 2.12).
- **La tolérance aux clusters de densités variées** définit la capacité de l'algorithme à réaliser des clusters ayant des densités différentes (Figure 2.12).



**Figure 2.12.** Clusters de tailles et de densités différentes [40].

- **La tolérance aux clusters de formes quelconques** définit la capacité de l'algorithme à réaliser des clusters ayant des formes différentes (Figure 2.13).
- **La tolérance aux clusters concentriques** définit la capacité de l'algorithme à réaliser des clusters concentriques, c'est-à-dire, inscrits les uns dans les autres (Figure 2.13).

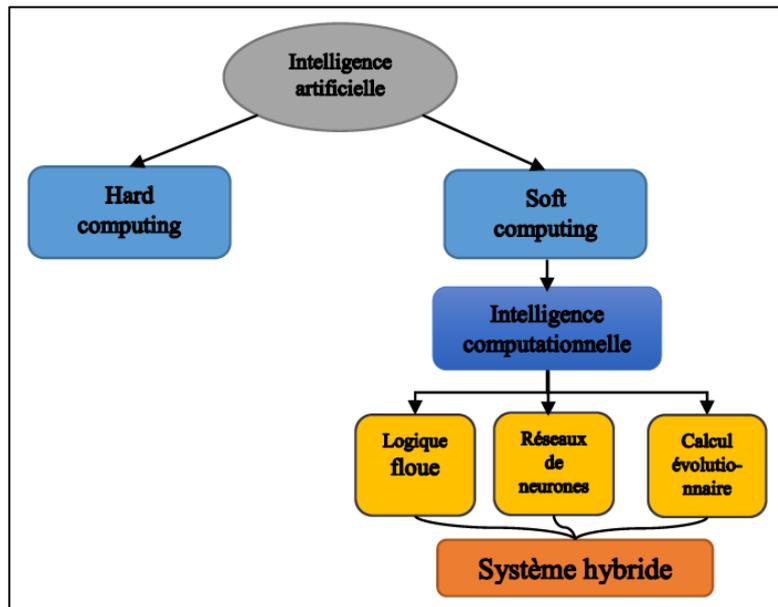


**Figure 2.13.** Clusters de formes variées et concentriques [40].

## 7. Vue générale sur l'intelligence computationnelle

L'intelligence computationnelle est associée au soft computing « *traduction française, l'informatique douce* », qui est caractérisé par l'utilisation de solutions inexactes pour des tâches difficiles d'un point de vue computationnel. Cela encourage l'intégration des techniques et outils du soft computing dans les applications quotidiennes et avancées. Ces outils sont utilisés indépendamment et conjointement en fonction du type du domaine d'applications. Elle, selon le professeur Zadeh, est « *une approche émergente à*

*l'informatique, qui est parallèle à la remarquable capacité de l'esprit humain à raisonner et à apprendre dans un environnement d'incertitude et d'imprécision* ». Le soft computing exploite la tolérance pour l'imprécision, l'incertitude et la vérité partielle pour atteindre la robustesse, la tractabilité, et un faible coût de la solution. Le terme intelligence computationnelle tel que défini par Zadeh, est une combinaison de l'informatique douce et le calcul numérique [41].



**Figure 2.14.** Position de l'intelligence computationnelle [41].

Jusqu'à maintenant, il n'y a pas de définition communément acceptée de l'intelligence computationnelle. Mais en général, l'intelligence computationnelle est un ensemble de méthodes de calculs inspirés de la nature et des approches pour résoudre les problèmes complexes du monde réel, dont la modélisation mathématique traditionnelle (classique) peut être inutile pour la raison que les processus pourraient être trop complexes et incertain. En effet, de nombreux problèmes de la vie ne peuvent pas être traduits en langage binaire (valeurs uniques de 0 et 1) pour les ordinateurs afin de les traiter. L'intelligence computationnelle fournit donc des solutions pour ces problèmes. Ses techniques sont en mesure de faire des prédictions ou des décisions concernant des événements futurs [41].

## 7.1. Historique

Lotfi Zadeh était celui qui a contribué le plus dans le domaine de la logique floue, et c'est le développeur des ensembles flous. A partir de là, jusqu'à ce que les systèmes flous des années 80 était un champ actif, produisant des noms tels que Mamdani, Sugeno, Takagi et Bezdek. Ensuite, les systèmes flous ont également connu un âge sombre dans les années 80, mais ont été relancés par des chercheurs japonais à la fin des années 80.

Aujourd'hui, il est un domaine très actif avec de nombreuses applications réussies, en particulier dans les systèmes de contrôle. En 1991, Pawlak introduit la théorie des ensembles approximatifs à l'informatique, où le concept fondamental est le constat d'une approximation inférieure et supérieure à l'espace d'entrée. Tous les éléments dans l'approximation inférieure ont une pleine adhésion, tandis que les éléments de frontière (ces éléments entre le rapprochement supérieur et inférieur) appartiennent à l'ensemble à un certain degré [41].

## 7.2. Domaines d'application de l'intelligence computationnelle

Chaque méthode composante de l'intelligence computationnelle a des domaines d'application pour lesquelles elle est particulièrement bien adaptée. Comme dans le contrôle, la prise de décision, l'analyse d'ADN...etc. Gardez à l'esprit que les domaines d'application peuvent se chevaucher; autrement dit, un problème de regroupement (clustering) donné peut être soit résoluble par un réseau de neurones ou d'un système flou par exemple, mais avec différents niveaux de performance. Notons que des combinaisons des méthodes peuvent produire des résultats différents [41].

## 8. Caractérisation des approches de l'intelligence computationnelle

Les techniques et approches de l'intelligence computationnelle englobent divers paradigmes dédiés à la résolution approximative des problèmes du monde réel dans la prise de décision, la classification des formes, et l'apprentissage. Ceux qui sont au premier rang de ces paradigmes et sont cités par la société d'Intelligence computationnelle « *Computational Intelligence Society* » (CIS) de l'IEEE établie en 2004, sont la logique floue, les réseaux de neurones, les algorithmes évolutionnaires, et les systèmes basés sur l'intelligence collective (Swarm Intelligence) comme faisant de l'intelligence computationnelle [41].

### 8.1. La logique floue

La logique floue est une extension de la logique classique qui permet la modélisation des imperfections des données et se rapproche dans une certaine mesure de la flexibilité du raisonnement humain. En utilisant le langage humain comme source d'inspiration, les systèmes flous (SF) modélisent l'imprécision linguistique et résolvent des problèmes incertains sur la base d'une généralisation de la logique traditionnelle, ce qui nous permet d'effectuer un raisonnement approximatif. La logique floue est une extension de la logique booléenne créée par Lotfi Zadeh en se basant sur sa théorie mathématique des ensembles flous, qui est une généralisation de la théorie des ensembles classiques. En introduisant la notion de degré dans la vérification d'une condition, permettant ainsi à une condition d'être dans un autre état que vrai ou faux, la logique floue confère une

flexibilité très appréciable aux raisonnements qui l'utilisent, ce qui rend possible la prise en compte des imprécisions et des incertitudes.

La logique floue peut incorporer des connaissances expérimentales humaines et lui donner un sens d'ingénierie pour modéliser et contrôler de tels systèmes mal définis avec non-linéarité et incertitude. La méthodologie de la logique floue traite généralement du raisonnement et de l'inférence à un niveau supérieur, tel que sémantique ou linguistique. Ce domaine de recherche comprend les ensembles et systèmes flous, la classification et la classification en flous, les contrôleurs flous, la synthèse linguistique, les réseaux de neurones flous, les ensembles et systèmes flous de type 2, etc. [35].

## 8.2. Réseaux de neurones artificiels

Le cerveau humain, qui possède une capacité extraordinaire d'apprentissage et de mémorisation, est un réseau dense de plus de 10 milliards de neurones, chacun connecté en moyenne à environ 10 000 autres neurones. Chaque neurone reçoit des signaux via des synapses, qui contrôlent les effets des signaux sur le neurone. Ces connexions synaptiques jouent un rôle important dans le comportement du cerveau. En utilisant le cerveau humain comme source d'inspiration, les Réseaux de Neurones Artificiels (RNA) sont des réseaux distribués massivement parallèles qui ont la capacité d'apprentissage et de généralisation à partir des exemples.

Les réseaux de neurones sont caractérisés par la puissance de calcul, la tolérance aux pannes, l'apprentissage à partir de données expérimentales et la capacité de généralisation. Ce sont essentiellement des algorithmes de calcul de bas niveau qui démontrent généralement de bonnes performances dans le traitement des données numériques. L'apprentissage se déroule sous différentes formes dans les réseaux de neurones, comme l'apprentissage supervisé, non supervisé, compétitif et de renforcement. La recherche sur les systèmes de contrôle basés sur des réseaux neuronaux a suscité un intérêt considérable au cours des dernières années, tout d'abord parce que les réseaux neuronaux se sont avérés capables d'approximer toute fonction non linéaire définie sur un ensemble compact de données avec une précision spécifiée, et deuxièmement, parce que la plupart des systèmes de contrôle présenter certains types de non-linéarité inconnue, ce qui convient aux réseaux de neurones en tant que technologie de contrôle appropriée [35].

## 8.3. Les algorithmes évolutionnaires

Les Algorithmes évolutionnaires (AEs) sont une famille d'algorithmes dont le principe est inspiré de la théorie de l'évolution pour résoudre des problèmes divers. Les AEs constituent une approche originale : il ne s'agit pas de trouver une solution analytique

exacte, ou une bonne approximation numérique, mais de trouver des solutions satisfaisant au mieux certains critères, souvent contradictoires. S'ils ne permettent pas de trouver à coup sûr la solution optimale de l'espace de recherche, du moins peut-on constater que les solutions fournies sont généralement meilleures que celles obtenues par des méthodes plus classiques, pour un même temps de calcul. Parmi les grandes classes d'algorithmes évolutionnaires on peut citer: les algorithmes génétiques et les stratégies évolutionnaires [35].

#### ❖ **Algorithme Evolutionnaire général**

1. Construction et évaluation d'une population initiale ;
2. Jusqu'à atteindre un critère d'arrêt :
  3. Sélection d'une partie de la population,
  4. Reproduction des individus sélectionnés,
  5. Mutation de la descendance,
  6. Evaluation du degré d'adaptation de chaque individu,
  7. Remplacement de la population actuelle par la nouvelle population.

### **8.4. Les systèmes basés sur l'intelligence collective (Swarm Intelligence)**

L'intelligence collective désigne les capacités cognitives d'une communauté résultant des interactions multiples entre les membres (ou agents) de la communauté. Des agents, au comportement très simple, peuvent ainsi accomplir des tâches complexes grâce à un mécanisme fondamental appelé synergie. Sous certaines conditions particulières, la synergie créée, par la collaboration entre individus, fait émerger des possibilités de représentation, de création et d'apprentissage supérieures à celles des individus isolés.

L'intelligence collective est observée notamment chez les insectes sociaux (fourmi, abeilles) et les animaux en mouvement (oiseaux migrateurs, bancs de poissons). En conséquence, plusieurs algorithmes basés sur le principe d'intelligence collective ont été introduits : on peut citer les algorithmes des abeilles artificielles et les essaims particuliers [35].

## **9. Implémentation et les synergies de l'intelligence computationnelle**

Ce qui est nécessaire est une unification des différentes méthodes de l'intelligence computationnelle, par conséquent, une approche combinée avec ses techniques et leur mise en œuvre est d'une importance pour une performance globale, et pour le coût du calcul et la commodité d'application. La combinaison synergique de ces méthodologies peut fournir de meilleurs modèles informatiques ou des algorithmes plus efficaces qui viendront compléter les limites de toute méthode unique individuelle. Cette combinaison

se fut appelée par de nombreux chercheurs, un système intelligent hybride. Zadeh (1994) pense que les systèmes intelligents hybrides sont certainement la voie de l'avenir [41].

Les différentes synergies des méthodes de l'intelligence computationnelle sont comme suite [41]:

- La logique floue avec les réseaux de neurones (système neuro-flou)
- Le calcul évolutionnaire avec les réseaux de neurones.
- Le calcul évolutionnaire avec la logique floue (synergisme évolutionnaire flou).
- L'hybride du calcul évolutionnaire, la logique floue et les réseaux de neurones (hybride de systèmes neuro-flou utilisant l'algorithme génétique).

Il ne nous ait pas caché qu'il existe d'autres combinaisons synergiques au-delà de celle-ci. Et comme nous pouvons le noter, que les hybrides des techniques ou méthodes de l'intelligence computationnelle sont généralement et quasiment construites sur une base de calculs évolutionnaires (algorithmes génétiques) vu que ces derniers ont la propriété d'optimisation et d'adaptation d'un comportement pour un système [41].

## 10. Application de l'Intelligence Computationnelle pour l'optimisation d'énergie dans les réseaux de capteur sans fils

Dans le domaine de la mise en réseau, les paradigmes de l'intelligence computationnelle ont été utilisés avec succès ces dernières années pour résoudre le problème de la consommation d'énergie dans les réseaux de capteurs sans fil. Outre l'algorithme LEACH, plusieurs autres algorithmes ont été proposés dans la littérature pour regrouper des nœuds de capteurs dans RCSF [35].

Dans [42], les auteurs ont proposé l'utilisation des AGs « *Algorithmes génétiques* » pour résoudre le problème de la sélection du meilleur nombre des CHs. La représentation chromosomique proposée était simplement une représentation binaire de 9 bits dans laquelle la valeur de bit égale à 1 représente un CH, et à 0 un nœud capteur (voir la figure 2.15). La fonction de fitness a été définie comme indiqué dans l'équation (2.4) :

<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>	<b>0</b>	<b>1</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------

**Figure 2.15.** 9-bit Représentation binaire des chromosomes [35].

$$\text{Fitness} = w * (D - \text{distance}_i) + (1 - w) * (N - H_i) \quad (2.4)$$

Où  $D$  est la distance totale de tous les nœuds à la station de base; la  $\text{distance}_i$  est la somme des distances des nœuds membres aux CHs et la somme des distances de tous les CH à la station de base;  $H$  nombre de CHs;  $N$  est le nombre total de nœuds; et  $w$  est un poids prédéfini. Les résultats ont montré que la disposition des clusters dépend de

l'emplacement de la station de base (récepteur). Un plus grand nombre CHs est élu lorsque la station de base est située au centre du réseau. Cependant, l'énergie résiduelle n'a pas été prise en compte dans la fonction de Fitness.

Les auteurs dans [43] ont proposé une méthode de calcul de la fonction de fitness tenant compte la différence d'énergie entre les chromosomes de la génération actuelle et de la génération précédente. Les chromosomes à faible différence sont sélectionnés. Dans leur travail, l'énergie est le seul critère pris en compte dans la fonction fitness.

