

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Béjaïa  
Faculté des Sciences Exactes  
Département d'Informatique

## *MÉMOIRE DE MASTER RECHERCHE*

En  
Informatique

Option  
*Réseaux et Systèmes Distribués*

### Thème

Clustering inégale pour le routage dans les  
réseaux de capteurs sans fil

Présenté par : M<sup>lle</sup> Nait Slimane Wissam  
M<sup>lle</sup> Redjdal Nawel

Soutenu le premier octobre 2020 devant le jury composé de :

Président	M. Atmani Mouloud	MCB	U. A/Mira Béjaïa.
Examineur	Mme. Houha Amel	MAA	U. A/Mira Béjaïa.
Promoteur	M. Mir Foudil	MAA	U. A/Mira Béjaïa.

Béjaïa, 01/10/2020.

## *\* Remerciements \**

Nous remercions Dieu de nous avoir accordé des connaissances de la science et de nous avoir aidés à réaliser ce travail.

Nos remerciements vont en premier lieu à notre encadrant M. MIR Foudil pour sa grande disponibilité, ses précieux conseils, ses encouragements et ses analyses qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.

Nous adressons nos sincères remerciements à chacun des membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nous nous remercions tous nos professeurs et toute l'équipe pédagogique de l'université qui ont travaillé avec abnégation pour nous permettre de suivre ce cursus. Ainsi qu'à tous ceux et celles qui ont contribué de près ou de loin à l'accomplissement de ce travail.

# Dédicaces

.... Je dédie ce travail :

A

Mes Parents et mes grands parents  
Pour tous leurs sacrifices, leurs soutiens,  
Leurs encouragements et leurs amours qui ont  
été la raison de ma réussite.  
Que dieu leur présente une bonne santé et une  
longue vie.

A

Mes sœurs et mes nièces Marisse et Darine  
Pour leur disponibilité à entendre mes  
frustrations et les sources de mon stress avec mes  
souhais de bonheur et de réussite dans leur vie.

A

Tous mes cher amis(es) que j'aime et qu'ils  
m'aiment, qu'ils trouvent dans ce travail  
l'expression de mes sentiments les plus  
affectueux, ainsi que ma binôme Nawel.

★ ★ ★

Wissam Nait slimane

# Dédicaces

.... Je dédie ce travail :

*A*

Mes Parents

Pour tous leurs sacrifices, leurs soutiens,  
Leurs encouragements et leurs amours qui ont  
été la raison de ma réussite.  
Que dieu leur présente une bonne santé et une  
longue vie.

*A*

Mes sœurs et Mes frères

Pour leur disponibilité à entendre mes  
frustrations et les sources de mon stress  
Avec mes souhaits de bonheur et de réussite dans  
leur vie.

*A*

Tous ceux que j'aime et qu'ils m'aiment.  
Qu'ils trouvent dans ce travail l'expression de  
mes sentiments les plus affectueux.

★ ★ ★

Redjdal Nawel

# Table des matières

Table des matières	i
Table des figures	v
Liste des tableaux	vi
Liste des algorithmes	vii
Notations et symboles	viii
Introduction générale	1
<b>1 Généralités sur les réseaux de capteurs sans fil</b>	<b>3</b>
1.1 Introduction	3
1.2 Réseaux sans fil	4
1.2.1 Réseaux avec infrastructures (réseaux cellulaires)	4
1.2.2 Réseaux sans infrastructures (réseaux ad hoc)	4
1.3 Réseaux de capteurs sans fil	7
1.3.1 Définition	7
1.3.2 Caractéristiques d'un RCSF	8
1.3.3 Domaines d'application des RCSFs	9
1.3.4 Problématiques dans les RCSFs	10
1.3.5 Description d'un nœud capteur	11
1.3.6 Architecture d'un nœud capteur	11
1.4 Notion de cluster et de clustering	13

---

1.4.1	Définition . . . . .	13
1.5	Architecture de clustering dans RCSF . . . . .	14
1.5.1	Avantages du clustering . . . . .	14
1.6	Architecture de clustering inégal dans RCSF . . . . .	15
1.6.1	Objectifs du clustering . . . . .	15
1.7	Conclusion . . . . .	16
<b>2</b>	<b>État de l'art des protocoles de routages dans les RCSFs</b>	<b>17</b>
2.1	Introduction . . . . .	17
2.2	Algorithmes de clustering conçus pour les RCSFs . . . . .	18
2.2.1	Low Energy Adaptive Clustering Hierarchy (LEACH) . . . . .	18
2.2.2	Low Energy Adaptive Clustering Hierarchy Centralised (LEACH-C) . . . . .	20
2.2.3	Protocole d'Élection Stable SEP (LEACH Hétérogène) . . . . .	21
2.2.4	Multi Hop Low Energy Adaptive Clustering Hierarchy (MH-LEACH) . . . . .	22
2.2.5	Two Level Low Energy Adaptive Clustering Hierarchy (TL-LEACH) . . . . .	22
2.2.6	Vice-cluster-head Low Energy Adaptive Clustering Hierarchy(V-LEACH) . . . . .	23
2.2.7	Tableau comparatif de LEACH et ses variantes . . . . .	24
2.3	Algorithmes de clustering inégaux conçus pour les RCSFs . . . . .	25
2.3.1	Energy-Aware Distributed Unequal Clustering (EADUC) . . . . .	25
2.3.2	Improved Energy-Aware Distributed Unequal Clustering ( Improved EADUC) . . . . .	29
2.3.3	Unequal Cluster-based Routing protocol (UCR) . . . . .	30
2.3.4	Energy and Coverage-aware Distributed Clustering Protocol (ECDC) . . . . .	34
2.3.5	Energy Neutral Clustering protocol (ENC) . . . . .	37
2.3.6	Balanced Clustering Algorithm with Distributed Self-Organization (DSBCA) . . . . .	40
2.3.7	Energy-Efficient Intra-Cluster Routing Algorithm (EIR) . . . . .	44

---

2.3.8	A Cluster-Head Rotation Approach for Sensor Networks (LAR-CH) . . . . .	46
2.3.9	Probability Driven Unequal Clustering Mechanism for WSN (PRODUCE) . . . . .	51
2.3.10	Energy Driven Unequal Clustering (EDUC) . . . . .	51
2.3.11	Energy Efficient Unequal Clustering (EEUC) . . . . .	52
2.3.12	Unequal Low Energy Adaptive Clustering Hierarchy (Unequal LEACH) . . . . .	52
2.3.13	Multi-hop Routing Protocol with Unequal Clustering (MRPUC)	52
2.3.14	Partition Energy Balanced and Efficient Clustering Scheme (PEBECS) . . . . .	53
2.3.15	Energy Aware Fuzzy Unequal Clustering algorithm (EAUCF)	53
2.3.16	Improved Fuzzy Unequal Clustering algorithm (IFUC) . . . . .	53
2.3.17	Energy Balanced Unequal Clustering for Wireless Sensor Networks (EBUC) . . . . .	54
2.3.18	Genetic Algorithm based Energy-Efficient Adaptive Clustering Hierarchical Protocol (GAEEP) . . . . .	54
2.3.19	Energy Degree Distance Unequal Clustering Algorithm (ED-DUCA) . . . . .	55
2.3.20	Unequal Clustering and Connected Graph Routing Algorithm (UCCGRA) . . . . .	55
2.3.21	Unequal Clustering Scheme (UCS) . . . . .	55
2.3.22	Coverage-preserving clustering protocol (CPCP) . . . . .	56
2.3.23	Hybrid Energy Efficient Distributed Clustering (HEED) . . . . .	60
2.3.24	Unequal Hierarchal Energy Efficient Distributed Clustering (UHEED) . . . . .	61
2.3.25	Hybrid Unequal Clustering with Layering protocol (HUCL) . . . . .	62
2.3.26	Tableau comparatif des protocoles de clustering inégales . . . . .	64
2.4	Conclusion . . . . .	66
<b>3</b>	<b>Le protocole UCP (Unequal Clustering Protocol)</b>	<b>67</b>
3.1	Introduction . . . . .	67

---

3.2	Hypothèses . . . . .	67
3.3	Description de la proposition . . . . .	68
3.3.1	Proposition . . . . .	68
3.3.2	Fonctionnement du protocole . . . . .	70
3.4	Simulation et évaluation des performances . . . . .	74
3.4.1	Le choix de MATLAB . . . . .	74
3.4.2	Simulation . . . . .	74
3.5	Conclusion . . . . .	82
	<b>Conclusion et perspectives</b>	<b>83</b>
	<b>Bibliographie</b>	<b>84</b>