Les auteurs dans [44] ont amélioré le travail précédent dans [33] en ajoutant l'énergie résiduelle au calcul de la fonction de fitness et en conservant la même représentation chromosomique. Les résultats ont été comparés avec LEACH et ont montré une répartition correcte des Clusters et une amélioration significative de la durée de vie du réseau.

Dans [45], les auteurs ont présenté un CH qui transmet les informations recueillies par les CHs et agrège / compresse les données avant de les transmettre à la station de base. Cependant, cette responsabilité supplémentaire du nœud se traduit par une plus grande perte d'énergie entraînant une dégradation inégale du réseau. Dans ce travail, un algorithme Firefly synchrone modifié est proposé pour améliorer les performances du réseau. Une simulation a montré que la technique proposée donne de bons résultats par rapport à LEACH.

Dans [46], les auteurs ont proposé une technique de sélection de CH basée sur des PSO « *Particle Swarm Optimization* » avec une fonction multi-objective prenant en compte la consommation d'énergie des CHs et les retards dans la transmission des paquets de données. Dans le système proposé, la dimension de chaque particule correspond au numéro ou à l'identifiant d'un nœud de capteur dans le réseau.

Néanmoins, aucune méthode ne s'est révélée être la meilleure dans tous les cas. Par conséquent, l'idée d'adopter une autre représentation chromosomique et d'ajouter une recherche exhaustive lorsque le nombre de nœuds de capteurs diminue semble offrir un potentiel considérable pour améliorer les capacités du processus de recherche et trouver la meilleure combinaison de clusters.

## 11. Travaux connexes

Il existe de nombreux travaux portant sur les techniques de clustering qui ont déjà été proposées pour fournir des solutions aux problèmes majeurs du regroupement ; surtout où l'ensemble des données à clustériser (l'ensemble des capteurs dans notre cas d'étude) n'est pas totalement connu dès le départ comme en clustering classique.

Les travaux les plus proches de notre problématique concernent des nouvelles techniques ou bien des nouveaux protocoles qui sont proposés spécialement pour éviter les problèmes du clustering classique et gagner le défi de la conservation d'énergie et le prolongement de la durée de vie du réseau de capteurs sans fil.

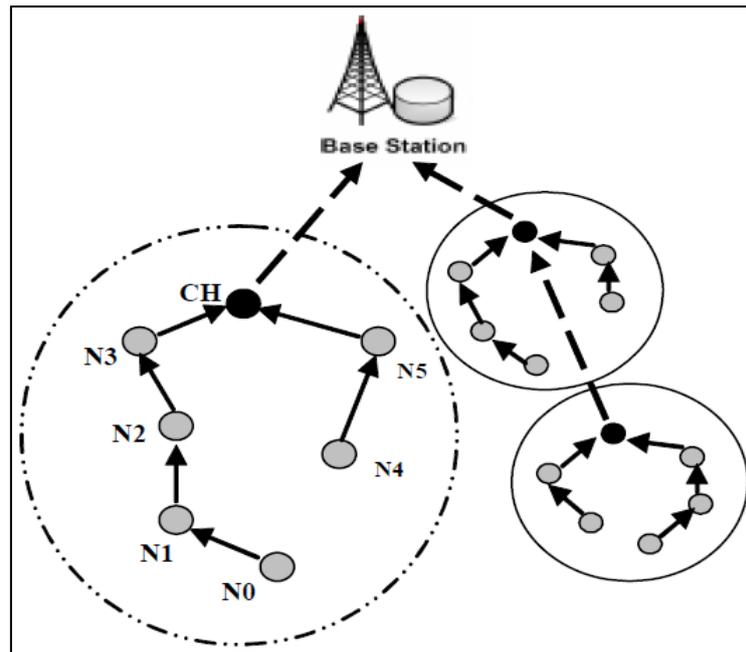
### 11.1. Travaux au niveau de routage :

#### 11.1.1. HEEP (Hybrid Energy Efficiency Protocol)

Dans [47], les auteurs ont proposé un nouveau protocole de routage appelé HEEP qui combine deux algorithmes: LEACH et PEGASIS. HEEP conseille une nouvelle auto-organisation du réseau qui rejoint les approches basées sur les clusters et les chaînes. Cette nouvelle approche est appelée approche de regroupement des chaînes (*chain clustering approach*), elle organise les nœuds de réseau en grappes de chaînes pour éviter les mauvaises dissipations d'énergie dans le protocole LEACH et réduit le retard de routage généré par le protocole PEGASIS. Les résultats expérimentaux dans [47] montrent que HEEP surpasse LEACH et LEACH-C avec plus de 110% de la durée de vie du réseau et réduit de 50% le délai de routage introduit dans PEGASIS.

Le protocole HEEP proposé combine les avantages des deux approches (approche basée sur les clusters) et (approche basée sur les chaînes). Comme LEACH, leur solution est basée sur une agrégation des clusters des capteurs. De plus, HEEP applique le principe de PEGASIS à l'intérieur des clusters où une chaîne de nœuds dans un même cluster est construite pour améliorer la dissipation d'énergie, donc les nœuds communiquent uniquement avec leurs voisins les plus proches et non directement avec le CH. Chaque CH envoie les données collectées au sein du cluster vers la BS via les voisins CH, en utilisant les techniques multi-sauts, qui limite la dépense énergétique. L'agrégation des données de chaque nœud dans une chaîne réduit la quantité d'échange de données entre les nœuds et leur CH, pour préserver une réserve d'énergie des nœuds.

La figure 2.16 montre l'organisation des nœuds au sein des clusters. N0 transmet ses données à son voisin le plus proche N1; N1 fusionne les données avec le sien et les envoie à un autre voisin, jusqu'à atteindre le CH qui les transmet à la BS. Par conséquent, dans cette nouvelle organisation (chains clustering), les distances de transmission sont abrégées et le nombre de nœuds communiquant avec le CH est considérablement réduit. Cela implique une meilleure minimisation de la consommation d'énergie et prolonge la durée de vie des CH. Concernant la latence introduit par la longue chaîne de nœuds dans le protocole PEGASIS, il est considérablement réduit car les chaînes formées sur les clusters sont de petite taille et fonctionnent en même temps.



**Figure 2.16.** L'organisation des nœuds avec l'approche chains clustering [47].

Pour accéder au support, ils utilisent la méthode proposée dans LEACH, où les CH établissent un plan de transmission (TDMA), dans lequel ils définissent l'heure exacte de chaque nœud où il peut transmettre ses données collectées. L'utilisation du programme TDMA évite les problèmes de collision et d'interférence à l'intérieur du même cluster. HEEP utilise le concept de rotation aléatoire des rôles du CH proposés par LEACH, qui contrôle la dissipation d'énergie et empêche que les nœuds choisis comme CHs ne meurent pas rapidement.

HEEP a deux phases essentielles: la phase d'initialisation qui forme des clusters enchaînés, et élit les CH, et la phase de transmission où les données collectées sont transmises.

A chaque phase de transmission, tous les nœuds appartenant au même cluster est délégué aux données de détection (tâche moins de ressources), où la tâche de ressources intensives (données de détection de transmission à la BS) est attribué au plus puissant nœud qui est le CH. Cela signifie que HEEP délègue la tâche de ressources intensive à un nœud(CH) à chaque phase de transmission et attribue la tâche de détection aux autres nœuds, même s'il existe des nœuds puissants dans le cluster. La consommation d'énergie peut donc ne pas augmenter s'il y a plusieurs nœuds puissants.

Le coût énergétique de la phase d'initialisation est très faible par rapport à la phase de transmission, et dépend du nombre de nœuds dans le réseau et la distance de la BS. Selon le réseau, les auteurs de [47] l'estiment par la règle suivante:

$$\sum_{i=1}^{i=N} (q_i E_{\text{elect}} + q_i E_{\text{fs}} d_{\text{toBS}}^2) \quad (2.5)$$

Où  $q_i$  est la taille du package de contrôle transmis par le nœud  $i$ ,  $E_{\text{elect}}$  est la puissance consommée par les circuits électronique (énergie de calcul),  $E_{\text{fs}}$  est l'énergie perdue dans l'espace de transmission,  $d_{\text{toBS}}$  est la distance entre le nœud  $i$  et la BS, et  $N$  c'est le nombre des nœuds vivants dans le réseau étudié. Ils calculent également l'énergie initiale totale des réseaux par l'équation (2.6):

$$E_{\text{total}} = NE_o \quad (2.6)$$

Où  $E_o$  est l'énergie initiale de chaque nœud.

La durée de vie du réseau est divisée en plusieurs cycles de la communication. Où le total approximatif des tours peut être estimé par:

$$T_{\text{round}} = E_{\text{total}} / E_{\text{round}} \quad (2.7)$$

$E_{\text{round}}$  est l'énergie totale dissipée dans le réseau pendant une rond, qui est égal à:

$$E_{\text{round}} = q \left[ 2NE_{\text{elect}} + NE_{\text{DA}} + NE_{\text{fs}} d_{\text{tonextnode}}^2 + E_{\text{mp}} d_{\text{CHtoBS}}^4 \right] \quad (2.8)$$

Où  $E_{\text{DA}}$  c'est le coût d'agrégation des données dépensé dans chaque nœud,  $d_{\text{CHtoBS}}$  est la distance moyenne entre le CH et la BS,  $d_{\text{tonextnode}}$  est la distance moyenne au nœud suivant dans la chaîne,  $q$  est la taille totale des données transmises,  $E_{\text{fs}}$  et  $E_{\text{mp}}$  dépendent du modèle d'amplificateur de l'émetteur utilisé.

À travers [47], les auteurs ont tenté de suggérer un nouvel algorithme qui évite les inconvénients des deux protocoles LEACH et PEGASIS tout en combinant leurs avantages. Leurs résultats expérimentaux montrent que HEEP surpasse efficacement ces deux protocoles.

### 11.1.2. CCBRP (Chain-Chain Based Routing Protocol)

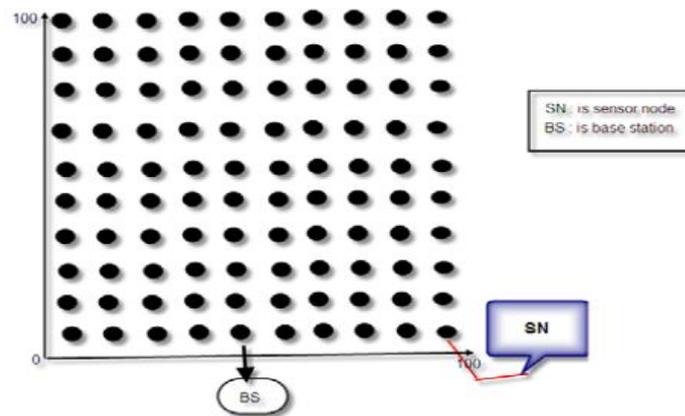
Dans [48], les auteurs ont proposé un protocole de routage efficace appelé CCBRP, il atteint à la fois une consommation d'énergie minimale et un retard minimum. Ce protocole divise principalement un RCSF en plusieurs chaînes (l'algorithme Greedy est utilisé pour former chaque chaîne comme dans le protocole PEGSIS) et fonctionne en deux phases. Dans la première phase, les nœuds capteurs de chaque chaîne transmettent des données à leurs nœuds leaders de la chaîne en parallèle. Dans la deuxième phase, tous les nœuds leaders de la chaîne forment une chaîne (également en utilisant l'algorithme Greedy), puis tous les nœuds leaders envoient leurs données à un nœud leader choisi au hasard, le nœud leader choisi fusionne ses données avec les

données reçues et les envoie à la BS. Les résultats expérimentaux présentés dans [48] démontrent que le CCBRP proposé surpasse LEACH, PEGASIS et CCM en ce qui concerne le produit de l'énergie consommée et de retard.

### 11.1.2.1. Fonctionnement de CCBRP :

CCBRP divise le RCSF en un nombre de chaînes; L'algorithme Greedy est utilisé pour construire chacune des chaînes comme dans PEGASIS. Chaque chaîne contient un certain nombre de nœuds de capteur, le nombre des chaînes et le nombre de nœuds de capteur dans chaque chaîne dépendent du nombre de nœuds de capteur dans le RCSF à l'étude.

Pour illustrer le CCBRP proposé, les auteurs de [48] considèrent un RCSF avec N nœuds capteurs répartis en zone de 2 dimensions ayant une taille de L (m) × L (m). Si N est égal à cent nœuds et ils supposent que le nombre de chaînes est égal à dix pour cent de N, alors il y a dix chaînes contenant chacune dix nœuds capteurs comme le montre la figure 2.17

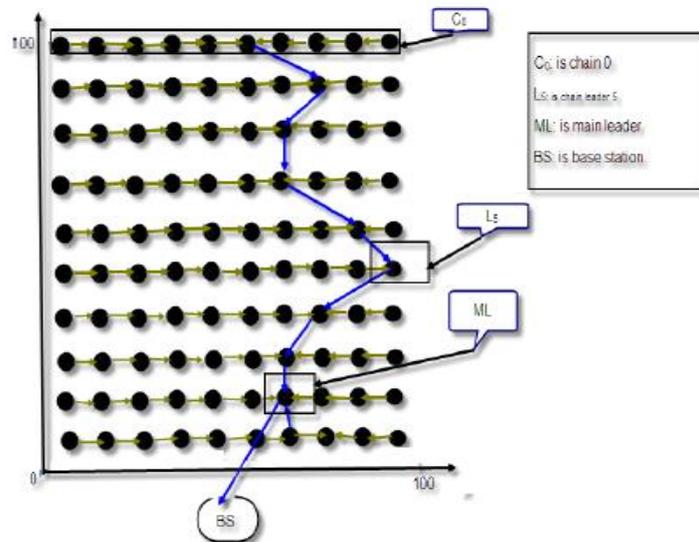


**Figure 2.17.** 100 nœuds capteurs divisés en 10 chaînes [48].

Le CCBRP proposé forme chacune des partitions des chaînes en utilisant l'algorithme Greedy et s'exécute en deux phases. La première phase commence par la sélection aléatoire d'un leader pour chaque chaîne, puis chaque leader envoie un message jeton aux deux extrémités de sa chaîne pour les avertir. Après cela, chacun des deux nœuds d'extrémité de chaque chaîne commence simultanément à envoyer ses données à son placard nœud voisin, les nœuds voisins reçoivent les données et fusionnent ses données avec les données reçues et les envoient au nœud suivant de la chaîne, etc. Ce processus est répété jusqu'à ce que les données aient atteint tous les nœuds leaders de la chaîne. L'algorithme 2.1 présente un pseudo-code pour la première phase du CCBRP, l'algorithme 2.2 présente un pseudo-code pour la deuxième phase du CCBRP proposé, et la Figure 2.18 illustre la transmission des données pour le CCBRP.

<b>Algorithme 2.1. Pseudo-code pour la 1<sup>ere</sup> phase de CCBRP [48].</b>	<b>Algorithme 2.2. Pseudo-code pour la 2<sup>eme</sup> phase de CCBRP [48].</b>
<p>Partition the <math>N</math> nodes of the given WNS into <math>n</math> chains (<math>C_0</math> to <math>C_{n-1}</math>), and let <math>y</math> (<math>y=N/n</math>) be the number of nodes in each chain <math>C_i</math></p> <p>For <math>i=0</math> to <math>n-1</math> do  {Select a chain leader <math>L_i</math> randomly  Case <math>CL_i</math>:  <math>C_i(1)</math>: the first node of <math>C_i</math>  { <math>CL_i</math> sends a token message to <math>C_i(y)</math> the end of this chain  <math>C_i(y)</math> sends its data (<math>i, y</math>) to <math>C_i(y-1)</math>;  While <math>y &gt; j</math>  <math>C_i(y-1)</math> fuses its own data (<math>i, y-1</math>) and received data (<math>i, y</math>)  <math>C_i(y-1)</math> sends the fused data to its neighbor <math>C_i(y-2)</math>;  <math>y=y-1</math>;  <math>C_i(N/n)</math>: the last node of <math>C_i</math>  { <math>x=1</math>;  <math>CL_i</math> sends a token message to <math>C_i(x)</math>  <math>C_i(x)</math> sends its data (<math>i, x</math>) to <math>C_i(x+1)</math>;  While <math>x &lt; j</math>  <math>C_i(x+1)</math> fuses its own data (<math>i, x+1</math>) and received data (<math>i, x</math>)  <math>C_i(x+1)</math> transmits the fused data to its neighbor <math>C_i(x+2)</math>;  <math>x=x+1</math>;  <math>C_i(2.. N/n - 1)</math>: all the intermediate nodes of <math>C_i</math>  { <math>x=1, y=N/n</math>;  <math>CL_i</math> sends a token message to <math>C_i(x)</math>;  If (<math>x &lt; j</math>) then  { <math>C_i(x)</math> sends its data (<math>i, x</math>) to <math>C_i(x+1)</math>;  While <math>x &lt; j</math>  <math>C_i(x+1)</math> fuses its own data (<math>i, x+1</math>) and received data (<math>i, x</math>)</p>	<p>Constitute a new chain from the chain leaders <math>L_i</math> (<math>i=0..n-1</math>) using Greedy algorithm.</p> <p>Randomly select the main leader <math>ML</math>.  Let <math>x=1</math> and <math>y=n</math>, <math>j</math> index of <math>ML</math> in <math>CLs</math>;  Case <math>ML</math>:  <math>CLs(0)</math>; the first node of <math>CLs</math>  { <math>ML</math> sends a token message to <math>CLs(y)</math> end of chain  <math>CLs(y)</math> sends its fused_data(<math>y</math>) to <math>CLs(y-1)</math>;  While <math>y &gt; j</math>  <math>CLs(y-1)</math> fuses its own fused_data(<math>y-1</math>) and received fused_data(<math>y</math>)  <math>CLs(y-1)</math> sends the fused data to its neighbor <math>CLs(y-2)</math>;  <math>y=y-1</math>;  <math>CLs(n-1)</math>; the last element of <math>CLs</math>  { Let <math>x=1</math>;  <math>ML</math> sends a token message to <math>CLs(x)</math>  <math>CLs(x)</math> sends its fused_data(<math>x</math>) to <math>CLs(x+1)</math>;  While <math>x &lt; j</math>  <math>CLs(x+1)</math> fuses its own fused_data(<math>x+1</math>) and received fused_data(<math>x</math>)  <math>CLs(x+1)</math> sends the fused data to its neighbor <math>CLs(x+2)</math>;  <math>x=x+1</math> ;  <math>CLs(1.. n)</math>; intermediate node in <math>CLs</math>  { Let <math>x=1, y=n</math>;  <math>ML</math> sends a token message to <math>CLs(x)</math>;  <math>CLs(x)</math> sends its fused_data(<math>x</math>) to <math>CLs(x+1)</math>;  If (<math>x &lt; j</math>)  { While <math>x &lt; j</math>  <math>CLs(x+1)</math> fuses its own fused_data(<math>x+1</math>) and the received fused_data(<math>x</math>)  <math>CLs(x+1)</math> sends the fused data to its neighbor <math>CLs(x+2)</math>;</p>

<pre> <i>Ci(x+1) transmits the fused data to its neighbor</i> <i>Ci(x+2);</i> <i>x=x+1; }</i> <i>CLi sends a token message to Ci(y);</i> <i>If(y&gt;j)</i> <i>{Ci(y) sends its data (i ,y) to Ci(y-1);</i> <i>While y &gt;j</i> <i>Ci(y-1) fuses its own data (i, y-1) and</i> <i>received data (i , y);</i> <i>Ci(y-1) transmits the fused data to its</i> <i>neighbor Ci(y-2);</i> <i>y=y-1 ;}</i> <i>}</i> <i>CLi fuses its data (i ,j) with the data</i> <i>received from Ci(j+1) and data received from</i> <i>Ci(j-1);</i> </pre>	<pre> <i>x=x+1;</i> <i>}</i> <i>Li sends a token message to CLs (y)</i> <i>If(y&gt;j)</i> <i>{</i> <i>CLs (y) sends its fused_ data(y) to</i> <i>CLs (y-1);</i> <i>While y &gt;j</i> <i>CLs (y-1) fuses its own fused_</i> <i>data(y-1) and received fused_ data(y) ;</i> <i>CLs (y-1) sends the fused data to its</i> <i>neighboring s CLs (y-2);</i> <i>y=y-1;}</i> <i>}</i> <i>ML fuses its data with data from CLs(j+1)</i> <i>and CLs(j-1)</i> <i>and sends it to the BS;</i> </pre>
---	---



**Figure 2.18.** La transmission des données pour CCBRP [48].

Dans [48], Le CCBRP proposé atteint à la fois la consommation d'énergie minimale et un retard minimum expérimenté. Les résultats expérimentaux montrent que la consommation d'énergie de ce protocole est presque la même que pour PEGASIS, 60% moins que LEACH et 10% moins que CCM pour un RCSF avec cent nœuds. Le retard du CCBRP est le même que de LEACH et CCM mais 75% moins que de PEGASIS, donc il surpasse LEACH, PEGASIS et CCM en ce qui concerne le produit de l'énergie consommée et le retard. Ainsi que ces résultats montrent que la taille du RCSF augmente la différence entre la consommation d'énergie du CCBRP proposé et de la

consommation d'énergie de PEGASIS. Cependant, l'inverse est vrai pour le protocole CCM; la différence entre la consommation d'énergie du CCM et la consommation d'énergie de PEGASIS augmente avec l'augmentation de la taille du RCSF.

### 11.1.3. Comparaison des différents protocoles de routage de clustering dans les RCSFs :

Dans cette section, nous comparons les différents algorithmes de routage de clustering pour les RCSFs, Nous résumons les différences entre les protocoles de routage de clustering dans les RCSFs selon quelques mesures importantes comme indiqué dans le tableau 2.3.

<b>Nom du protocole</b>	<b>Efficacité énergétique</b>	<b>Stabilité de cluster</b>	<b>Evolutivité</b>	<b>Délai de livraison</b>	<b>Equilibrage de charge</b>	<b>Complexité l'algorithme</b>
<i>LEACH</i>	<i>très lent</i>	<i>modéré</i>	<i>très lent</i>	<i>très petit</i>	<i>modéré</i>	<i>lent</i>
<i>HEED</i>	<i>Modéré</i>	<i>haut</i>	<i>modéré</i>	<i>modéré</i>	<i>modéré</i>	<i>modéré</i>
<i>DWEHC</i>	<i>très haut</i>	<i>haut</i>	<i>modéré</i>	<i>modéré</i>	<i>très bon</i>	<i>modéré</i>
<i>PANEL</i>	<i>Modéré</i>	<i>lent</i>	<i>lent</i>	<i>modéré</i>	<i>bon</i>	<i>haut</i>
<i>TL-LEACH</i>	<i>Lent</i>	<i>modéré</i>	<i>modéré</i>	<i>petit</i>	<i>mal</i>	<i>lent</i>
<i>UCS</i>	<i>très lent</i>	<i>haut</i>	<i>Lent</i>	<i>petit</i>	<i>mal</i>	<i>modéré</i>
<i>EECS</i>	<i>Modéré</i>	<i>haut</i>	<i>lent</i>	<i>petit</i>	<i>modéré</i>	<i>très haut</i>
<i>EEUC</i>	<i>Haut</i>	<i>haut</i>	<i>haut</i>	<i>modéré</i>	<i>bon</i>	<i>haut</i>
<i>ACE</i>	<i>Modéré</i>	<i>très lent</i>	<i>modéré</i>	<i>petit</i>	<i>modéré</i>	<i>très haut</i>
<i>BCDCP</i>	<i>très lent</i>	<i>haut</i>	<i>très lent</i>	<i>petit</i>	<i>bon</i>	<i>très haut</i>
<i>PEGASIS</i>	<i>Lent</i>	<i>lent</i>	<i>très lent</i>	<i>très grand</i>	<i>modéré</i>	<i>haut</i>
<i>TEEN</i>	<i>très haut</i>	<i>haut</i>	<i>lent</i>	<i>petit</i>	<i>bon</i>	<i>haut</i>
<i>APTEEN</i>	<i>Modéré</i>	<i>très lent</i>	<i>lent</i>	<i>petit</i>	<i>modéré</i>	<i>très haut</i>
<i>TTDD</i>	<i>très lent</i>	<i>très haut</i>	<i>lent</i>	<i>très grand</i>	<i>bon</i>	<i>lent</i>
<i>CCS</i>	<i>Lent</i>	<i>lent</i>	<i>lent</i>	<i>grand</i>	<i>très mal</i>	<i>modéré</i>
<i>HGMR</i>	<i>Lent</i>	<i>Haut</i>	<i>très haut</i>	<i>modéré</i>	<i>mal</i>	<i>lent</i>

**Tableau 2.3.** Comparaison des différents protocoles de routages de clustering dans les RCSFs [49].

## 11.2. Travaux au niveau de construction des clusters

### 11.2.1. Approche basée sur les colonies d'Abeilles Artificielles [35]

L'algorithme ABC (*Artificial Bee Colony*) est un algorithme basé sur l'intelligence en essaim, qui simule le comportement de recherche de nourriture des colonies d'abeilles mellifères. L'approche proposée dans [35] basée sur les colonies d'abeilles artificielles peut être illustré par la figure 2.19.

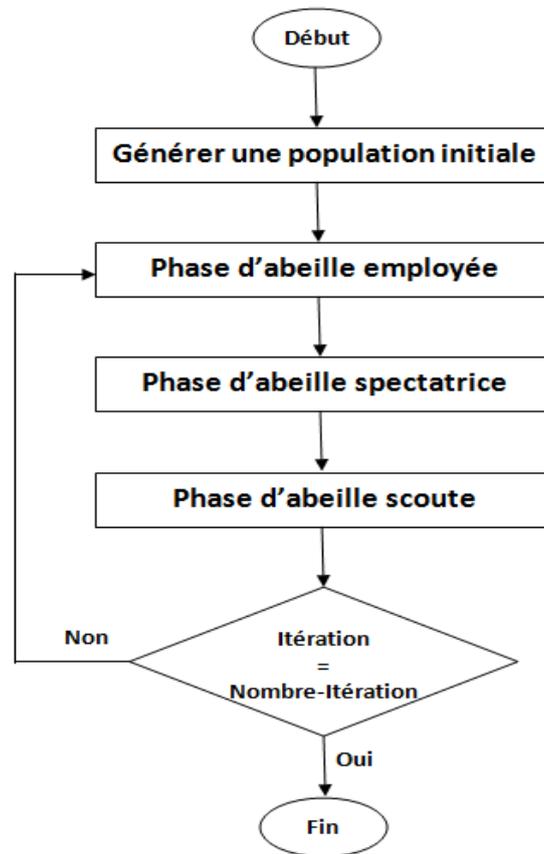


Figure 2.19. Les phases de l'algorithme ABC [35].

- **Initialisation de la population**

La génération de la population initiale de l'approche proposée dans [35] basée sur l'algorithme ABC passe par deux étapes. La première étape consiste à déterminer les CHs de chaque individu. Pour qu'un individu soit un CH, une sélection aléatoire est appliquée pour désigner tel individu est un CH. Le pourcentage des CHs par rapport au nombre total de nœuds dans un individu est fixé à 20 %. La deuxième étape consiste à affecter les nœuds réguliers au CH le plus proche.

- **Phase d'abeille employée**

Afin d'adapter l'algorithme ABC au problème de Clustering d'un RCSF, l'auteur de [35] définit une nouvelle équation pour mettre à jour la solution de chaque abeille artificielle. Cette équation est exprimée comme suit:

$$V_i = X_i \oplus (X_i \ominus X_k) \quad (2.9)$$

Où :  $X_i$  est une source de nourriture actuel et  $X_k$  est une autre source de nourriture choisie aléatoirement. L'équation (2.9) est utilisée pour créer une nouvelle solution  $V_i$  basée sur la solution actuelle  $X_i$ . Dans l'équation (équation 2.9), les deux opérations  $\ominus$  et  $\oplus$  ont la signification suivante:

- Operation  $\ominus$

Pour chaque ressource de nourriture  $X_i$  (solution potentielle), il sélectionne  $X_k$  une autre ressource de nourriture de manière aléatoire et compare l'identificateur du CH sur chaque nœud en utilisant l'énergie requise pour l'envoi d'un message à partir du nœud et du CH correspondant.

- Operation  $\oplus$

Il applique cette opération à la ressource de nourriture courant  $X_i$  et au résultat de l'opération  $\ominus$ . Pour effectuer cette opération, il compare l'identifiant du CH sur chaque nœud en utilisant la distance entre le nœud et le CH associé.

Dans la phase d'abeille employée il applique les étapes suivantes :

- Evaluer l'ensemble des solutions de la population en appliquant la fonction de fitness (2.10).

$$\text{Fitness} = \text{RE} + \text{SE} + w * (\text{D-distance}_i) + (1-w) * (\text{N-H}) \quad (2.10)$$

$$0 \leq w \leq 1$$

Où

- **D** : représente la distance totale de tous les nœuds à la station de base;
  - **distance<sub>i</sub>** : représente la somme des distances des nœuds membres aux CHs et la somme des distances de tous les CHs à la station de base;
  - **H** : représente le nombre de CHs;
  - **N** : représente le nombre total de nœuds;
  - **w** : représente un poids prédéfini.
  - **RE** : représente la somme des énergies résiduelles dans les CHs,
  - **SE** : représente la somme des énergies requises pour l'envoi d'informations depuis les CHs vers la station de base.
- Sélectionnez une solution  $X_k$  aléatoirement ( $k \neq i$ )
  - Générer une nouvelle solution  $V_i$  par l'équation (2.9) utilisant  $X_i$  et  $X_k$
  - La nouvelle solution générée par l'équation (2.9) est évaluée en utilisant la fonction de fitness (2.10), si le résultat correspondant à la nouvelle solution  $V_i$  est supérieur au résultat de  $X_i$ , alors la nouvelle solution  $V_i$  sera choisie sinon on garde la solution  $X_i$ .
- **Phase d'abeille spectatrice**
- Dans la phase d'abeille spectatrice, nous allons appliquer les étapes suivantes :
- Evaluer l'ensemble des solutions de la nouvelle population (résultat de la phase employée).