# Table des figures

1.1	Réseau avec infrastructure (cellulaire) . . . . .	4
1.2	Réseaux Ad Hoc . . . . .	5
1.3	Changement de la topologie d'un réseau ad hoc . . . . .	6
1.4	Réseau de capteurs sans fil . . . . .	8
1.5	Applications des réseaux de capteurs sans fil . . . . .	10
1.6	Un nœud capteur . . . . .	11
1.7	Architecture d'un nœud capteur . . . . .	12
1.8	Exemple de structure de clusters . . . . .	13
1.9	Architecture de clustering dans RCSF. . . . .	14
1.10	Architecture de clustering inégal dans RCSF. . . . .	15
3.1	Déploiement du réseau . . . . .	76
3.2	Energie résiduelle par rounds . . . . .	77
3.3	Durée de vie de réseau . . . . .	78
3.4	Durée jusqu'à épuisement d'énergie du premier nœud en fonction du nombre de nœuds . . . . .	79
3.5	Durée jusqu'à l'épuisement d'énergie de 50% des nœuds en fonction du nombre de nœuds . . . . .	80
3.6	Durée jusqu'à l'épuisement d'énergie du dernier nœud en fonction du nombre de nœuds . . . . .	81

# Liste des tableaux

2.1	Tableau comparatif du protocole LEACH et ses variantes. . . . .	24
2.2	Tableau comparatif des protocoles de clustering inégales. Hé :Hétérogène. Ho :Homogène. M-saut :Multi-sauts. Eq :Équilibrage de charge	65
3.1	Paramètres de simulation . . . . .	75
3.2	Épuisement d'énergie du premier nœud du réseau en fonction du nombre de nœuds. . . . .	80
3.3	Épuisement d'énergie de 50% de nœuds en fonction du nombre de nœuds déployés. . . . .	81
3.4	Épuisement d'énergie du dernier nœud du réseau en fonction du nombre de nœuds. . . . .	82

# Liste des algorithmes

1	LEACH	19
2	LEACH-C	21
3	EADUC	28
4	UCR	33
5	ECDC	36
6	ENC	39
7	DSBCA	43
8	EIR	46
9	LAR-CH	50
10	CPCP	59
11	Collecte des informations des nœuds voisins (pour chaque nœud su réseau)	71
12	Compétition des CHs(pour chaque nœud du réseau)	72
13	Phase de transmission (pour chaque CH)	73

# Notations et symboles

<b>A</b>	<b>AODV</b>	<b>A</b> dhoc <b>O</b> n <b>D</b> emand <b>D</b> istance <b>V</b> ector.
	<b>ADC</b>	<b>A</b> nalog to <b>D</b> igital <b>C</b> onverter.
<b>C</b>	<b>CPU</b>	<b>C</b> entral <b>P</b> rocessing <b>U</b> nit.
	<b>CH</b>	<b>C</b> luster <b>H</b> ead.
	<b>CM</b>	<b>C</b> luster <b>M</b> embre.
	<b>CPCP</b>	<b>C</b> overage <b>P</b> reserving <b>C</b> lustering <b>P</b> rotocol.
<b>D</b>	<b>DSR</b>	<b>D</b> ynamic <b>S</b> ource <b>R</b> outing.
	<b>DSBCA</b>	<b>B</b> alanced <b>C</b> lustering <b>A</b> lgorithm with <b>D</b> istributed <b>S</b> elf- <b>O</b> rganization.
	<b>DARPA</b>	<b>D</b> efence <b>A</b> dvenced <b>R</b> esearch <b>P</b> roject <b>A</b> gency.
	<b>DSDV</b>	<b>D</b> estination <b>S</b> equenced <b>D</b> istance <b>V</b> ecteur.
<b>E</b>	<b>EADUC</b>	<b>E</b> nergy <b>A</b> ware <b>D</b> istributed <b>U</b> nequal <b>C</b> lustering.
	<b>EDDUCA</b>	<b>E</b> nergy <b>D</b> egree <b>D</b> istance <b>U</b> nequal <b>C</b> lustering <b>A</b> lgorithm .

---

<b>EBUC</b>	<b>E</b> nergy <b>B</b> alanced <b>U</b> nequal <b>C</b> lustering.
<b>ECDC</b>	<b>E</b> nergy and <b>C</b> overage-aware <b>D</b> istributed <b>C</b> lustering protocol.
<b>ENC</b>	<b>E</b> nergy <b>N</b> eutral <b>C</b> lustering protocol .
<b>EIR</b>	<b>E</b> nergy-Efficient <b>I</b> ntra-Cluster <b>R</b> outing Algorithm.
<b>EAUCF</b>	<b>E</b> nergy <b>A</b> ware <b>F</b> uzzy <b>U</b> nequal <b>C</b> lustering algorithm.
<b>G</b> <b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning <b>S</b> ystem.
<b>GAEEP</b>	<b>G</b> enetic algorithm based <b>E</b> nergy <b>E</b> fficient <b>A</b> daptive <b>P</b> rotocol.
<b>H</b> <b>HEED</b>	<b>H</b> ybrid <b>E</b> nergy <b>E</b> fficient <b>D</b> istributed <b>C</b> lustering.
<b>HCPEE</b>	<b>H</b> ierarchical <b>C</b> lustering <b>P</b> rotocol with <b>E</b> nergetic <b>E</b> fficiency.
<b>I</b> <b>IEEE</b>	<b>I</b> nstitute of <b>E</b> lectrical and <b>E</b> lectronics <b>E</b> ngineers.
<b>IFUC</b>	<b>I</b> mproved <b>F</b> uzzy <b>U</b> nequal <b>C</b> lustering algorithm.
<b>L</b> <b>LEACH</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchical.
<b>LEACH-C</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchical <b>C</b> entralised.
<b>LEACH-F</b>	<b>F</b> ixed number of cluster <b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchical.
<b>M</b> <b>MH-LEACH</b>	<b>M</b> ulti <b>H</b> op <b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchical.
<b>MRPUC</b>	<b>M</b> ulti-hop <b>R</b> outing <b>P</b> rotocol with <b>U</b> nequal <b>C</b> lustering.

---

<b>N</b>	<b>NS</b>	Noeud <b>S</b> ource
	<b>NR</b>	Noeud <b>R</b> elais.
	<b>NC</b>	Noeud <b>C</b> ollecteur de données.
<b>O</b>	<b>ONR</b>	<b>O</b> ffice <b>N</b> aval <b>R</b> esearch.
<b>P</b>	<b>PRODUCE</b>	<b>P</b> robability <b>D</b> riven <b>U</b> nequal <b>C</b> lustering <b>M</b> echanism for WSN.
	<b>PEBECS</b>	<b>P</b> artition <b>E</b> nergy <b>B</b> alanced and <b>E</b> fficient <b>C</b> lustering <b>S</b> cheme.
<b>Q</b>	<b>QoS</b>	<b>Q</b> uality of <b>S</b> ervice.
<b>R</b>	<b>RCSF</b>	<b>R</b> éseau de <b>C</b> apteur <b>S</b> ans <b>F</b> il.
	<b>RC</b>	<b>R</b> ayon de <b>C</b> ommunication.
<b>S</b>	<b>SB</b>	<b>S</b> tation de <b>B</b> ase.
	<b>SEP</b>	<b>P</b> rotocole <b>É</b> lection <b>S</b> table.
<b>T</b>	<b>TL-LEACH</b>	<b>T</b> wo <b>L</b> ovel <b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchical.
	<b>TDMA</b>	<b>T</b> ime <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess.
<b>U</b>	<b>UCR</b>	<b>U</b> nequal <b>C</b> luster based <b>R</b> outing protocol.
	<b>UHEED</b>	<b>U</b> nequal <b>H</b> ierarchal <b>E</b> nergy <b>E</b> fficient <b>D</b> istributed <b>C</b> lustering.
	<b>ULEACH</b>	<b>U</b> nequal <b>L</b> EACH.
	<b>UCS</b>	<b>U</b> nequal <b>C</b> lustering <b>S</b> cheme.
	<b>UCCGRA</b>	<b>U</b> nequal <b>C</b> lustering and <b>C</b> onected <b>G</b> raph <b>R</b> outing <b>A</b> lgorithm.
<b>V</b>	<b>V-LEACH</b>	<b>V</b> ice cluster head <b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchical.

# Introduction générale

Depuis leur création, les RCSFs sont utilisés dans plusieurs domaines tels que : le domaine militaire, le domaine médical, etc. Et ce grâce à la facilité et le coût réduit de leurs installations.

Un RCSF est un ensemble de nœuds qui coopèrent pour faire la collecte des données dans une zone géographique. Le problème principal de ces réseaux est les capacités réduites des nœuds à cause de leurs petites tailles (le problème d'énergie), cela a motivé la création de nouveaux protocoles de routages pour résoudre ce problème.

Par ailleurs, le clustering est l'une des solutions proposée pour faire face au problème de l'énergie limitée des nœuds dans les RCSFs, cette solution consiste à regrouper les nœuds en clusters et de leur attribuer des tâches spécifiques. Il existe le clustering égal qui consiste à créer des clusters de tailles égales et le clustering inégal, ce dernier sert à éliminer le problème des points chauds en créant des clusters de tailles inégales pour équilibrer la charge entre les CHs (Cluster Head), le clustering inégal réduit les tailles de clusters proche de la station de base et la taille de cluster augmente tant que la distance à la SB augmente.