- Faire correspondre à chaque valeur de fitness un pourcentage  $P_i$  en appliquant l'équation (2.11)

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (2.11)$$

Où  $fit_i$  est la valeur de fitness de la solution  $i$  qui est proportionnelle à la quantité de nectar de la source de nourriture dans la position  $i$  et  $SN$  est le nombre de sources alimentaires qui est égal au nombre d'abeilles employées.

- Sélectionner une solution  $X_i$  avec un algorithme de sélection par roulette en utilisant le  $P_i$  calculé précédemment comme paramètre, pour chaque abeille spectatrice correspond une solution sélectionnée de cette manière.
- Sélectionner une solution  $X_k$  aléatoirement ( $k \neq i$ )
- Générer une nouvelle solution  $V_i$  par l'équation (2.9) utilisant  $X_i$  et  $X_k$
- La nouvelle solution générée par l'équation (2.9) est évaluée en utilisant la fonction de fitness (2.10), si le résultat correspondant à la nouvelle solution  $V_i$  est supérieur au résultat de  $X_i$ , alors la nouvelle solution  $V_i$  sera choisie sinon on garde la solution  $X_i$ .
- Incrémenter la valeur  $Compteur_i$  si la  $fitness(X_i) > fitness(V_i)$

---

### **Algorithme 2.3. Algorithme basé sur ABC pour le Clustering d'un RCSF**

---

*Entrée: Initialisation*

- $Psize=30$  % Taille de population %
- Population  $P$  % 30 Particule%
- $NombreIteration = 30$  ; % Nombre d'itération pour chaque cycle%
- $Itération = 1$  ;

*Début:*

- Répéter

% phase d'abeille employée %

Pour  $i=1$  à  $Psize$  Faire

- Sélectionnez  $k$  aléatoirement %  $k \in \{1, 2, \dots, n\}$  et  $k \neq i$  %
- Mettre à jour la solution de l'abeille artificielle  $i$  en utilisant l'équation 2.9
- Calculer la fitness de  $V_i$  et  $X_i$
- Si  $fitness(V_i) > fitness(X_i)$  Alors

$X_i = V_i$  ;

- Finsi

Finpour

% phase d'abeille spectatrice %

Pour  $i=1$  à  $Psize$  Faire

- Calculer  $fitness(X_i)$
- Calculé la probabilité de sélection en utilisant l'équation 2.11

Finpour

Pour  $i=1$  à  $Psize$  Faire

- Sélection  $X_i$  % Sélection par la roulette %
  - Sélectionnez  $k$  aléatoirement %  $k \in \{1, 2, \dots, n\}$  et  $k \neq i$  %
  - Calculer  $V_i$  en utilisant l'équation 2.9
  - Calculer la fitness de  $V_i$  et  $X_i$
  - Si  $fitness(V_i) > fitness(X_i)$  Alors
-

---

```

Xi = Vi ;
- Sinon
Conteuri = Conteuri + 1 ;
- Finsi
Finpour
% phase d'abeille scoute %
Pour i=1 à Psize Faire
- Si Conteuri > Trail Alors
Xi = Nouvelle Xi % création d'une solution aléatoire %
- Finsi
Finpour
Itération = Itération + 1 ;
• Jusqu'à Itération = NombreIteration
• Solution := la meilleure Abeille de P
Fin.

```

---

- **Phase d'abeille scoute**

Cette étape est exécutée seulement si la même solution (une configuration) est sélectionnée (dans la phase d'abeille spectatrice) un nombre de fois *Trail*. Dans cette situation, la solution sélectionnée est abandonnée et elle est remplacée par une nouvelle solution générée aléatoirement.

L'algorithme ABC (avec ces différentes phases) est répété *NombreIteration* fois. À la fin de ces itérations nous choisissons la meilleure solution (la meilleur solution de la population dont la consommation d'énergie est la plus faible) pour un cycle.

### 11.2.2. Approche basée sur les algorithmes génétiques

Les auteurs de [50] ont utilisé une méthode efficace basée sur une algorithme génétique pour résoudre le problème d'optimisation d'un RCSF, alors que chaque chromosome est divisé en un nombre spécifique des gènes en fonction de la taille du réseau. Chaque gène du chromosome a la valeur 1 ou zéro, où 1 ou 0 indique que ce nœud est respectivement un cluster-head ou un nœud régulier. La fonction de fitness utilisée dans leur travail dépend de deux facteurs principaux, notamment *Distance<sub>i</sub>*, qui est la somme des distances des nœuds réguliers aux CHs et de tous les CHs au nœud sink, et *H<sub>i</sub>*, qui indique le nombre des CHs disponibles dans le réseau. L'AG devrait trouver le chromosome qui a le plus petit nombre de CHs et la plus petite *Distance<sub>i</sub>* comme décrit dans la fonction de fitness suivante:

$$\mathit{Fitness} = W * ( D - \mathit{Distance}_i ) + ( 1 - W ) * ( N - H_i ) \quad (2.12)$$

Où *D* est la somme des distances de tous les nœuds au sink, *N* est le nombre total de nœuds et *W* est un nombre poids prédéfini, ( $0 \leq W \leq 1$ ), utilisé pour déterminer l'intérêt de l'algorithme. Par exemple, *W = 0* signifie que le nombre de CHs dans l'algorithme est pris en compte lorsque *W = 1* signifie que seule la distance de communication est prise en compte. AG commence par une sélection aléatoire de nœuds dans le réseau comme des CHs ; en utilisant la fonction de fitness, l'algorithme ajuste ces CHs.

L'approche proposée dans [50] est une méthode efficace et efficiente pour résoudre le problème de communication de longues distances dans lequel il est capable de trouver une bonne solution pour les topologies des réseaux uniformes et non uniformes. Leurs résultats montrent que le nombre de CHs représente environ 10% du nombre total de nœuds. Ils ont également réalisé une réduction d'environ 80% des communications de longue distance par rapport à l'approche de transmission directe. Cette approche est considérée une référence forte à ce mémoire.

## Conclusion

Ce chapitre avait pour but de présenter une vue générale du problème du clustering en introduisant les notions de base en classification non supervisée, et de mettre en évidence les étapes de base de la procédure de clustering, ses différentes applications possibles et quelques approches de ce type de classification. Nous avons présenté aussi à travers ce chapitre le domaine de l'intelligence computationnelle qui englobe des techniques efficaces à la résolution d'une grande gamme de problèmes complexes d'optimisation. A la fin du chapitre, nous avons cité quelques travaux connexes des différents protocoles de routages dans les RCSFs et des algorithmes de l'intelligence computationnelle qui ont été utilisé pour concevoir des protocoles de routage dans ce type de réseaux.

Le chapitre suivant est consacré à la présentation de notre approche qu'on va la proposé pour traiter le problème de clustering dans les RCSFs.

## Chapitre 3

# Contribution : l'approche proposée pour le traitement du problème de clustering dans les RCSFs

*"La science est un trésor dont l'usage fait le prix : chaque fois que vous instruisez,  
celui qui vous interroge, vous augmentez votre science."*

***Mocharrafoddin SAADI***

---

### Introduction

Un RCSF est un ensemble de nœuds de capteurs sans fil formant un réseau temporaire sans l'aide d'une infrastructure établie ou d'une administration centralisée. Cependant, l'énergie est une ressource limitée dans ces réseaux. De plus, le déploiement de tels réseaux dans un environnement hostile, où il n'y a pas d'accès possible, a posé un problème fondamental, à savoir l'alimentation énergétique des capteurs. Initialement équipé d'une pile d'énergie, un capteur sans fil pourrait être hors service avant la fin de sa mission. Optimiser la durée de vie de ces capteurs, et par conséquent la durée de vie des réseaux, est un défi majeur.

Dans ce chapitre, nous présentons notre contribution qui consiste à proposer une approche basée sur une technique computationnelle pour la construction et l'organisation du réseau de capteurs sans fil avec un protocole de routage basé sur une hybridation des protocoles de routage hiérarchiques pour minimiser la consommation d'énergie au sein des nœuds capteurs et donc prolonger la durée de vie du réseau.

### 1. Concepts de base de notre approche

La consommation d'énergie est un problème critique dans les réseaux de capteurs sans fil, c'est pour cela nous avons utilisé un algorithme génétique pour le clustering afin de minimiser les communications longue distance entre les capteurs et la station de base ce qui conduit à réduire la consommation d'énergie du réseau et prolonger la durée de vie de réseau en général.

## 1.1. Algorithme génétique

L'algorithme génétique est un modèle d'apprentissage automatique qui tire son comportement des processus d'évolution nature [51]. L'algorithme génétique crée un certain nombre de chromosomes au hasard pour effectuer une population initiale puis les chromosomes les plus en forme survivent à la génération suivante.

### 1.1.1. Source d'inspiration

Les algorithmes génétiques s'inspirent des mécanismes de l'évolution biologique pour les transposer à la recherche de solutions adaptées au problème qu'on cherche à résoudre. L'évolution biologique procède en sélectionnant des génotypes (intégrés aux chromosomes), sur la base de l'adaptation relative à leur environnement des phénotypes qu'ils génèrent (la qualité de cette adaptation est mesurée par la performance « fitness » relative de chaque génotype). Les génotypes qui sont les mieux adaptés à leur environnement ont une plus grande facilité à se reproduire et la reproduction (croisement) assure le croisement des gènes les plus performants dans la population [35].

### 1.1.2. Présentation d'un algorithme génétique

Un algorithme génétique implémente une version très simplifiée et très schématique des mécanismes de l'évolution biologique. À partir d'un problème qu'on cherche à résoudre, un algorithme génétique est défini, par quatre éléments de base suivants [35]:

- **Individu/chromosome/séquence** : une solution potentielle du problème qui correspond à une valeur codée de la variable (ou des variables) en considération;
- **Population** : un ensemble de chromosomes ou de points de l'espace de recherche (donc des valeurs codées des variables);
- **Environnement** : l'espace de recherche (caractérisé en termes de performance correspondant à chaque individu possible);
- **Fonction de performance (fitness)** : terme qui désigne la fonction d'évaluation d'un individu. Cette Fonction est liée à la fonction à optimiser et permet de définir le degré de performance d'un individu (d'une solution).

#### 1.1.2.1. Chromosome

Toute solution à tout problème en AG appelé chromosome ou individu. Le chromosome est représenté par un vecteur de paramètres également appelés (gènes) [50]. Si la valeur du gène est 1, alors le nœud correspondant est un CH et 0 autrement. Chaque chromosome représente une topologie de réseau alors que sa taille doit être égale à la taille du réseau (c.-à-d. nombre de capteurs). D'un autre côté, le nombre des chromosomes (topologies) est déterminé par le paramètre de taille de la population dans lequel l'augmentation de la taille de la population conduit à augmenter le nombre de

topologies suggérées. Le chromosome (topologie du réseau), qui est montré sur la figure 3.1, indique que le réseau a 10 capteurs. Les capteurs 4 et 9 sont des CHs et les autres sont des nœuds normaux.

S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
0	0	0	1	0	0	0	0	1	0

**Figure 3.1.** Un chromosome d'un RCSF [51].

### 1.1.2.2. Opérations de l'algorithme génétique

Les algorithmes génétiques ont trois opérations de base [51]: *sélection*, *croisement* et *mutation*. L'opération de sélection est utilisée pour choisir le chromosome (individus) en fonction de leur valeur physique par rapport à la population de la génération actuelle. En général, la roulette est utilisée pour effectuer l'opération de sélection dans laquelle le chromosome qui a une grande valeur de fonction de fitness a une probabilité plus élevée pour survivre à la prochaine génération sur les autres. Dans l'opération de croisement, les chromosomes sont recombinaisonnés résultant deux nouveaux chromosomes enfants à ajouter à la population de la prochaine génération. Généralement, l'algorithme génétique a une probabilité de croisement entre 0,6 et 1. L'augmentation de cette valeur entraîne l'augmentation de l'occurrence du croisement. Il est à noter que l'opérateur de croisement a de nombreux types dont le point unique de l'opérateur de croisement, ce qui signifie qu'après avoir sélectionné les chromosomes, AG générera un nombre aléatoire (entre 0 et taille du réseau -1) à séparer de ce point en deux parties pour être ensuite recombinaisonnés. Enfin, la mutation est appliquée à chaque bit (gène) d'un chromosome. En règle générale, l'opérateur de mutation retourne les bits de la structure des données chromosomiques. Généralement, l'algorithme génétique a une probabilité de mutation entre 0,01 et 0,001. Similaire à l'opérateur de croisement, l'augmentation de cette probabilité augmentera l'occurrence de mutation.

### 1.1.2.3. fonction de fitness

Mesurer la forme physique ou la performance des chromosomes se fait par une fonction de fitness spécifique. La fonction de fitness établit la base de la sélection des chromosomes qui sera muté. La fonction de fitness utilisée dans [50] ; quand  $D_i$  et  $H_i$  ont de petites valeurs, cela signifie que la fonction de fitness a une grande valeur. Dans ce cadre, l'algorithme génétique essaie de trouver la plus grande valeur de fonction de fitness afin d'obtenir le nombre minimum de clusters et les communications longue distance dans le but de maximiser la durée de vie du réseau.

#### 1.1.2.4. Résiliation de l'algorithme génétique

L'algorithme génétique est résilié juste après un nombre spécifié de générations (courses). Après cela, il recherche le meilleur chromosome trouvé dans la population [51]. Dans la plupart des cas c'est bon de dire qu'une valeur de fonction de fitness plus élevée signifie un minimum nombre de CHs et communications longue distance. Cependant, après un certain nombre d'essais (selon les résultats de simulation: supérieur ou égal à la taille de réseau), les variations des résultats de l'AG seront faibles. Par conséquent, AG trouvera la solution qui consiste à avoir la plus grande valeur de la fonction fitness.