Dans ce travail, nous avons proposé un nouveau protocole de routage distribué appelé UCP (Unequal Clustering Protocol). Il partitionne le réseau en clusters de tailles inégales en se basant sur des rayons de compétitions ( $R_c$ ) prenant en compte la distance des nœuds à la SB et aussi l'énergie des CHs et leurs nœuds voisins. Ce qui pallie au problème de points chauds et équilibre la charge entre les nœuds.

Ce manuscrit est organisé en trois chapitres et chaque chapitre est détaillé comme suit : dans le premier chapitre, nous avons présenté quelques généralités sur les réseaux de capteurs sans fil. Le deuxième chapitre est un état de l'art de la recherche sur quelques protocoles de routage dans les RCSFs sera présenté et accompagné de différents algorithmes de chaque protocole. Le troisième chapitre est

consacré à la présentation de notre protocole de routage hiérarchique appelé UCP, à la fin nous concluons notre mémoire par une conclusion générale ainsi que des perspectives.

# Généralités sur les réseaux de capteurs sans fil

## 1.1 Introduction

Un réseaux de capteurs sans fil (Wireless Sensor Networks : WSNs en anglais) est un ensemble de nœuds inter connectés entre eux avec un support sans fil avec des signaux radio, ces nœuds sont chargés de collecter des données spécifiques selon l'environnement où ils sont placés.

Avant l'apparition de la notion du clustering, les nœuds transmettent leurs données à la station de base directement ce qui entraine l'épuisement de leurs batteries rapidement, pour résoudre ce problème, les chercheurs ont penser à utiliser le clustering.

Le clustering est le fait de regrouper les nœuds de réseau en petits clusters (groupes) où chaque cluster a un cluster head(CH) et des membres de cluster, ces derniers collectent les données de l'environnement et les transmettent vers les CHs et ces derniers à leurs tours transmettent leurs données vers la station de base en les agrégeant dans un seul paquet de données. De cette façon y'aura une économie d'énergie des nœuds.

Donc nous abordons d'abord dans ce chapitre une présentation sur le clustering, les RCSFs et les réseaux sans fil .

## 1.2 Réseaux sans fil

Un réseau sans fil (en anglais : Wireless network) est un réseau qui connecte différents nœuds ou capteurs entre eux par ondes radio. Les réseaux mobiles nous évitent la mise en réseau d'un câblage trop coûteux. Ils se divisent en deux classes, les réseaux avec infrastructures et les réseaux sans infrastructures [29].

### 1.2.1 Réseaux avec infrastructures (réseaux cellulaires)

Le réseau cellulaire comprend deux ensembles d'entités : les stations fixes et les stations mobiles, le type de liaison entre les stations fixes est filaires, elle est fiable avec haut débit, la liaison entre les SBs et les unités mobiles (UMs) est sans fil avec une bande passante assez limitée qui réduit la taille des informations échangées [29].

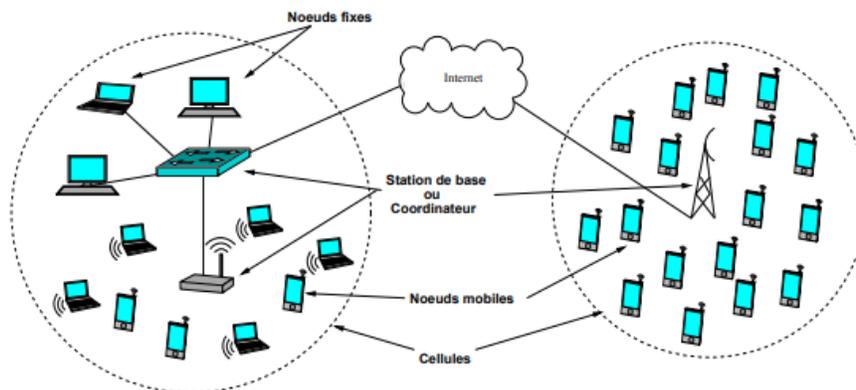


FIGURE 1.1 – Réseau avec infrastructure (cellulaire) [29].

### 1.2.2 Réseaux sans infrastructures (réseaux ad hoc)

Dans les réseaux sans infrastructure, il n'existe aucune partie filaire. Les nœuds communiquent directement entre eux avec des ondes radio. L'absence d'infrastructure

fixe oblige les nœuds du réseau à participer à la découverte des nœuds voisins et à la maintenance des routes dans le but d'acheminer les informations [33].

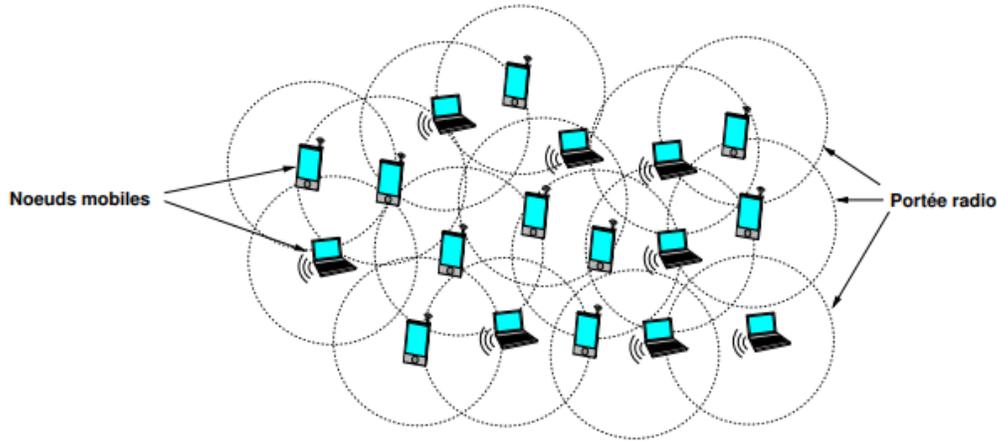


FIGURE 1.2 – Réseaux Ad Hoc [33].

### 1.2.2.1 Caractéristiques des réseaux mobiles ad hoc

Parmi les Caractéristiques des réseaux adhoc, nous pouvons citer :

- **Le médium de communication** : la communication sans fil ne nécessite aucun câblage et le médium sans fil est un médium de diffusion, c'est à dire, tous les nœuds qui sont dans la portée de transmission de l'émetteur peuvent recevoir le message [33] [30].
- **Une topologie dynamique** : Les unités du réseau se déplacent d'une façon libre, y'a possibilité de départ ou bien d'intégration ces dernières. Par conséquent, la topologie du réseau peut changer d'une manière rapide et aléatoire. Les liens de la topologie peuvent être unis ou bidirectionnels [13]. Comme le montre la figure suivante :

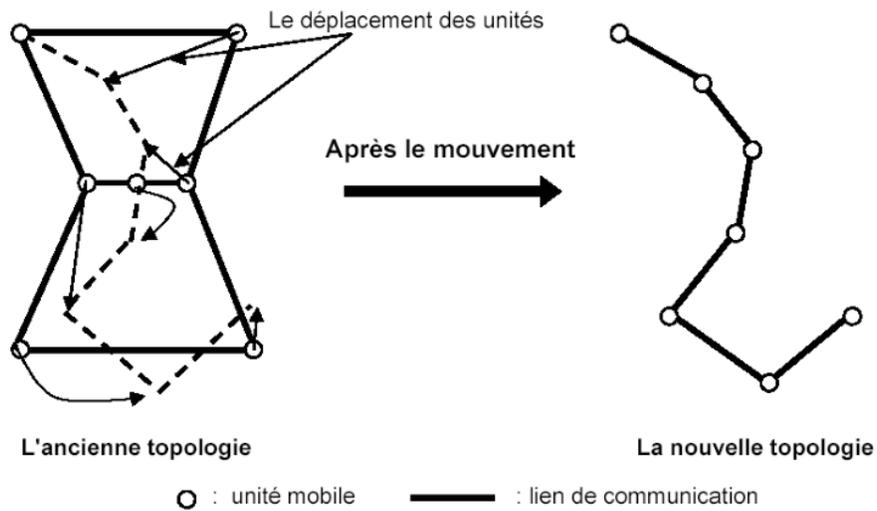


FIGURE 1.3 – Changement de la topologie d'un réseau ad hoc [13].

- **Une bande passante limitée** : La bande passante réservée à un hôte est limitée à cause de l'utilisation de médium radio qui est un médium de diffusion [13].
- **L'hétérogénéité des nœuds** : Les nœuds utilisés dans un réseau peuvent être différents en terme de types , de capacité de traitement(CPU, mémoire), de logiciel,etc. Cependant, malgré l'hétérogénéité des nœuds, ils doivent être en mesure de coopérer pour accomplir les tâches requises dans le réseau [33].

### 1.2.2.2 Avantages

1. Les réseaux ad hoc peuvent être déployés dans un environnement quelconque [13].
2. Le cout d'exploitation du réseau est faible[13].
3. Le déploiement d'un réseau ad hoc est simple : ne nécessite aucun câblage pré

et rapide puisqu'il est immédiatement fonctionnel dès que les terminaux sont présents [13].

### 1.2.2.3 Inconvénients

1. Limitation des possibilités de communication (Si une station envoie un signal radio, y'a que les stations qui sont dans son rayon qui vont recevoir ce signal, c'est à dire pour que deux stations qui sont loin l'une de l'autre se communiquent il faut qu'il y'aura un ensemble de stations pouvant assumer la fonction de routeur, afin de faire suivre les paquets de donnée échangés entre les deux stations) [13].
2. La sécurité dans les réseaux ad hoc est difficile à contrôler, y'a toujours des risques d'écoute, de vol et du modification de données [13].
3. La faible autonomie des batteries, ces dernières ne sont pas rechargeables ce qui entraine la mort des nœuds en créant des trous de couverture [13].

## 1.3 Réseaux de capteurs sans fil

### 1.3.1 Définition

Un RCSF, est un type spécial de réseau ad-hoc défini par un ensemble de nœuds déployés dans une zone géographique appelée, zone de sensation qui coopèrent pour accomplir une tâche spécifique [15].

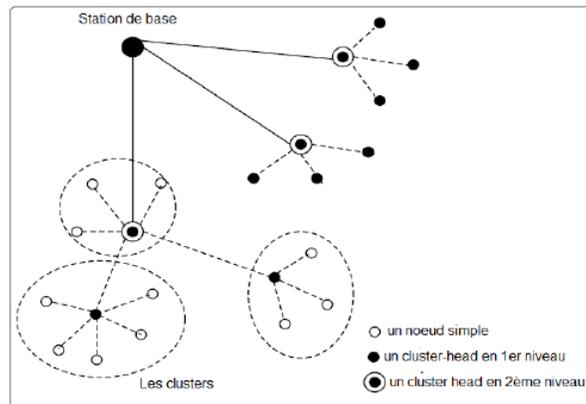


FIGURE 1.4 – Réseau de capteurs sans fil [15].

### 1.3.2 Caractéristiques d'un RCSF

Les caractéristiques importantes des réseaux de capteurs sans fil sont :

- **Topologie** : La topologie que l'on retrouve généralement dans les réseaux de capteurs est la suivante : les noeuds de capteurs sont déposés d'une manière hétérogène dans une zone, chaque noeud peut communiquer avec les autres noeuds qui sont situés dans son rayon de communication ou bien avec des noeuds plus loin en utilisant des noeuds routeur [6].
- **Routage** : Sa signifie le chemin que la données suit à partir d'un noeud source vers un noeud destination [6].
- **Mise à l'échelle** : Le protocole reste performant même si y'aura une augmentation importante dans le nombre des noeuds (être efficace quel que soit le nombre de capteurs dans le réseau) [6].
- **Energie limitée** : Les capteurs sont équipés de batteries avec une énergie limitée et non rechargeable. L'énergie consommée pour les calculs est négli-

geable mais celle consommée lors des communications est importante, donc il est nécessaire de limiter le nombre des communications [16].

- **Faible puissance de calcul** : Les capteurs souffrent d'un manque de puissance de calcul, cette faible puissance ne permet pas d'utiliser des algorithmes complexes dans les réseaux de capteurs sans fil [16].
- **La tolérance aux fautes** : Remédier aux différentes erreurs Les capteurs peuvent ne pas fonctionner correctement, ils peuvent tomber en panne à cause des altérations d'états comme les phénomènes climatiques (humidité, chaleur). Dans un cas pareil, le réseau doit être capable de détecter ce type d'erreur et d'y remédier [6].

### 1.3.3 Domaines d'application des RCSFs

Plusieurs domaines d'applications des RCSFs existent, et parmi eux on cite :

- **Domaine militaire** : Les applications militaires ont été les locomotives de la recherche pour les réseaux de capteurs. Dans le domaine militaire, les réseaux de capteurs sans fil offre beaucoup d'avantages, le ce types de réseau s'installe rapidement et sans aucune infrastructure. Ainsi, il offre un atout de taille pour surveiller les mouvements de l'ennemi[37].
- **Domaine urbain** : Les capteurs dans les milieux urbains sont exploités pour la localisation des bus, pour la sécurité [37].
- **Domaine médical** : Cette classe inclut des applications qui se chargent de fournir une interface d'aide pour les handicapés, collecter des informations physiologiques humaines de meilleure qualité, facilitant ainsi le diagnostic de certaines maladies, surveiller en permanence les malades et les médecins à

l'intérieur de l'hôpital [11].



FIGURE 1.5 – Applications des réseaux de capteurs sans fil [11]

### 1.3.4 Problématiques dans les RCSEs

Parmi les problématiques relevées, nous pouvons citer :

- Consommation d'énergie :

L'économie d'énergie fait partie des problématiques primordiales dans les réseaux de capteurs : les coûts de recharge des sources d'énergie sont très élevés, voire impossibles. Économiser l'énergie des capteurs est donc primordiale pour leur bon fonctionnement. En effet, un réseau de capteurs ne peut pas survivre si la perte de nœuds est très importante car ceci engendre des pertes de communications dues à une très grande distance entre les nœuds restants [12] [37].

- Sécurité :

La sécurité dépend de l'application, elle peut être critique. Souvent, les nœuds capteurs ne sont pas protégés contre les mauvaises manipulations ou attaques. Le réseau devrait donc permettre la détection des intrusions, et un fonctionnement robuste dans le cas de défaillance. Cependant, l'écoute, le brouillage, et les attaques de retransmission peuvent entraver ou empêcher l'opération, c'est pourquoi le contrôle d'accès, l'intégrité des messages, et la confidentialité doivent être garantis [4].

### 1.3.5 Description d'un nœud capteur

C'est un dispositif qui permet de transformer une grandeur physique observée (humidité, température, pression, etc. ) en une grandeur utilisable (intensité électrique, position d'un flotteur) et de la communiquer à un centre de contrôle par l'intermédiaire d'une station de base [14].



FIGURE 1.6 – Un nœud capteur [14].

### 1.3.6 Architecture d'un nœud capteur

Un nœud-capteur est composé de plusieurs éléments :

**Unité d'acquisition des données (L'unité de capture)** : Elle permet la

capture des données, c'est à dire la mesure des grandeurs physiques ou analogiques et leur conversion en données numériques. Elle est composée d'un capteur chargé de récupérer les signaux analogiques et de l'ADC (Analog to Digital Converter) qui reçoit ces signaux par la suite et qui a pour rôle de transformer et de communiquer les données analogiques en données numériques compréhensibles pour l'unité de traitement [48].

**Unité de traitement des données :** C'est l'unité principale du capteur, elle permet de contrôler le bon fonctionnement des autres unités et de faire les différents traitements. Elle peut embarquer un système d'exploitation sur certains capteurs afin de faire fonctionner le capteur. Elle peut aussi être couplée à une unité de stockage, qui servira par exemple à y enregistrer les informations transmises par l'unité de capture [48].

**Unité de transmission de données :** Elle a pour fonction la transmission et la réception de l'information. Équipée d'un couple émetteur/récepteur, elle permet la communication au sein du réseau [48].

**Source d'énergie :** Elle fournit l'énergie pour toutes les autres unités. En général, c'est l'équivalent d'une batterie ou une pile alimentant le capteur [48].

**Système de localisation :** Sert à déterminer la position des nœuds [48].

**Module de mobilité :** Sert à déplacer un nœud d'un lieu à un autre [48].

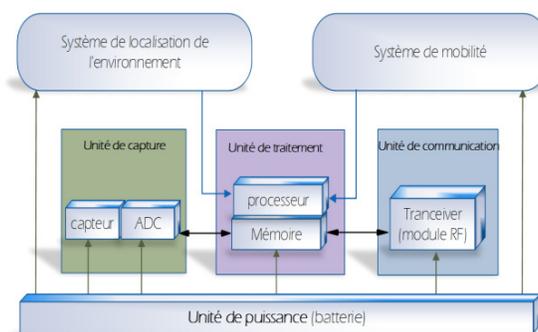


FIGURE 1.7 – Architecture d'un nœud capteur [48]

## 1.4 Notion de cluster et de clustering

### 1.4.1 Définition

Un cluster est un sous-ensemble de nœuds connexe, et la structuration ou clustering est le processus de regroupement des nœuds en clusters ce qui donne au réseau une structure hiérarchique [43]. Comme la montre la figure ci-dessus.

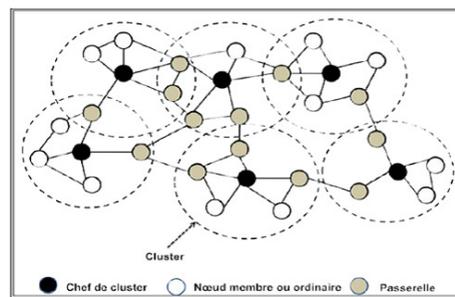


FIGURE 1.8 – Exemple de structure de clusters [27].

Les clusters comportent trois types de nœuds :

1. Le chef de cluster ou "cluster-Head" (CH) qui permet de coordonner les membres de son cluster, d'agréger et /ou de traiter les données collectées et de les transmettre au collecteur de données. Il est sélectionné pour ce rôle de manière déterministe (chef de cluster prédéfini), ou de manière aléatoire (chef de cluster élu parmi les nœuds du réseau selon une métrique bien particulière ou une combinaison de métriques) [27].
2. Un nœud passerelle ou "gateway" qui possède des liens inter-clusters et peut donc accéder à des clusters voisins et acheminer les données entre eux [27].
3. Un nœud ordinaire qui n'a pas de lien avec les autres clusters mais qui devient membre d'un cluster quand il s'attache à son chef [27].