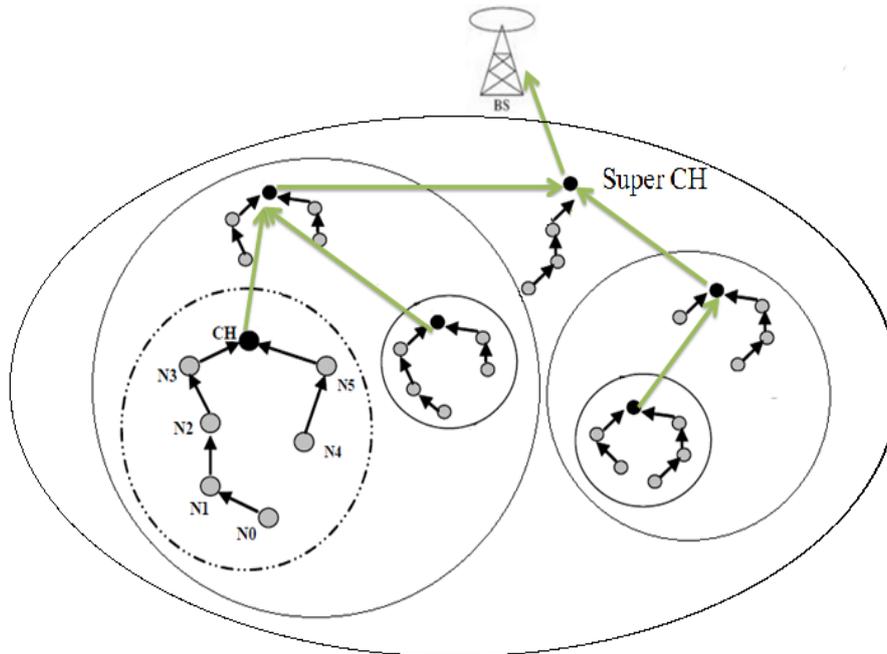
### 1.2. L'imbrication des clusters

La structure des RCSFs basée sur les groupements (clusters) était proposée pour réduire le coût et la complexité de fonctionnement du réseau, ainsi pour prolonger sa durée de vie. Cette configuration par clusters a donné naissance à une communication hiérarchique de deux niveaux. Le clustering hiérarchique en particulier est un ensemble de clusters imbriqués organisés en arborescence. Dans notre approche on utilise ce type de clustering pour construire des arbres de clusters des nœuds capteurs, dans lesquels deux clusters sont disjoints ou l'un inclut l'autre. Le cluster de tous les nœuds d'un RCSF est la racine de l'arbre.

## 2. Le principe de l'approche proposée

La minimisation de la consommation d'énergie et la maximisation de la durée de vie de réseau sont essentielles exigences dans la conception d'une mise en réseau optimale des protocoles de routage dans les RCSFs. Plusieurs protocoles de routage avec différents objectifs ont été déjà proposés pour l'efficacité énergétique ; parmi eux, le routage hiérarchique est considéré comme l'approche la plus favorable en termes de simplicité et d'efficacité énergétique. Deux grandes approches utilisent ce type de routage : une organisation basée sur les clusters et une organisation basée sur les chaînes. L'approche proposée est basée sur ces deux approches du clustering hiérarchique en utilisant un algorithme de l'intelligence computationnelle. L'objectif est d'organiser un RCSF par regrouper les nœuds capteurs dans des clusters à l'aide d'un clustering hiérarchique en utilisant l'algorithme génétique; et développer un protocole de routage à cette topologie qui permet la prolongation de la durée de vie de réseau, ce protocole combine les avantages des protocoles basés clusters (LEACH, HEED,...) et les protocoles basés chaînes (PEGASIS, ZHLS,...). Dans notre protocole une chaîne des nœuds dans chaque cluster est construite pour améliorer la dissipation d'énergie, les nœuds communiquent uniquement avec leurs voisins les plus proches et non directement avec le CH. Chaque CH envoie les données collectées au sein du cluster vers la BS via leurs superCHs,

suivant une chaîne construite pour relier les clusters imbriqués. L'agrégation des données de chaque nœud dans une chaîne réduit la quantité d'échange de données entre les nœuds et leur CH, pour préserver l'énergie des nœuds. La figure 3.2 montre l'organisation des nœuds au sein des clusters.



**Figure 3.2.** L'organisation des clusters d'un RCSF selon l'approche proposée.

N0 transmet ses données à son voisin proche N1; N1 fusionne les données avec le sien et les envoie à un autre voisin, jusqu'à atteindre CH, chaque CH fusionne les données des nœuds de son cluster qu'il les reçoit avec ces données et les envoie à son superCH, jusqu'à atteindre la BS. Par conséquent, dans cette nouvelle organisation (selon l'approche proposée), les distances de transmission sont abrégées et le nombre de nœuds communiquant avec le CH est considérablement réduit. Les chaînes formées dans les clusters ainsi que la chaîne externe qui est composée de l'ensemble des CHs et superCHs sont de petite taille et fonctionnent en même temps. Cela implique une meilleure conservation d'énergie et prolonge la durée de vie des CHs. Pour accéder au support, nous utilisons la méthode proposée dans LEACH et HEEP, où les CHs établissent un plan de transmission TDMA, dans lequel ils définissent l'heure exacte de chaque nœud où il peut transmettre ses données collectées. L'utilisation du programme TDMA évite les problèmes de collision et d'interférence à l'intérieur du même cluster.

## 2.1. La description et le déroulement du protocole de routage proposé

Le protocole proposé se déroule en périodes et chaque période se compose de deux étapes :

- La première étape pour le clustering qui consiste à la formation des clusters et à l'élection des CHs.
- La deuxième étape concerne la transmission de données.

### 2.1.1. Formation des clusters et l'élection des CHs par une technique computationnelle

Notre approche consiste à utiliser le clustering hiérarchique en utilisant l'algorithme génétique avec une détermination de la taille des clusters finaux (feuilles de l'arbre). L'arbre généré par le clustering hiérarchique peut correspondre à une taxonomie simple et significative ainsi que la division d'un RCSF en clusters ayant à peu près la même taille présente plusieurs avantages : les puissances de calcul des différents clusters sont similaires et la complexité d'une procédure exécutée sur un cluster doit être limitée par une fonction de cette taille, si un processus nécessite un ensemble d'un nombre donné de nœuds, la hiérarchie des clusters permet d'identifier un ensemble de nœuds de taille appropriée.

Après avoir appliqué l'algorithme génétique, le clustering hiérarchique sera appliqué sur la topologie sélectionnée. Généralement, l'approche proposée comporte les étapes suivantes:

---

#### **Algorithme 3.1. Les étapes de l'approche proposée pour la formation des clusters et l'élection des CHs.**

---

1. Déployer un certain nombre de capteurs dans le territoire et déterminer l'emplacement de la station de base.
2. Déterminer la taille de la population (le nombre de chromosomes ou topologies diverses).
3. Fixer la valeur de  $K$  qui est la taille approximative des clusters.
4. Déterminer le nombre d'exécutions (générations): combien d'analyses de l'AG seront appliquées sur les chromosomes.
5. Appliquer l'algorithme génétique sur le réseau comme suit:
  - a) L'algorithme génétique générera aléatoirement un nombre de chromosomes selon une valeur prédéterminée (taille de la population).
  - b) La fonction de fitness sera calculée pour chaque chromosome dans la population; la fonction de fitness utilisée dans notre approche est la même que celle employée dans [50].
  - c) Appliquer les opérations d'algorithmes génétiques (sélection, croisement et mutation).
  - d) Les étapes précédentes seront répétées jusqu'à ce qu'un nombre prédéterminé de parcours soient atteints.
  - e) Enfin, AG choisit le chromosome le plus adapté (solution) de la population.
6. Afin de calculer la taille de chaque cluster. Si la taille  $T$  du cluster  $X$  est supérieure à  $2K$  alors répéter la dernière étape précédente (appliquer l'algorithme génétique sur l'ensemble des nœuds de cluster  $X$ ).
7. Après avoir terminé l'algorithme génétique, le clustering hiérarchique sera effectué par les étapes suivantes:
  - a) Chaque CH calcule sa distance au sink et tous les CHs.
  - b) Si la distance au sink est la distance minimale par rapport aux distances pour atteindre d'autres CHs, donc le CH sera connectée directement au sink, sinon, il sera connecté au

*CH le plus proche.*

- c) *Le nombre de CHs connectés forment un groupe. Le CH le plus proche au sink est choisi pour être un super cluster head et rassemble les données requises des CHs au sein de son groupe pour être ensuite relayé au nœud sink.*

### 2.1.2. Transmission de données

A chaque étape de transmission, tous les nœuds appartenant au même cluster est délégué à la détection des données, où la tâche de ressources intensives (transmission de données à la BS) est attribuée aux plus puissants nœuds qui sont les CHs. Le coût énergétique de la phase de clustering est très faible par rapport à la phase de transmission, et dépend du nombre de nœuds dans le réseau et la distance à la BS.

Selon le réseau, nous pouvons l'estimer par la règle (3.1) utilisée dans [47]:

$$\sum_{i=1}^{i=N} (q_i E_{\text{elect}} + q_i E_{\text{fs}} d_{\text{toBS}}^2) \quad (3.1)$$

Où  $q_i$  est la taille du package de contrôle transmis par le nœud  $i$ ,  $E_{\text{elect}}$  est la puissance consommée par électronique circuits (énergie de calcul),  $E_{\text{fs}}$  est l'énergie perdue dans l'espace de transmission,  $d_{\text{toBS}}$  est la distance entre le nœud  $i$  et le BS, et  $N$  est le nombre de nœuds vivants dans le réseau.

L'étape de transmission se compose en deux phases : construction des chemins et l'envoi des données.

#### 2.1.2.1. Construction des chemins

Quand le processus d'élection des CHs est terminé, l'étape de la construction des chemins commence, selon l'organigramme suivant :

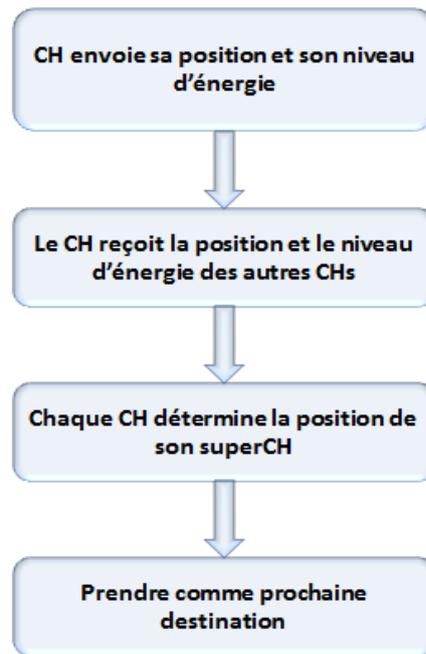


Figure 3.3. Schéma de construction des chemins.

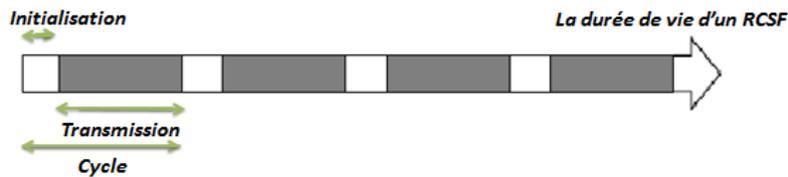
#### 2.1.2.1.1. Construction des chaînes

##### ➤ Construction des chaînes à l'intérieur des clusters

Dans ce travail, nous présentons deux approches pour la construction des chaînes qui sont l'approche dynamique et l'approche statique. Ces deux approches ont été également proposées dans l'article [47]. Ainsi que nous incluons des nouveaux résultats expérimentaux.

##### ✓ Avec une approche dynamique

L'approche dynamique de la construction des chaînes est divisée en plusieurs cycles d'exécution. Chaque cycle commence par une phase d'initialisation suivie d'une phase de transmission (figure 3.4.). La formation de clusters s'effectue dans un chemin dynamique et périodique au niveau de chaque phase d'initialisation par la station de base, chaque nœud du réseau envoie un paquet de données vers la BS contenant l'identifiant du nœud, la réserve d'énergie et la localisation sur le réseau (en utilisant, par exemple, le système de localisation GPS). La station de base lancera par commande l'opération de formation de clusters puis passera à l'élection des CHs. Chaque nœud a un tableau des nœuds qui composent la chaîne de transmission, chacun peut envoyer ses données capturées directement au nœud voisin dans la chaîne de transmission.



**Figure 3.4.** Les étapes d'exécution avec l'approche dynamique.

Pour construire les chaînes, nous avons adopté la même idée utilisée dans PEGASIS et HEEP où les nœuds d'un même cluster forment une chaîne des voisins les plus proches. Afin de réduire le degré de latence introduit par la chaîne, nous pouvons permettre à certains nœuds (les nœuds des clusters différents) de transmettre simultanément. Pour éviter les problèmes d'interférence, nous pouvons autoriser des nœuds géographiquement séparés d'émettre au même intervalle de temps.

Nous avons utilisé l'algorithme proposé dans [47] pour construire les chaînes; il suit le même concept utilisé dans PEGASIS où nous commençons par le nœud le plus éloigné de la BS (choisir un nœud arbitrairement s'il y a égalité, c'est-à-dire un ensemble de nœuds ayant la même distance par rapport à la BS). Ce nœud représente la tête de la chaîne. Ensuite, le nœud le plus proche de la tête de la chaîne est sélectionnée pour être ajoutée et devenir le nouveau chef de la chaîne. Les voisins successifs sont sélectionnés de cette façon parmi les nœuds non visités dans le cluster. L'opération est répétée jusqu'à ce que tous les nœuds soient dans la chaîne. En effet, l'algorithme proposé commence par le nœud plus éloigné pour assurer que les nœuds plus éloignés de CH ont des voisins proches. La construction des chaînes de plus petites tailles qui ont toutes la même tête (CH) réduit le problème de latence généré par la longueur de la chaîne dans PEGASIS. Ainsi, chaque cluster peut contenir X nombre de chaînes qui transmettent simultanément les données capturées au CH. Le nombre X dépend de la situation géographique du CH dans le cluster.