## 1.5 Architecture de clustering dans RCSF

L'architecture du clustering dans les RCSF est illustrée dans la Figure 1.9 ; Il existe deux types du trafic en cluster RCSF : transmission de données au sein d'un cluster défini comme le trafic intra-cluster et transmission de données entre clusters qui est appelé comme trafic inter-cluster. Les membres du cluster détectent les paramètres du monde réel et transmettent la valeur détectée à son CH. Le CH reçoit et agrège les données pour supprimer les données redondantes et transmettre les données agrégées à la SB directement ou via des CH intermédiaires. Les CHs près du SB consomment plus d'énergie et évacue l'énergie plus rapidement que les CHs plus éloignés de SB[9].

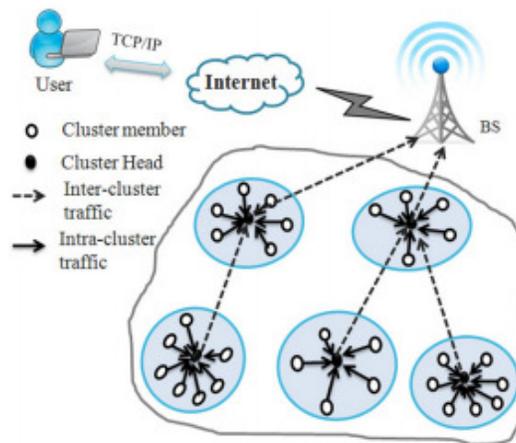


FIGURE 1.9 – Architecture de clustering dans RCSF.  
[9]

### 1.5.1 Avantages du clustering

la consommation d'énergie est réduite en améliorant l'utilisation de la bande passante, la réduction des frais généraux, une connectivité accrue, une topologie de réseau stabilisée, une diminution des retards et la réduction de la taille de la table de routage.

## 1.6 Architecture de clustering inégal dans RCSF

L'architecture du clustering inégal dans les RCSF est montré sur la Figure 1.10 ; Un clustering inégal réduit la taille des clusters plus près de SB et la taille du cluster augmente à mesure que la distance entre la SB et le CH augmente. La taille du cluster est directement proportionnelle à la distance entre les CHs et la SB. Quand la distance à la SB augmente, la taille du cluster augmente également. Petit cluster près de la SB indique moins de nœuds membres CMs et moins de trafic intra-cluster. Ainsi, les petits clusters consomment moins d'énergie pour le trafic intra-cluster et se concentre davantage sur le trafic inter-cluster. De même, des clusters plus grands plus éloignés de SB indiquent plus de CMs et plus de consommation d'énergie sur le trafic intra-cluster. Un clustering inégal permet à tous les CHs de dépenser la même quantité d'énergie pour que les CHs près de la SB dépendent autant d'énergie que les CHs plus éloignés de SB. Donc, le clustering inégal élimine les problèmes de points chauds en équilibrant la charge efficacement.

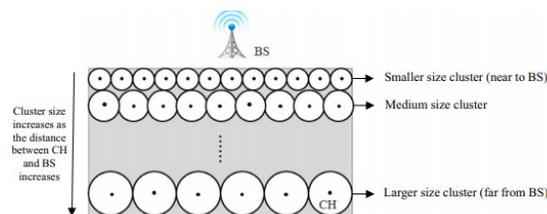


FIGURE 1.10 – Architecture de clustering inégal dans RCSF.

[9]

### 1.6.1 Objectifs du clustering

L'objectif principal du clustering est de réduire les communications. Cependant, cela englobe plusieurs autres objectifs, on cite entre autres :

- **Équilibrer la charge** : C'est-à-dire répartir équitablement les tâches les plus coûteuses dans le réseau afin d'éviter des points de congestion ou une

consommation de ressources déséquilibrée entre les nœuds du réseau[53].

- **Prolonger la durée de vie du réseau** : Le clustering vise à retarder l'épuisement des ressources des nœuds du réseau par la réduction des communications [7] [55].
- **Optimiser la bande passante** : Le clustering vise à optimiser l'utilisation de la bande passante du réseau et ce, en minimisant les communications et en évitant les duplications de messages et les retransmissions inutiles [18].
- **Assurer une qualité de service (QoS)** : Les ruptures de liens de communication provoquées par la mobilité des nœuds ou d'autres fautes transitoires entraînent un arrêt momentané du service fourni par l'applicatif du réseau. Toutefois, il est primordial d'assurer une qualité de service malgré ces ruptures, afin d'assurer le service requis [18].

## 1.7 Conclusion

Dans ce premier chapitre, nous avons d'abord présenté les deux classes de réseaux mobiles sans fil avec et sans infrastructures, ensuite nous avons vu les réseaux de capteurs sans fil ainsi que ses caractéristiques et ses domaines d'utilisation. Enfin, nous avons défini la notion de clustering et exposé ses objectifs.

# État de l'art des protocoles de routages dans les RCSFs

## 2.1 Introduction

Dans les RCSFs, le routage est un processus de sélection des chemins pour transmettre les données d'un nœud source à un nœud destination appelé Station de Base (SB) et ce, afin d'assurer la connectivité de tout le réseau. Chaque nœud qui compose le réseau peut participer au routage de données directement à la SB, et dans le cas où un nœud n'est pas capable de transmettre les données de manière directe, il est possible d'utiliser le routage multi-sauts en utilisant des nœuds intermédiaires. Cependant, les RCSFs sont influencés par certains critères, on cite entre autres la limitation de la ressource énergétique disponible au niveau de chaque nœud, ainsi que la topologie qu'ils forment dans le réseau. C'est pourquoi il existe aujourd'hui maintes recherches sur le domaine des RCSFs dont l'objectif est d'assurer le prolongement de la durée de vie du réseau, et ce à travers la conception de protocoles de routage. Ainsi, la consommation d'énergie représente un défi majeur pour le routage de données. De ce fait, le développement de nouveaux protocoles de routage dans les RCSFs est essentiel. Ces protocoles de routage doivent permettre l'établissement des routes pour le routage des données dans le réseau d'un nœud à un autre jusqu'à arriver au SB, tout en optimisant la consommation d'énergie de manière efficace. Dans ce chapitre, nous discuterons les grandes orientations en matière de conservation de l'énergie dans les réseaux de capteurs et nous porterons une attention particulière au clustering qui est l'une des solutions les plus prometteuses de conservation d'énergie.

## 2.2 Algorithmes de clustering conçus pour les RCSFs

### 2.2.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [32] est un protocole de routage hiérarchique qui s'exécute en rounds où chaque round se compose de deux phases :

1. Phase de construction(Set-up phase).
2. Phase de communication(Steady-state phase).

• **Fonctionnement du protocole LEACH : Phase de construction (Set-up phase) :**

Au départ, chaque nœud  $n$  choisi un nombre aléatoire entre 0 et 1 et le compare avec le seuil  $T(n)$ . Si le nombre choisi est inférieur au seuil  $T(n)$  donc le nœud  $n$  devient CH pour le round courant sinon il reste un nœud membre (CM).  $T(n)$  est défini comme suit :

$$T(n) = \begin{cases} \frac{P}{1-P \times (r \bmod \frac{1}{P})} & \text{si } n \in G, \\ 0 & \text{sinon,} \end{cases} \quad (2.1)$$

$P$  : le pourcentage désiré le CH pendant 1 round par exemple [5%, 10%].

$r$  : numéro de round courant.

$G$  : l'ensemble des nœuds qui n'ont pas été élu comme CH dans 1/p dernier round.