---

**Algorithme 3.2. Algorithme de construction de chaînes [47].**

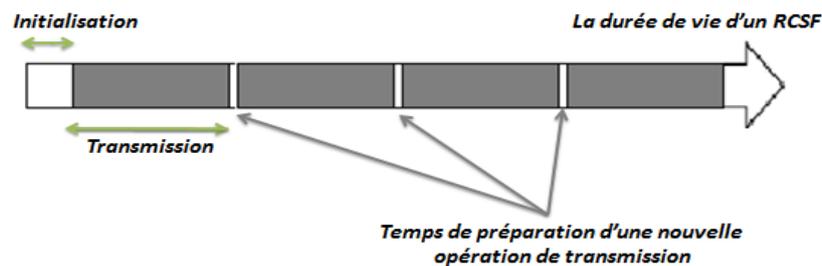
---

1. COMMENCER
2. Chaîne (i): la table des chaînes à construire,
3. Tête: la tête de la chaîne,
4. N: ensemble des nœuds appartenant au même cluster,
5.  $i = 1$
6. Supprimer (N, CH)
7. Chaîne (i) =  $\{\}$
8. Tête = le nœud le plus éloigné de CH
9. Supprimer (N, tête)
10. Tant que (N  $\neq \Phi$ ) faire
11. Le plus proche = le nœud le plus proche de la tête de la chaîne
12. Si (distance (tête, CH) > distance (tête, plus proche))
13. tête = le plus proche
14. Ajouter (chaîne (i), tête)
15. Supprimer (N, tête)
16. Fin si

17. Sinon
18. Ajouter (Chaîne (i), CH)
19.  $i = i + 1$
20. Aller à (7) // chaîne cassée //
21. Fin sinon
22. Fin tant que
23. FIN

### ✓ Avec une approche statique

Avec cette approche nous avons une seule phase d'initialisation où se forment les clusters, suivie de plusieurs tours de transmission.



**Figure 3.5.** Les étapes d'exécution avec l'approche statique.

Lorsque les chaînes de clusters sont organisées une seule fois, On peut empêcher les nœuds de perdre plus d'énergie en raison de la périodicité de la transmission des paquets de commande à la BS à chaque phase d'initialisation imposée par l'approche de clustering dynamique. De plus, les frais généraux engendrés par cet échange est considérablement réduit. La phase de transmission est la même que celle utilisée par l'approche dynamique. La phase d'initialisation n'étant accomplie qu'une seule fois et donc l'algorithme de formation des clusters et d'élection des CHs sera exécuté uniquement une fois. Par conséquent, la station de base devra construire une chaîne de nœuds (en utilisant le même algorithme utilisé dans l'approche dynamique « Algorithme 3.2.») pour chaque cluster et CH précédemment élu. Les informations sur les chaînes construites seront rassemblées par les CHs sous forme de tables indexées. La figure 3.6 présente un exemple d'une table des chaînes formée.

<i>CH1</i>	<i>Chaîne 1</i>
<i>CH2</i>	<i>Chaîne 2</i>
<i>CH3</i>	<i>Chaîne 3</i>
-	-
-	-
-	-
<i>CHn</i>	<i>Chaîne n</i>

**Figure 3.6.** Exemple d'un tableau des chaînes indexées.

Les tableaux des chaînes seront transmis à tous les nœuds du réseau à la fin de la phase d'initialisation. Au début de chaque cycle d'exécution, chaque nœud du réseau achemine

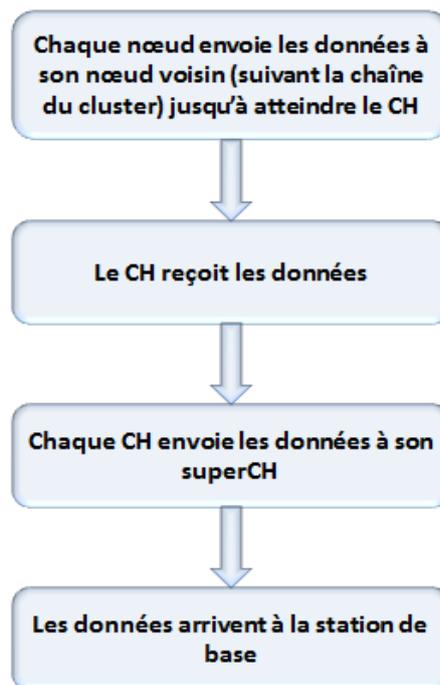
ses données collectées, en utilisant la chaîne respective, avec son CH pour ce cycle. Si un nœud capteur épuise sa réserve d'énergie les données sont acheminées directement à son voisin successif dans la chaîne afin d'éviter les pertes d'informations.

### ➤ Construction des chaînes externes

Les chaînes externes sont des chaînes comportent l'ensemble des CHs pour acheminer les données collectées des CHs à la station de base. Nous pouvons travailler avec le même principe de construction des chaînes à l'intérieur des clusters pour construire les chaînes externes avec les deux approches utilisées précédemment (l'approche dynamique et l'approche statique). Le nombre de CHs est limité donc les chaînes externes construites seront de petites tailles qui nous évitent le problème de latence généré par la longueur des chaînes dans les approches classiques basées chaîne.

#### 2.1.2.2. L'envoi de données

Cette phase consiste de transmettre les données collectées par les nœuds à leurs CHs, ensuite chaque CH transfère ces données à leur superCH jusqu'à la station de base selon l'organigramme suivant:



**Figure 3.7.** Schéma de transmission de données.

Une fois que la création des clusters associés et la construction des chemins sont achevées, la transmission des données peut commencer. Chaque nœud envoie ces données à son nœud voisin selon la chaîne de son cluster jusqu'à atteindre son CH associé et chaque CH agrège les données reçues en un seul paquet et l'envoie à son superCH jusqu'à ce qu'il atteigne la station de base.

## Conclusion

Dans ce chapitre, une nouvelle approche basée sur l'intelligence computationnelle a été développée pour traiter le problème de Clustering afin de proposer un meilleur protocole permettant une consommation d'énergie optimale. Ça nous permet de passer au dernier chapitre qui comporte l'étude et l'implémentation du protocole proposé.

# Chapitre 4

## Simulation et analyse des performances

*"On fait la science avec des faits, comme on fait une maison avec des pierres, mais une accumulation de faits n'est pas plus une science qu'un tas de pierres n'est une maison."*

**Henri POINCARÉ**

---

### Introduction

Avant sa mise en place, le déploiement d'un réseau de capteurs nécessite une phase de simulation, afin de s'assurer du bon fonctionnement de tous les dispositifs. En effet, pour de grands réseaux sans fil le nombre de capteurs peut atteindre plusieurs milliers et donc un coût financier relativement important. Il faut donc réduire au maximum les erreurs de conception possibles en procédant à une phase de validation.

Les réseaux de capteurs sans fil disposent des caractéristiques, comme la densité importante des nœuds, leurs autonomies énergétiques limitées et la topologie qu'ils forment, exigent des protocoles de routage spécifiques, différents de ceux déployés dans les réseaux usuels. De ce fait, le développement de nouveaux protocoles de routage s'avère indispensable. Ces protocoles doivent tenir compte de l'aspect fonctionnel de ces réseaux tout en optimisant l'efficacité énergétique des nœuds pour prolonger la durée de vie de réseau de capteurs.

Dans ce chapitre, nous allons en premier lieu, présenter la plate-forme logicielle que nous avons utilisé pour l'implémentation et la simulation, ensuite, nous allons évaluer le nouveau protocole de routage proposé avec discussion de résultats. Les performances de notre protocole et leur efficacité dans la consommation énergétique sera comparé avec le célèbre protocole de routage LEACH.

### 1. A propos de l'environnement MATLAB

MATLAB (*MATrix LABoratory*) 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.

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. Dans MATLAB, les IHM (*Interface Homme Machine*), sont appelés GUI (*Graphique User Interfaces*). Elles permettent à des objets graphique (boutons, menus, cases à cocher, ...) d'interagir avec un programme informatique [52].

Partout dans le monde, des millions d'ingénieurs et de scientifiques utilisent MATLAB pour analyser et concevoir les systèmes et produits de demain. MATLAB est présent dans des systèmes automobiles de sécurité active, des véhicules spatiaux, des appareils de surveillance médicale, des réseaux électriques intelligents et des réseaux mobiles LTE. Il est utilisé dans les domaines de l'apprentissage automatique, le traitement du signal, la vision par ordinateur, les communications, la finance computationnelle, la conception de contrôleurs, la robotique et bien plus [53].

L'environnement de MATLAB possède 4 fenêtres :

- Au centre l'invite de commande (command window).
- En haut à droite le contenu de l'espace courant de travail (workspace).
- Gauche la liste des fichiers du répertoire courant (currentfolder).
- En bas à droite l'historique des commandes tapées (command history).

### 1.1. Principales fonctionnalités du MATLAB [53]

- Langage de haut niveau pour le calcul scientifique et technique;
- Environnement bureau pensé pour l'exploration itérative, la conception et la résolution de problèmes;
- Graphiques destinés à la visualisation de données et outils conçus pour créer des tracés personnalisés;
- Applications dédiées à l'ajustement de courbes, la classification de données, l'analyse de signaux et bien d'autres tâches spécialisées;
- Boîtes à outils additionnelles conçues pour répondre à de nombreux besoins spécifiques aux ingénieurs et aux scientifiques;
- Outils permettant la création d'applications avec interface utilisateur personnalisée;
- Interfaces vers C/C++, Java, .NET, Python, SQL, Hadoop et Microsoft Excel;
- Options de déploiement libre de droits permettant de partager des programmes MATLAB avec les utilisateurs finaux.

## 1.2. Choix de l'environnement MATLAB

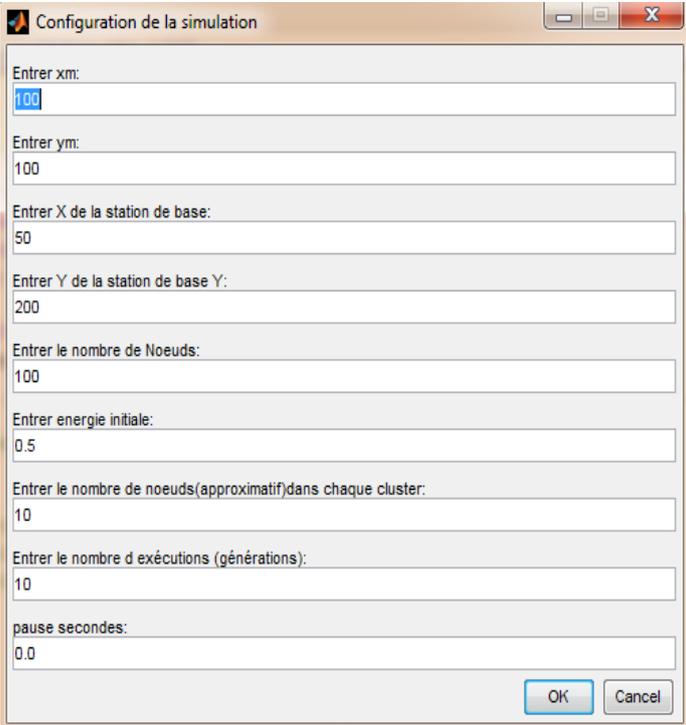
MATLAB est le logiciel le plus accessible et le plus productif conçu pour les ingénieurs et les scientifiques. Pour nos tâches d'analyse de données, de développement d'algorithmes et de création de modèles, MATLAB nous fournit un environnement qui encourage l'exploration et la découverte. MATLAB associe un langage haut niveau à un environnement bureau élaboré pour les flux de travail itératifs scientifiques et techniques [53].

Toutes les fonctionnalités mentionnées du MATLAB ainsi que sa nature qu'elle nous permet de résoudre des problèmes de calcul scientifique plus rapidement, nous ont fait le choisir comme un environnement d'implémentation pour notre protocole.

## 2. Les étapes d'exécution du protocole

### 2.1. Déterminer les paramètres de simulation

Pour l'implémentation de mise en œuvre de notre protocole proposé, nous avons utilisé l'outil de simulation MATLAB. Il y'a plusieurs paramètres dont nous avons besoin et qui conduisent à bien adapter la configuration de réseau et l'algorithme proposé par rapport à l'objectif de notre travail, tout en entrant les paramètres nécessaires dans une zone conçue à la détermination des paramètres initiaux du réseau.



Paramètre	Valeur
Entrer xm:	100
Entrer ym:	100
Entrer X de la station de base:	50
Entrer Y de la station de base Y:	200
Entrer le nombre de Noeuds:	100
Entrer energie initiale:	0.5
Entrer le nombre de noeuds(approximatif)dans chaque cluster:	10
Entrer le nombre d exécutions (génération):	10
pause secondes:	0.0

Figure 4.1. Les paramètres de simulation.

### 2.2. Lancer la simulation

Après la détermination des paramètres, on doit commencer la simulation et attendre que ça se termine pour voir l'affichage des résultats.

### 3. Paramètres de simulation

Nous assumons que tous les nœuds ont une position fixe durant toute la période de simulation pour simuler l'approche statique de notre protocole, et donc exécuter la phase d'initialisation pour une seule fois avec plusieurs tours de transmission. Le tableau suivant présente les paramètres utilisés dans notre modèle de simulation.

<i>Paramètre</i>	<i>Valeur</i>
<i>Taille du réseau</i>	<i>100 x 100 m<sup>2</sup></i>
<i>Localisation de la station de base</i>	<i>(50,200)</i>
<i>Nombre des nœuds</i>	<i>100</i>
<i>Energie initiale de chaque nœud</i>	<i>0.5 J</i>
<i>Taille approximative de chaque cluster</i>	<i>10 nœuds</i>
<i>Nombre d'exécutions (générations)</i>	<i>10</i>

**Tableau 4.1.** Les paramètres de simulation.

Sur une zone d'intérêt de 100m<sup>2</sup>, 100 nœuds capteurs de type normal sont aléatoirement déployés avec une station de base positionnée aux coordonnées (50m, 200m).

- ✓ Dans le cas de transmission ou de réception des paquets de données ainsi que le cas d'exécution des opérations de traitement de données au sein des capteurs, chaque nœud dans le réseau va consommer de l'énergie. Pour cette simulation, on néglige l'énergie consommée lors des calculs simples et les mises à jour en raison de leurs valeurs presque inexistantes.
- ✓ Tous les nœuds du réseau commencent la simulation par énergie initiale égale à 0.5 J et une quantité de données illimitées à transmettre à la station de base, l'énergie de la station de base est illimitée. Chaque nœud fonctionne lors des opérations de transmission c'est un nœud vivant mais lorsqu'il consomme sa réserve d'énergie est considéré comme mort, il ne peut ni transmettre ni recevoir des données.
- ✓ La probabilité de collision de signal et d'interférence dans le canal est ignorante et l'émetteur radio grâce à l'utilisation d'un plan de transmission TDMA, l'amplificateur radio et l'unité de fusion de données sont les principaux consommateurs d'énergie d'un nœud capteur.

### 4. Mesure de performance

- ✓ ***Durée de vie du réseau*** : L'intervalle de temps à partir du début de la simulation jusqu'à la mort du dernier nœud capteur.
- ✓ ***Vitesse de transmission des données*** : le nombre de paquets des données envoyés ou reçus par unité de temps.