Les CMs doivent garder leurs récepteurs allumés pour recevoir les annonces et les notifications des CHs. Ces notifications sont envoyées par des signaux radio, en se basant sur la force de ce signal, chaque CM choisit à quel cluster appartenir pour ce round( choisir le plus fort). Si y'a une égalité lors de la comparaison des signaux, les CMs choisissent leur CH aléatoirement. Après la création des clusters, les CHs vont créer un calendrier TDMA pour indiquer pour chaque CM son temps alloué pour la transmission de données. le CH

## Chapitre 2. Etat de l'art sur les protocoles de routages dans les RCSFs19

---

envoi ce calendrier aux CMs. **Phase de communication (Steady-state)** : Une fois les clusters sont créés, et le Shedule TDMA est fixé, la transmission des données commence :

- 1 Les récepteurs de CMs peuvent ne pas être allumés, autrement dis la radio sera désactivée juste si y aura des données à envoyer et cela nous permettra de préserver beaucoup d'énergie.
- 2 Un CH doit toujours garder son récepteur allumé pour la réception des informations .
- 3 Une fois toutes les données ont été reçues ; CH va compresser les données en un seul signal et l'envoi à la SB [32]

---

### Algorithm 1 LEACH

---

- 1: **Entrées** :  $p, r$
- 2: **Sorties** :  $T_n$
- 3: choisir un nombre aléatoire  $x$  entre 0 et 1
- 4: Calculer  $T_n$

$$T(n) = \begin{cases} \frac{P}{1 - P \times (r \bmod \frac{1}{P})} & \text{si } n \in G, \\ 0 & \text{sinon,} \end{cases} \quad (2.2)$$

- 5: **if** ( $x < T_n$ ) **then**
  - 6:   le nœud devient CH et envoie un message d'avertissement aux autres nœuds
  - 7:   laisser son récepteur allume pour recevoir les demandes d'appartenance par les CMs
  - 8:   créer le calendrier TDMA
  - 9:   envoyer le calendrier aux CMs
  - 10:   laisser le récepteur allume pour recevoir les données
  - 11:   agréger les données
  - 12:   envoyer a la station de base
  - 13: **else**
  - 14:   laisser les récepteurs allume pour recevoir le message d'avertissement
  - 15:   décider a quel cluster appartenir en se basant sur la force du signal du message d'avertissement
  - 16:   informer le CH de ce cluster en envoyant un message
  - 17:   éteignez son récepteur jusqu'à son temps alloué pour la transmission
  - 18:   transmettre les données
  - 19: **end if**
-

- **Avantages et inconvénients de LEACH**

### Avantages

1. La rotation aléatoire des CHs permet d'économiser l'énergie et d'augmenter la durée de vie de réseaux.
2. La charge d'énergie est répartie sur l'ensemble des nœuds.
3. L'agrégation des données permet de réduire le taux de communications avec la SB.
4. L'auto-organisation des clusters est possible.

### Inconvénients

1. Les nœuds les plus éloignés du CH meurent rapidement par rapport aux plus proches.
2. Si tout les nœuds génèrent des nombres supérieur au seuil  $T(n)$  donc y'aura pas de CH.
3. LEACH ne prend pas en considération l'énergie résiduelle des nœuds.
4. LEACH ne résout pas le gaspillage de l'énergie entre les CHs et la SB.

### 2.2.2 Low Energy Adaptive Clustering Hierarchy Centralised (LEACH-C)

Le protocole LEACH ne garantit pas la distribution équitable des CHs dans le réseau, une version centralisée de LEACH appelée LEACH-C est donc proposée. Cette dernière permet de déterminer, à partir de la position exacte des nœuds, la configuration optimale pour minimiser l'énergie dépensée [31].

- **Fonctionnement du protocole LEACH-C** LEACH-C utilise la même phase de transmission que la version originale de LEACH. Durant la phase de formation de clusters la station de base reçoit de chaque nœud des informations concernant leurs localisations, et leurs réserves d'énergie. Ensuite, elle exécute un algorithme de formation de clusters en désignant les CHs selon leurs positions dans le réseau. Néanmoins, LEACH-C n'est pas adaptée aux réseaux denses [31] [10].

---

### Algorithm 2 LEACH-C

---

```
1: chaque nœud envoie sa localisation et son énergie résiduelle a la station de base
   (avec le GPS)
2: la station de base calcule l'énergie résiduelle moyenne de tout les nœuds
3: le nœud ayant l'énergie résiduelle inférieur a l'énergie résiduelle moyenne ne va
   pas devenir CH pour ce round
4: la station de base envoie les informations aux nœuds avec un message qui contient
   les IDs des CHs
5: if (l'ID du nœud est identique a l'ID reçu dans le message) then
6:   devenir CH pour ce round
7:   recevoir les données , les agréger et les transmettre a la station de base
8: else
9:   désactiver son récepteur jusqu'au temps alloué pour la transmission
10:  transmettre les données
11: end if
```

---

- **Avantages et inconvénients de LEACH-C**

#### Avantages

1. LEACH-C permet une diminution remarquable de la consommation énergétique.

#### Inconvénients

1. La version centralisée n'est pas adaptée aux réseaux de grande dimension.

### 2.2.3 Protocole d'Élection Stable SEP (LEACH Hétérogène)

Dans le protocole stable d'élection (SEP), il existe deux types de nœuds appelés nœuds normaux et nœuds avancés. Ces nœuds ont une énergie initiale différente. Les nœuds avancés sont plus souvent placés en CH que les nœuds normaux, car ces nœuds ont plus d'énergie que les nœuds normaux.

SEP super forme par rapport à LEACH, LEACH-C et *MOD\_LEACH*, car les nœuds avancés contribuent systématiquement à agir en tant que CH qui conduit à moins de dissipation d'énergie[50].

### 2.2.4 Multi Hop Low Energy Adaptive Clustering Hierarchy (MH-LEACH)

MH-LEACH utilise la même pratique que LEACH pour la sélection des CHs et la formation de cluster. Le rôle de CH est également le même, c'est-à-dire effectuer l'agrégation des paquets reçus.[22]. Mais, MH-LEACH possède plus de durée de vie, en effet, pendant la communication inter-cluster, CH envoie des paquets de données au CH situé à proximité de la SB au lieu de les envoyer directement à la SB et économise ainsi de l'énergie de CH. De plus, dans la communication intra-cluster, au lieu d'envoyer des paquets de données directement au CH, les CMs envoient les données aux nœuds voisins qui se trouvent sur le chemin de CH et les enregistrent. Le CH ou le nœud de cluster intermédiaire prend sa propre décision en fonction de son niveau d'énergie à accepter ou non les paquets de données. Ainsi, si un CH ou un nœud n'accepte pas les paquets de données, le nœud du capteur tente de localiser un autre CH selon l'entrée des nœuds dans la table de routage. Le protocole MH-LEACH fonctionne en deux phases : En phase 1, tous les CHs diffusent un message d'annonce et construisent leur table de routage en fonction du niveau de signal reçu. Ensuite, ils font leur chemin vers la SB via le CH le plus proche. En phase 2, chaque CH envoie ces routes initiales (à partir de la table de routage) à la SB.

### 2.2.5 Two Level Low Energy Adaptive Clustering Hierarchy (TL-LEACH)

TL-LEACH résout le problème de distribution d'énergie inégale de LEACH dû à la sélection aléatoire des CHs [52]. Auparavant, les CHs étaient sélectionnées de manière aléatoire, de sorte que différents CHs avaient une énergie résiduelle différente, ce qui entraîne la mort rapide des CHs loin de la SB. TL-LEACH résout ce problème en divisant la tâche de la collecte avec l'agrégation des données des nœuds au sein du cluster et la transmission des données collectées à la SB entre CH secondaire et primaire respectivement. Le CH secondaire est responsable de la collecte et agréger

les données des nœuds membres, puis les transférer au CH primaire ; le CH primaire est responsable de la transmission des données reçues de CH secondaire à la SB.

- **Avantages et inconvénients de TL-LEACH**

### Avantages

1. Ce protocole réduit l'énergie de transmission de données.
2. Les CHs meurent tôt par rapport à d'autres nœuds, loin de la station de base et TL-LEACH améliore l'efficacité énergétique en utilisant un CH comme nœud de relais entre les CHs de cluster [31].

### Inconvénients

1. Frais généraux supplémentaires pour l'élection des CHs secondaires et la formation des clusters.

## 2.2.6 Vice-cluster-head Low Energy Adaptive Clustering Hierarchy(V-LEACH)

Dans LEACH, en raison de la sélection aléatoire des CHs, il est fort possible qu'un CH sélectionnée n'ait pas suffisamment d'énergie pour transmettre des données à la SB. Ainsi, toutes les données contenues dans le CHs sont perdues [49]. V-LEACH fournit une solution à ce problème en créant un CH supplémentaire appelée vice-tête de cluster.

En V-LEACH les CHs appliquent la même fonction du LEACH, c'est-à-dire collecter, compresser et transmettre des données à la SB.

Si le CH meurt, le *vice\_CH* effectue le devoir de CH.

La sélection de CH et de CH secondaire est effectuée sur la base de la distance et de l'énergie résiduelle.

2.2.7 Tableau comparatif de LEACH et ses variantes

Protocole	Taille des clusters	Intra-communication	Inter-communication	Type de nœuds	Méthode	Élection des CHs
LEACH	Égale	1-saut	1-saut	nœud membre et CH	Distribuée	Déterministe
LEACH-C	Égale	1-saut	1-saut	nœud membre et CH	Centralisée	Déterminé par la SB
V-LEACH	Égale	1-saut	1-saut	nœud membre et CH et vice-CH	Distribuée	Déterministe
TL-LEACH	Égale	1-saut	multi-saut	nœud membre et CH et CH secondaire	Distribuée	Déterministe
MH-LEACH	Égale	1-saut	multi-saut	nœud membre et CH	Distribuée	Déterministe
H-LEACH	Égale	1-saut	1-saut	nœud membre et CH	Distribuée	Déterministe

Tableau 2.1 – Tableau comparatif du protocole LEACH et ses variantes.

## 2.3 Algorithmes de clustering inégaux conçus pour les RCSFs

### 2.3.1 Energy-Aware Distributed Unequal Clustering (EADUC)

Le protocole EADUC se divise en rounds, où chaque round contient une phase de configuration et une phase de transmission de données[56].