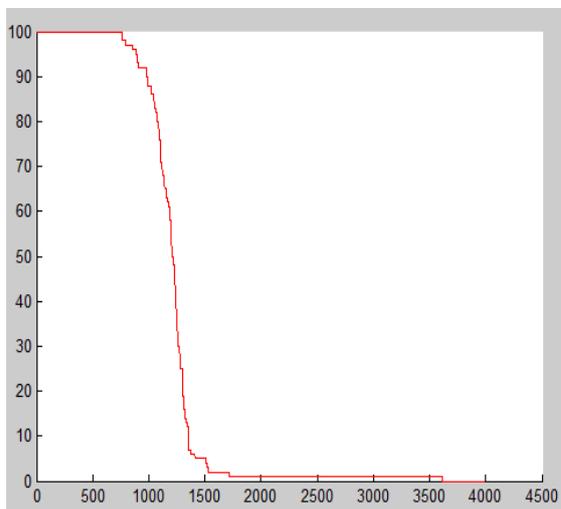
- ✓ **Période de stabilité** : L'intervalle de temps entre le début du fonctionnement du réseau jusqu'à la mort du premier capteur.
- ✓ **Période d'instabilité** : L'intervalle de temps entre la mort du premier capteur jusqu'à la mort du dernier capteur.
- ✓ **Premier nœud mort** : Nombre de ronds après lesquels le premier capteur est mort.
- ✓ **Dernier nœud mort** : Nombre de ronds après lesquels tous les nœuds capteur sont morts.

## 5. Simulation et résultats

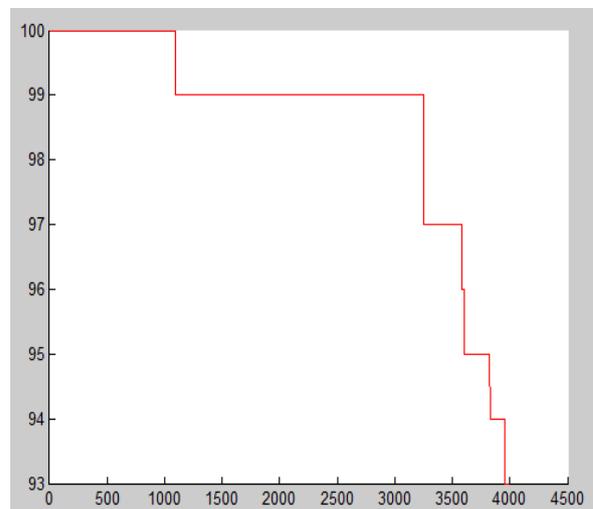
Les résultats de simulation que nous avons obtenus seront présentés dans cette partie, ces résultats seront comparés avec celles obtenus par le protocole LEACH qu'on a le développé pour prolonger la durée de vie du RCSF fonctionné avec ce protocole. On a modifié quelques paramètres de la simulation de ce protocole classique ainsi qu'on a remplacé la formule de sélection aléatoire des cluster-heads (4.1) par la formule (4.2) :

```
%Election of cluster heads
if(temp_rand<= (p/(1-p*mod(r,round(1/p)))) (4.1)
    countCHs=countCHs+1;
```

```
%Election of cluster heads
if(temp_rand<= (p/(1-p*mod(r,round(1/p))))*(S(i).E/totoal_energy) (4.2)
    countCHs=countCHs+1;
```



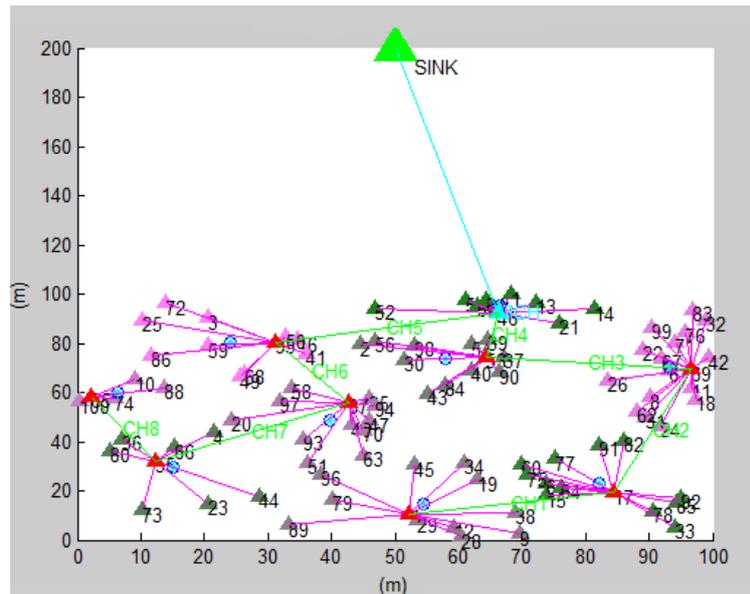
**Figure 4.2.** Durée de vie du réseau avec LEACH classique.



**Figure 4.3.** Durée de vie du réseau avec LEACH développé.

L'évaluation de notre protocole ce fait en terme d'économie d'énergie et de prolongation de la durée de vie global du réseau.

Notre modèle d'expérimentation comme nous avons dit précédemment est établi 100 nœuds répartis aléatoirement sur une surface carrée de 100 x 100 m<sup>2</sup> et une station de base. La figure suivante présente le déploiement aléatoire des nœuds capteurs :

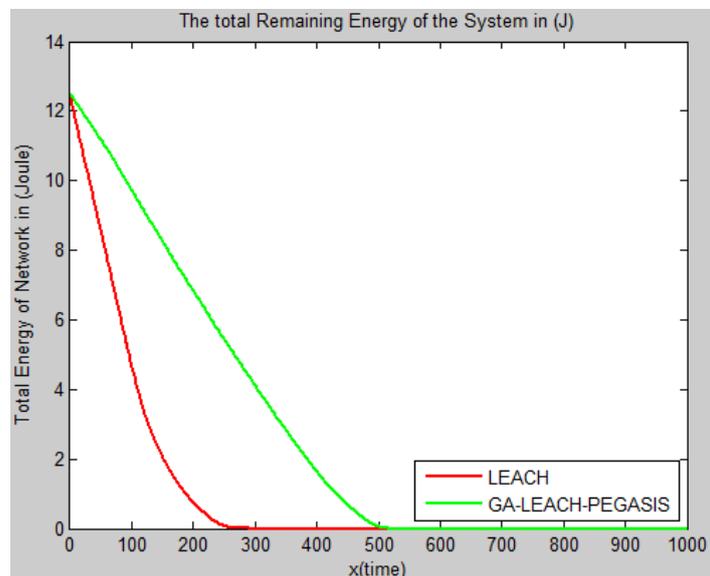


**Figure 4.4.** Le déploiement aléatoire des nœuds capteurs.

Après la simulation de notre protocole et le protocole LEACH, nous comparons leurs résultats ainsi que nous mesurons les performances du protocole proposé par les métriques mentionnées précédemment.

### 5.1. L'énergie résiduelle

La figure 4.5. , montre l'énergie résiduelle par rapport à chaque rond pour le protocole LEACH et notre protocole proposé.



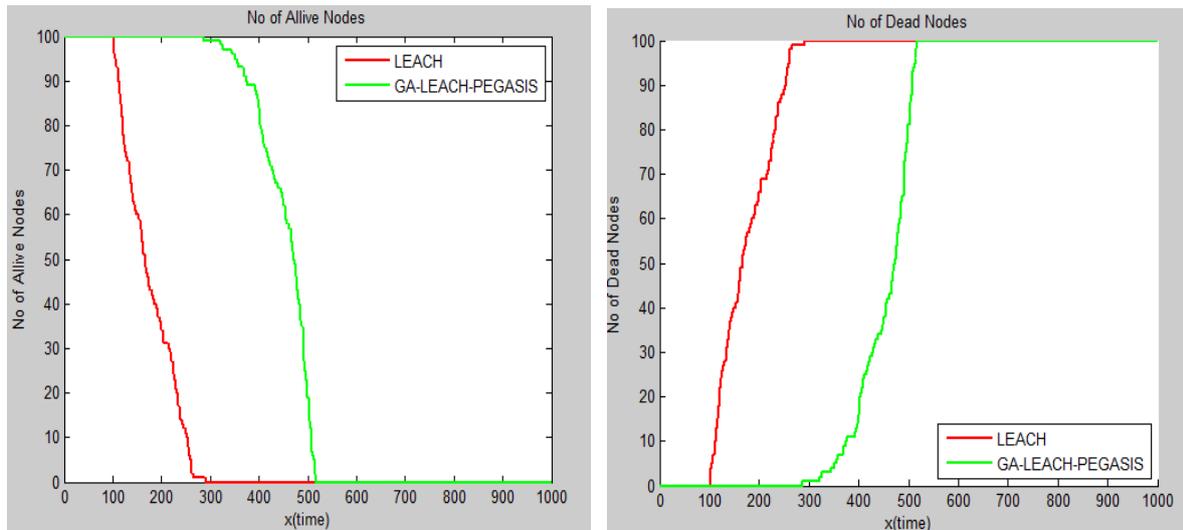
**Figure 4.5.** La consommation d'énergie versus les ronds (temps d'exécution).

On remarque qu'après chaque rond, l'énergie se diminue pour les deux protocoles LEACH et le protocole proposé. Cependant, en utilisant notre nouveau protocole, le réseau réserve mieux son énergie et par conséquent prolonge sa durée de vie et les nœuds restent vivants au minimum jusqu'à 500 unités de temps. Par contre dans le protocole LEACH l'énergie de réseaux est consommée entièrement à l'environ de 256 unités de

temps. Cette amélioration apportée par notre protocole est grâce à l'utilisation de l'algorithme génétique pour le choix optimal des positions des CHs. Cette optimisation est basée sur la maximisation de l'énergie totale de réseau et la minimisation de la communication entre l'ensemble des nœuds et leurs CHs ainsi que la communication entre les CHs et la station de base.

## 5.2. La durée de vie

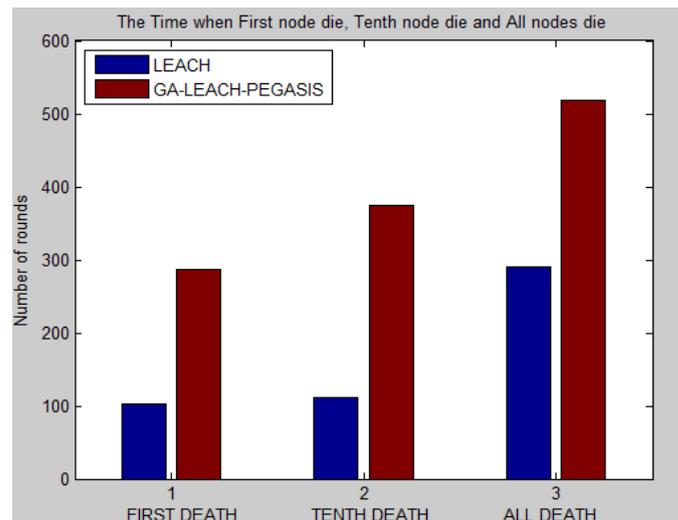
La figure 4.6. , montre les résultats de notre protocole et le protocole LEACH en termes de nœuds vivants et la figure 4.7. , montre les résultats des deux protocoles en termes de nœuds morts.



**Figure 4.6.** Nombre de nœuds vivants      **Figure 4.7.** Nombre de nœuds morts.

D'après les graphes au-dessus, on peut constater que le nombre des nœuds vivants diminue avec l'évolution des ronds pour les deux protocoles LEACH le protocole proposé. Mais pour notre protocole il reste des nœuds vivant jusqu'à 533 unité de temps contrairement au LEACH dont les nœuds sont totalement mort après 280 unité de temps. Cela due à la distribution optimal des CHs, l'affectation équilibré des membres à leurs CHs et aussi l'utilisation de technique de routage basée sur les chaînes qui permettent de préservé au mieux l'énergie des nœuds capteurs.

La figure 4.8. , montre les résultats obtenus en termes du premier nœud mort, dixième nœud mort et dernier nœud mort selon les deux protocoles par apport à chaque rond.



**Figure 4.8.** Les ronds relatifs au 1<sup>er</sup> nœud mort, 10<sup>ème</sup> nœud mort et dernier nœud mort.

La figure ci-dessus représente le nombre des nœuds mort dans les deux protocoles. On remarque que dans LEACH une diminution très rapide dans le nombre des nœuds mort en fonction du nombre de ronds. Le premier nœud mort dans le rond 93 et le dixième nœud mort dans le rond 110 et tous les nœuds sont morts après le 276<sup>ème</sup> rond. Ceci est ce qui provoque une durée de vie très court chez LEACH. Par contre dans notre protocole proposé on remarque une durée de vie plus longue, avec une diminution lente dans le nombre des nœuds mort en fonction du rond où Le premier nœud mort dans le 258<sup>ème</sup> rond et le dixième nœud mort dans le rond 408 et tous les nœuds sont morts après le 534<sup>ème</sup> rond.

Nous pouvons constater que, en termes d'optimisation de la durée de vie du réseau, notre protocole est toujours plus performant que LEACH et la durée de vie du réseau augmente proportionnellement au nombre de ronds.

### 5.3. Le nombre des CHs

La figure 4.9. , représente le nombre de CHs dans les deux protocoles. On constate que dans LEACH il y a un nombre élevé de CHs qui implique une consommation élevée d'énergie ; par contre dans notre protocole, un nombre fixe de CHs va mort lentement.

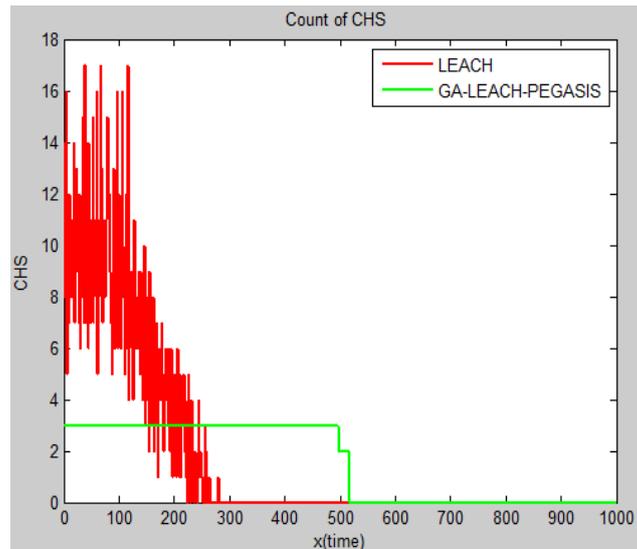


Figure 4.9. Nombre de CHs par unité de temps.

#### 5.4. La transmission des données au sink

La figure 4.10. , montre les résultats de simulation en termes de nombre de paquets reçu par la station de base selon les deux protocoles. Elle montre que notre protocole assure une agrégation de données ce qui offre une amélioration significative dans le nombre des données reçus dans le réseau.

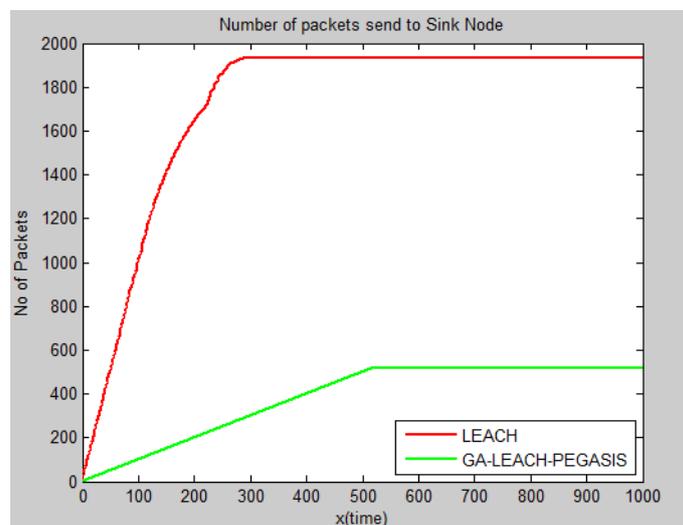


Figure 4.1. Nombre de paquets reçus par la station de base.

## 6. Discussion

Après avoir comparé les performances des deux protocoles LEACH notre protocole proposé, nous avons remarqué que notre protocole présente plusieurs avantages notamment dans la consommation d'énergie, l'augmentation des données reçues par la station de Base, un réseau avec une durée de vie plus longue et une bonne distribution des CHs. Ces avantages sont grâce à l'utilisation de l'algorithme génétique dans la sélection optimale des CHs et l'utilisation des chaînes conçues à l'intérieur des clusters

ainsi que les chaînes entre les CHs et les super-CHs pour le routage des données vers la station de base. Cependant, LEACH présente une variation énorme dans le nombre des CHs par rond, cela conduit à une mauvaise distribution des CHs sur le réseau qui conduit à l'apparition des nœuds isolés qui envoient leurs données directement à la station de base. Ce type de communication directe avec la station de base provoque une forte consommation d'énergie qui affecte la durée de vie globale du réseau.

## **Conclusion**

Dans ce chapitre, nous avons présenté l'environnement de simulation avec lequel nous avons travaillé, à savoir le simulateur MATLAB. Par la suite, nous avons exposé les paramètres de simulation ainsi que les résultats obtenus de la simulation qui confirment l'efficacité de notre protocole proposé.

Après avoir implémenté et simulé les deux protocoles LEACH et notre protocole proposé, nous avons démontré que ce dernier apporte beaucoup d'amélioration en termes de la conservation d'énergie, la quantité de données transmises à la station de base et la durée de vie de réseau.

# Conclusion Générale & Perspectives

*" La démocratie dont je suis partisan,*

*C'est celle qui donne à tous les mêmes chances de réussite,*

*et ensuite à chacun selon sa capacité."*

**Henry FORD**

---

## 1. Conclusion

L'objectif visé à travers ce projet de fin d'études est d'utiliser l'une des techniques de l'intelligence computationnelle pour la résolution des problèmes des réseaux de capteurs sans fil. Dans un premier temps, Nous avons présenté une introduction aux réseaux de capteurs sans fil concentrant sur leurs protocoles de routage et les techniques d'optimisation de l'énergie pour prolonger la durée de vie des RCSFs, Ensuite nous avons présenté un état de l'art sur le clustering et les paradigmes de l'intelligence computationnelle. Dans ce contexte, nous avons proposé une approche basée sur les algorithmes de l'intelligence computationnelle pour traiter le problème de Clustering afin de proposer un meilleur protocole permettant une consommation d'énergie optimale ainsi qu'une prolongation de la durée de vie des RCSFs. L'efficacité du protocole proposé a été évaluée et comparée à celle d'un protocole traditionnel (utilisant la technique de Clustering) LEACH. Les résultats expérimentaux ont montré que l'approche proposée optimise de manière efficace la consommation d'énergie par rapport aux protocoles traditionnels.

## 2. Perspectives

Notre travail ne s'arrête pas à ce stade, nous proposons comme amélioration de ce travail l'utilisation du clustering au sein d'un réseau de capteurs sans fil qui comporte des nœuds hétérogènes.

# Références bibliographiques

- 
- 
- [1] I. Aloui, "Une approche agent mobile pour les réseaux de capteurs", thèse de doctorat de l'université Mohamed Khider-Biskra.
  - [2] Clément SAAD, "Quelques contributions dans les réseaux de capteurs sans fil : Localisation et Routage", thèse de doctorat de École Doctorale 166 « I2S Mathématiques et Informatique » Laboratoire d'Informatique (EA 4128), Année 2009.
  - [3] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins : Security Protocols for Sensor Networks", *Wirel. Netw.* 8 (2002), no. 5, 521–534.
  - [4] C. Intanagonwiwat and R. Govindan and D. Estrin, "Directed Diffusion : a Scalable and Robust Communication Paradigm for Sensor Networks", *ACM MobiCom 2000*, Boston, MA (2000).
  - [5] F. Ye et al., "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks", *10th Int. Conf. Comp. Commun. and Networks* (2001), 304–309.
  - [6] B. Sahraoui, "Etude d'un protocole de routage basé sur les colonies de Fourmis dans les réseaux de capteurs sans fil ", mémoire de fin d'études pour l'obtention du diplôme de Master en Informatique, Université Abou Bakr Belkaid-Tlemcen, 2012-2013.
  - [7] Cesare Alippi, Giuseppe Anastasi, Cristian Galperti, Francesca Mancini, and Manuel Roveri, "Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications", In *Proceedings of the 4th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'07)*, pages 1\_6, Pisa, Italy, October 2007.
  - [8] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks", *1st Workshop. Sensor Networks and Apps.*, Atlanta, GA (2002).
  - [9] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, Enrico Gregori, and Andrea Passarella, "Understanding the real behavior of mote and 802.11 ad hoc networks: An experimental approach", *Pervasive and Mobile Computing*, pp. 237\_256, 2005.
  - [10] I. Akyildiz, W. Su, E. Cayirci, Y. Sankarasubramaniam. "A survey on sensor networks", *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102-114, Georgia Institute of Technology, Atlanta, USA. Août 2002.

- [11] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Micro sensor Networks", In proc of the Hawaii International Conference on Systems Science, vol. 8, pp. 8020, January 2000.
- [12] M. Badet, W. Bonneau. "Mise en place d'une plateforme de test et d'expérimentation", Projet de Master Technologie de l'Internet 1ere année, Université Pau et des pays de l'Adour. 2006.
- [13] Chérif DIALLO, "Techniques d'amélioration du routage et de la formation des clusters multi-sauts dans les réseaux de capteurs sans fil", thèse de doctorat, Ecole Doctorale EDITE, septembre 2010.
- [14] M. Ilyas and I. Mahgoub. "Handbook of sensor networks Compact wireless and wired Sensing Systems", ISBN 08493196864. CRC PRESS LLS, USA, 2005.
- [15] R. Kacimi, R. Dhaou, A.-L. Beylot, A. Delye de Mazieux, V. Gauthier, M. Marot, J. Vaudour, and M Becker, "Etat de l'art sur les réseaux de capteurs sans", In Livrable Projet CAPTEURS, SP1, V1.2, IRIT-ENSEEIH et INT, Rapport de recherche INT n\_05001 RST, 2006.
- [16] R. Kacimi, "Techniques de conservation d'énergie pour les réseaux de capteurs sans", thèse de doctorat, Institut National Polytechnique de Toulouse, septembre 2009.
- [17] Y. Chen and Q. Zhao, "On the Lifetime of Wireless Sensor Networks", in IEEE Communication letters, Vol. 9, No. 11, November 2005, pp. 976-978.
- [18] M. Esseghir, N.Bouabdallah, and G.Pujolle, "A Novel Approach for Improving Wireless Sensor Network Lifetime", in IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, 2005, pp. 2701-2706.
- [19] N.Mekki, K.Mohammedi, "Techniques de conservation d'énergie pour les réseaux de capteurs sans fil", Mémoire de Master, Université Dr. Tahar Moulay Saida, 2017-2018.
- [20] A. K. DWIVEDI, O. P. VYAS, "Network Layer Protocols for Wireless Sensor Networks: Existing Classifications and Design Challenges", International Journal of Computer Applications (0975 – 8887) Volume 8– No.12, 2010.
- [21] José CECÍLIO, João COSTA, Pedro FURTADO, "Survey on Data Routing in Wireless Sensor Networks", Springer-Verlag Berlin Heidelberg, 2010.
- [22] H.KARL, A.WILLIG, v"Protocols and Architectures for Wireless Sensor networks", Editorial John Wiley & Sons Ltd, ISBN 13978-0-470-09510-2, 2006.
- [23] R.Ababsa, "Etude comparative entre les protocoles hiérarchiques et le protocole multi-chaine PEGASISI", mémoire pour l'obtention du diplôme de Master, Université Larbi Ben M'hidi-Oum El Bouaghi, 2018.
- [24] Y. Benabbassi, "Application de la redondance pour la surveillance par réseau de capteurs sans fil : Cas du réseau de capteurs image sans fil ", thèse de doctorat, Université d'Oran, 2014.
- [25] Yacine younes, "Minimisation d'énergie dans un réseau de capteurs ", mémoire de magister, Université Mouloud Mammeri De Tizi-Ouzou.

- [26] S.Meldjem, Ch.Khalfi, N.Merabtine, K.Hadj Rabah "Les réseaux de capteurs ", Université des sciences et de la Technologie Houari Boumediene, 2013- 2014.
- [27] M. Heinzelman, ET W. Perillo, "Wireless Sensor Network Protocols", [éd.] CRC Hall. 2005.
- [28] K.Bouchakour, "Routage hiérarchique sur les réseaux de capteurs sans fil: Protocole KhLCH (K-hop Layered Clustering Hierarchy) ", mémoire présenté pour l'obtention d'un diplôme de MAGISTER EN INFORMATIQUE, Ecole Doctorale, 2012.
- [29] Y. Yaser, "Routage pour la Gestion de l'Energie dans les Réseaux de Capteurs Sans Fil", Thèse de Doctorat, université de Haute Alsace, 08 juillet 2010.
- [30] M. Aissani, A. Mellouk, N. Badache, and M. Boumaza, "A Novel Approach for Void Avoidance in Wireless Sensor Networks", Int'l Journal of Communication Systems (IJCS), John Wiley & Sons Editions, vol. 23(8), pp. 945-962, 2010.
- [31] S.Khedairia, "Contribution à la classification non supervisée : application aux données environnementales", thèse de Doctorat en informatique, université Badji Mokhtar Annaba, 2013-2014.
- [32] M.Mbao, "Distance sémantique et carte conceptuelle", Mémoire de stage de Master effectué au laboratoire LGI2P/EMA, Université Montpellier, 2007.
- [33] L.Candillier, "Contextualisation, visualisation et évaluation en apprentissage non supervisé", Doctoral dissertation, Université Charles de Gaulle-Lille III, 2006.
- [34] A.Blansché, "Classification non supervisée avec pondération d'attributs par des méthodes évolutionnaires", Thèse de doctorat, Université Louis Pasteur, 2006.
- [35] M.Sandeli, "Apport des paradigmes d'intelligence computationnelle pour la résolution des problèmes dans les réseaux de capteurs sans fil", Thèse de Doctorat en science, Université Abdelhamid Mehri-Constantine2, 2019.
- [36] Mr Dj.E. Boubiche, "Protocole de routage pour les réseaux de capteurs sans fil", MEMOIRE DE MAGISTERE EN INFORMATIQUE, Université de l'Hadj Lakhdar-Batna, Promotion 2007-2008.
- [37] S.Boulfekhar, "Approches de minimisation d'énergie dans les réseaux de capteurs", Mémoire de Magistère, Université Abderahmane Mira de Bejaïa, 2006.
- [38] Wendi Beth Heinzelman, "Application-Specific Protocol Architectures for Wireless Network", IEEE Transactions on Wireless Communications, Massachusetts Institute of Technology, June 2000.
- [39] S. Mujumdar, "Prioritized Geographical Routing In Sensor Networks", Thèse, Université Vanderbilt, Mai 2004.
- [40] O.Alaoui Ismaili, "Predictive clustering describe and predict simultaneously", Thèse de Doctorat, Université Paris-Saclay, 2016.
- [41] Y.Bouchal, "Intelligence computationnelle, Application : Imagerie par résonance magnétique(IRM)", Mémoire de fin de cycle pour l'obtention du diplôme de Master, Université Abderrahmane MIRA-Bedjaia, 2015-2016.

- [42] S.Jin, M.Zhou, et A. S.Wu, "Sensor network optimization using a genetic algorithm", in Proc. 7th World Multiconf. Systemics, Cybernetics and Informatics, 2003.
- [43] A.Karimi, S. M.Abedini, F.Zarafshan and S. A. R.Al-Haddad, "Cluster head selection using fuzzy logic and chaotic based genetic algorithm in wireless sensor network", J. Basic Appl. Sci. Research, Vol. 3, p. 694–703, 2013.
- [44] M. A.Mehr, "Design and implementation a new energy efficient clustering algorithm using genetic algorithm for wireless sensor networks", World Acad. Sci. Eng. Technol. 53 (2011) 430–433.
- [45] M.Baskaran, and C.Sadagopan, "Synchronous firefly algorithm for cluster head selection in WSN", J. Sens. Netw. 79(6) 780–789, 2015.
- [46] P.Kuila, and P. K.Jana, "Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach", Eng. Appl. Artif. Intell. 33 127–140, 2014.
- [47] D.E. Boubiche, A. Bilami, " HEEP(Hybrid Energy Efficiency Protocol) based on chain clustering",Int. J. Sensor Networks, Vol. 10, Nos. 1/2, pp.25–35, 2011.
- [48] Samia A. Ali, Shreen K. Refaay, "Chain-Chain Based Routing Protocol ", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011.
- [49] X. Liu, "A Survey on Clustering Routing Protocols in Wireless Sensor Networks", South China University of Technology, Guangzhou 510641, China, *Sensors* 2012.
- [50] Shiyuan Jin, Ming Zhou, and Annie S. Wu, "Sensor Network Optimization Using a Genetic Algorithm, " *Proceedings of the 7th World Multiconference on Systemics, Cybernetics, and Informatics*, Orlando, FL, July 27-30, 2003.
- [51] Mohammad M. Shurman, Mamoun F. Al-Mistarihi, Amr N. Mohammad, Khalid A. Darabkh, and Ahmad A. Ababnah, "Hierarchical Clustering Using Genetic Algorithm in Wireless Sensor Networks", Jordan University of Science and Technology, MIPRO 2013.
- [52] V. Kawadia and P. R. Kumar, "Power Control and Clustering in Ad Hoc Networks", IEEEINFOCOM, 2003.
- [53] <https://matlabpourtous.com/presentation/presentation-de-matlab/>