- **Fonctionnement de EADUC :**

1. **Phase de configuration de clusters :**

cette phase est subdivisée en trois sous phases pour former les clusters qui sont

- (a) ***Sous phase de collecte d'information des nœuds voisins :***

Chaque nœud envoie un *Node\_Msg* qui contient son Id et  $E_r$  dans un rayon  $r$ , au même temps, il reçoit un message *Node\_Msg* de ses nœuds voisins, et chaque nœud calcul l'énergie résiduelle moyenne  $E_a$  de ses nœuds voisins en appliquant la formule suivante :

$$E_a = \frac{1}{d} \sum_{i=1}^d S_i \cdot E_r \quad (2.3)$$

$d$  : c'est le nombre de nœuds voisins.

$S_i$  : c'est un nœud voisin.

$E_r$  : c'est l'énergie résiduelle de  $S_i$ .

Ensuite, chaque nœud calcule son temps d'attente  $t$  avant la diffusion du message *Head\_Msg* en appliquant la formule suivante :

$$t_i = \begin{cases} \frac{E_a}{E_r} \times T_2 \times V_r & \text{si } E_a \geq E_r, \\ T_2 \times V_r & \text{si } E_a < E_r, \end{cases} \quad (2.4)$$

$E_r$  : c'est l'énergie résiduelle du nœud.

$V_r$  : est une valeur réelle distribuée aléatoirement dans  $[0.9, 1]$  qui est introduite pour réduire la probabilité que deux nœuds envoient *Head\_Msg* en même temps.

$T2$  : le temps nécessaire pour la deuxième sous phase.

- (b) **Sous phase de compétition des CHs** : Dans cette phase, si le nœud ' $S_i$ ' ne reçoit pas de *Head\_Msg* lorsque le temps  $T_i$  se termine, il envoie le message *Head\_Msg* pour avertir qu'il sera un CH dans le rayon de compétition  $R_{c_i}$ , sinon il quitte la compétition. Afin de générer des clusters inégaux, chaque nœud doit calculer son propre rayon de compétition  $R_c$  en utilisant la formule ci-dessous :

$$R_c = [1 - \alpha \frac{d_{max} - d(S_i, SB)}{d_{max} - d_{min}} - \beta(1 - \frac{E_r}{E_{max}})] R_{max} \quad (2.5)$$

$d_{max}$  et  $d_{min}$  : la distance maximal et la distance minimale entre les nœuds du réseau et la SB.

$d(S_i, SB)$  : la distance entre le nœud  $S_i$  et la SB.

$\alpha$  et  $\beta$  : sont les facteurs pondérés dans  $[0, 1]$ .

$R_{max}$  : c'est la valeur maximale du rayon de compétition.

D'après les chercheurs, si  $d(S_i, BS)$  et  $E_r$  sont grand donc ils génèrent un grand RC qui signifie que les CHs avec une énergie résiduelle plus élevée et plus loin de la SB, contrôleront des zones de cluster plus grandes. D'une part, les CHs plus proches de la SB pourraient économiser de l'énergie pour la transmission de données. D'autre part, les CHs avec énergie résiduelle inférieure contrôlent les plus petits clusters dans l'ordre d'éviter leurs décès prématuré et prolonger la durée de vie du réseau .

- (c) **Sous phase de formation de cluster** : Chaque nœud non CH choisit le CH le plus proche et envoie le message *Join\_Msg* ( contient l'id et l'énergie résiduelle de ce nœud). Chaque CH crée une liste de programmation de nœud selon la réception du message *Join\_Msg*, et

envoi la liste de programme aux membres du cluster par diffusion du message *Schedule\_Msg*.

### 2. Phase de transmission de données

La phase de transmission de données doit être plus longue que la phase de configuration pour économiser la surcharge de l'algorithme et prolonger la durée de vie du réseau.

Communication intra-cluster : les membres du cluster collectent les données locales de l'environnement et envoient les données collectées aux CHs.

Communication inter-cluster : les CHs reçoivent et agrègent les données de leurs membres de cluster, puis envoient les données agrégées aux nœuds de saut suivants. On introduit une distance *DIST\_TH*, si  $d(S_i, SB) < DIST\_TH$  alors le nœud fait une transmission directe vers la station de base Sinon le nœud  $S_i$  construit une liste des nœuds candidats pour effectuer la transmissions à partir de ces nœuds voisins et calcule leurs paramètre  $E_{relay}$  suivant la formule suivante :

$$E_{relay} = d^2(S_i, S_j) + d^2(S_j, SB) \quad (2.6)$$

Le nœud choisit le nœud ayant l'énergie résiduelle la plus élevée et une valeur de  $E_{relay}$  la plus petite.

**Algorithm 3** EADUC

---

- 1: **Entrées** :  $V_r, E_r, T_2, d(S_i, S_j), d(S_j, SB), DIST\_TH$
- 2: **Sorties** :  $d, E_a, t_i, R_c, E_{relay}$ ,
- 3: envoyer  $Node\_Msg (S_i.id; S_i.E_r )$
- 4: a la reception de  $Node\_Msg$  calculer le nombre des noeuds voisins  $d$
- 5: Calculer  $E_a$

$$E_a = \frac{1}{d} \sum_{i=1}^d S_i \cdot E_r \quad (2.7)$$

- 6: Calculer  $t_i$

$$t_i = \begin{cases} \frac{E_a}{E_r} \times T_2 \times V_r & \text{si } E_a \geq E_r, \\ T_2 \times V_r & \text{si } E_a < E_r, \end{cases} \quad (2.8)$$

- 7: **if** ( $(t_i = 0)$  et (non réception de  $Head\_Msg$ )) **then**
- 8:     envoyer  $Head\_Msg$  dans le rayon  $R_c$

$$R_c = [1 - \alpha \frac{d_{max} - d(S_i, SB)}{d_{max} - d_{min}} - \beta(1 - \frac{E_r}{E_{max}})] R_{max} \quad (2.9)$$

- 9:     créer un calendrier TDMA
- 10:    envoyer le calendrier TDMA aux CMs
- 11:    agréger les données
- 12:    **if** ( $d(S_i, SB) < DIST\_TH$ ) **then**
- 13:        envoyer directement à la station de base
- 14:    **else**
- 15:        Calculer  $E_{relay}$

$$E_{relay} = d^2(S_i, S_j) + d^2(S_j, SB) \quad (2.10)$$

- 16:        Envoyer par le nœud routeur choisit
  - 17:    **end if**
  - 18: **else**
  - 19:     quitter la compétition
  - 20:     envoyer  $Join\_Msg$
  - 21:     collecter les données et les envoyer aux CHs
  - 22: **end if**
-

- **Avantages et inconvénients de EADUC**

### Avantages

1. Prolonger la durée de vie du réseau.
2. Il n'y a pas des nœuds isolés dans EADUC.

### Inconvénients

1. L'équilibrage de charge entre les CHs n'est pas assuré.
2. Les clusters sont non équilibrés.

### 2.3.2 Improved Energy-Aware Distributed Unequal Clustering ( Improved EADUC)

EADUC amélioré[26], diffère de EADUC dans l'expression de calcul de rayon de compétition  $R_c$ . EADUC se base sur la distance entre la SB et  $E_r$  de CH, Par contre en EADUC amélioré, ils ajoutent le nombre des nœuds voisins

- **Fonctionnement du protocole EADUC amélioré :**

La formule de calcul de rayon de compétition est la suivante :

$$R_c = [1 - \alpha \frac{d_{max} - d(s_i, SB)}{d_{max} - d_{min}} - \beta(1 - \frac{E_r}{E_{max}}) + \gamma(1 - \frac{S_i(nb)}{nb_{max}})]R_{max} \quad (2.11)$$

$d(s_i, SB)$  est la distance entre le nœud j et SB.

$S_i(nb)$  est le nombre des nœuds voisins par rapport au nœud j.

$nb_{max}$  est la valeur max des nœud voisins.

Après la terminaison de la sous-phase de compétition des CHs, la sous-phase de formation de cluster commence :

1. Dans cet phase, les CMs choisissent le CH le plus proche en envoyant le *Join\_Msg*.
2. Le CH envoi un calendrier TDMA par *Schedul\_Msg* pour les CMs pour qu'ils puissent envoyer des données, les CMs restent en état actif durant leurs slots, sinon se mettent en veille .

**Phase de transmission de données :**

La formule de calcul de  $E_{relay}$  est comme suit :

$$E_{relay} = \frac{s_j \cdot E_{jr} * s_j \cdot Count * ER_x * DM - (s_j \cdot Count + 1) * EDA * DM - ET_x \cdot DM}{E_{max}} \quad (2.12)$$

$s_j \cdot E_{jr}$  :c'est l'énergie résiduel courante pour le  $j^e$  nœud.

$ER_x$  :c'est le cout énergétique dans la réception des données à partir des CMs (la taille de paquets DM).

$EDA$  :le cout énergétique d'agrégation .

$ET_x$  :le cout énergétique de la transmission des paquets de données entre les CHs(à travers *relay\_dist* jusqu'à la SB).

$E_{max}$  :c'est l'énergie résiduelle initial(max).

### 2.3.3 Unequal Cluster-based Routing protocol (UCR)

- **Fonctionnement du protocole UCR :** Les opérations de ce protocole sont divisées en rounds, où chaque round contient les étapes suivante [17] :

1. **Sélection de CHs :**

Chaque nœud  $S_i$  génère un nombre aléatoire entre [0,1] Si le nombre choisis est inférieur au seuil T qui est un nombre aléatoire, le nœud  $S_i$  devient un *ch\_provisoire* . Chaque ch-provisoire diffuse un message *COMPTE\_HEAD\_MSG* qui contient (id,  $E_r$ ,  $R_i$ ) dans le rayon  $R_0$ , ou  $R_i$  se calcule avec la formule suivante :

$$R_i = [1 - c \frac{d_{max} - d(s_i, SB)}{d_{max} - d_{min}}] R_0 \quad (2.13)$$

Et il reçoit au même temps le même message des autres ch-provisoire. Donc chaque ch-provisoire conserve un ensemble  $S_{ch}$  de tous les *ch\_provisoire\_adjacent*.

*ch\_provisoire* construit son  $S_{ch}$  : Le *ch\_provisoire*  $S_j$  est un nœud adjacent de  $S_i$  si le  $S_j$  est dans le Rayon de compétition de  $S_i$  ou vice-versa c'est-à-dire Quand les nœuds adjacents  $S_j$  ont tous reçus le *COMPTE\_HEAD\_MSG* ;

Si  $d(S_i, S_j) < \max(R_i, R_j)$  alors Ajouter le nœud  $S_j$  dans  $S_i.S_{ch}$  Et donc la construction de  $S_{ch}$  du nœud  $S_i$  est terminée.

$S_i$  diffuse un *FINAL\_HEAD\_MSG* ( $id(S_i)$ ) pour informer les Ch-provisaires-adjacents qui est devenu un CH. et tous les nœuds qui ont reçus le message quittent la compétition donc les CHs sont construits.

**2. Formation de clusters :**

Après la construction des CHs , chaque Ch diffuse un *CH\_ADV\_MSG* à tous les nœuds membres ; et puis chaque CM choisit le CH le plus proche et renvoie au CH choisit un *JOIN\_CLUSTER\_MSG*. Ensuite chaque CH crée un *Schedule\_TDMA* et le transmet aux nœuds du cluster.

**3. Transmission de données :**

La transmission intra cluster similaire à LEACH [32].

La transmission inter cluster : utilisation de routage multi-sauts pour augmenter la durée de vie du réseau. Ils ont introduit un seuil TD-MAX dans ce protocole de routage multi-sauts ; Si  $d(S_i, (ch).SB) < TD-MAX$  ; La transmission des données directement à la SB ; Sinon, le nœud  $S_i$  qui est un CH envoie la donnée à un autre CH et ce dernier l'envoie à la SB et pour la sélection de CH distant (c'est-à-dire plus proche de la station de base) un ensemble de nœuds voisins  $R_{CH}$  du nœud CH  $S_i$  est défini comme suit :

$$S_i.R_{CH} = S_j | d(s_i, s_j) \leq xR_i, d(s_j, SB) < d(s_i, SB) \quad (2.14)$$

$x$  est l'entier minimum qui permet à  $S_i.R_{CH}$  de contenir au moins un élément (s'il n'existe pas un tel  $x$ , définissez  $S_i.R_{CH}$  comme un ensemble vide, donc y'aura pas de nœud relais). Le choix du nœud relais avec plus d'énergie résiduelle pourrait équilibrer la consommation d'énergie pour prolonger la durée de vie du réseau. D'un autre côté, la diminution du

## Chapitre 2. Etat de l'art sur les protocoles de routages dans les RCSFs<sup>32</sup>

coût énergétique par le paquet contribue également à la durée de vie du réseau. donc le nœud  $S_i$  choisit  $M$  nœuds ayant des coûts  $E_{relay}$  plus petits parmi les éléments de  $R_{CH}$ . Parmi ces  $M$  nœuds, un CH choisit le nœud ayant l'énergie la plus élevée comme CH suivant. et l'équation du calcul de  $E_{relay}$  est la suivante :

$$E_{relay} = d^2(s_i, s_j) + d^2(s_j, SB) \quad (2.15)$$

**Algorithm 4** UCR

---

```

1: Entrées :  $t, S_i.E_r, S_j.E_r, S_{id}, d(S_i, SB), TD\ MAX$ 
2: Sorties :  $S_i.S_{CH}, E_{relay}, S_{eligible}, S_i.R_{CH}$ 
3: générer un nombre aléatoire  $x$ 
4: if ( $x < t$ ) then
5:   devenir CH
6:   envoyer COMPETE_HEAD_MSG
7:   if ( $S_i.E_r > S_j.E_r$ ) then
8:     envoyer FINAL_HEAD_MSG
9:   end if
10:  recevoir QUIT_ELECTION_MSG
11:  enlever  $S_j$  de  $S_i.S_{CH}$ 
12:  envoyer un message CH_ADV_MSG
13:  créer un calendrier TDMA
14:  introduire un seuil TD MAX
15:  if ( $d(S_i, SB) < TD\ MAX$ ) then
16:    transmission directe a la station de base
17:  else
18:    construire l'ensemble des nœuds CH adjacents de ce nœud CH  $S_i.R_{CH}$ 
19:    calculer  $E_{relay}(S_i, S_j)$ 
20:    if (la table  $S_{eligible}$  n'est pas pleine ) then
21:      ajouter  $S_j$  a  $S_{eligible}$ 
22:    else
23:      trouver  $S_m$  qui assure :  $E_{relay}(S_i, S_m) = \max E_{relay}(S_i, S_n), S_n$  appartient
a  $S_{eligible}$ 
24:      if ( $E_{relay}(S_i, S_j) < E_{relay}(S_i, S_n)$ ) then
25:        remplacer  $S_m$  par  $S_j$  dans  $S_{eligible}$ 
26:      end if
27:    end if
28:    enlever  $S_j$  de  $R_{CH}$ 
29:    choisir un nœud dans  $S_{eligible}$  pour être un relais
30:  end if
31: else
32:  recevoir COMPETE_HEAD_MSG
33:  if ( $d(S_i, S_j) < \max(R_i, R_j)$ ) then
34:    ajouter  $S_j$  a  $S_i.S_{CH}$ 
35:  end if
36:  recevoir FINAL_HEAD_MSG
37:  if ( $S_j$  appartient a  $S_i.S_{CH}$ ) then
38:    envoyer QUIT_ELECTION_MSG
39:  end if
40:  choisir le CH le plus proche pour lui envoyer JOIN_CLUSTER_MSG
41:  recevoir le calendrier TDMA
42:  transmettre les données selon TDMA
43: end if

```

---

### 2.3.4 Energy and Coverage-aware Distributed Clustering Protocol (ECDC)

Le protocole de clustering distribué sensible à l'énergie et à la couverture (de zone, de point)[24].

Il s'agit d'un ensemble d'opérations qui sont divisés en rounds où chaque round contient :

**1. Phase de configuration de clusters :**

Cette phase est divisée en 3 sous-phases :

— **Collecte d'information dont la durée est T1 :**

Tous les nœuds sont en état plain (i.e sont des nœuds simple au début ni CH ni CM), Chaque nœud diffuse un *NODE\_MSG* (id(nœud), id(du point d'intérêt couvert par ce nœud),  $E_r$ ).dans son rayon de transmission  $R_c$ . A la réception des *NODE\_MSG* par les nœuds, ils calculent leurs importance de couverture  $Clp$  et l'énergie résiduelle moyenne de ses nœuds voisins  $E_r$ . L'énergie résiduelle moyenne des nœuds voisins de  $S_i$  est :

$$E_{ave}(S_i) = \frac{1}{|ND(S_i)|} \sum_{j=1}^{|ND(S_i)|} E_{cur}(S_j), \quad (2.16)$$

la formule de couverture  $Clp$  est :

$$CL_p(S_i) = \frac{|C(S_i)| - |O(S_i)|}{|U_{i=1}^n C(S_i)|}, \quad (2.17)$$

A la fin de la sous-phase de collecte d'information, chaque nœud calcule son temps d'attente  $t_i$  pour qu'il diffuse un message « *HEAD\_MSG* »

$$t_i = \begin{cases} \left[ \alpha \frac{E_{ave}(S_i)}{E_{cur}(S_i)} + (1 - \alpha)CL \right] T2Vr, & E_{cur}(S_i) \geq E_{ave}(S_i), \\ T2Vr, & E_{cur}(S_i) < E_{ave}(S_i), \end{cases} \quad (2.18)$$

— **Sélection de CH dont la durée est T2 :**

Dans cette sous-phase, Si le nœud  $S_i$  ne reçoit aucun *HEAD\_MSG* avant

que son  $t_i$  se termine, il diffuse un *HEAD\_MSG* dans le  $R_c$  pour affirmer les nœuds qu'il est CH ; Sinon, il abandonne la compétition et reste CM.

- **Formation de clusters dont la durée est T3** : Dans cette sous-phase, chaque CM envoie un *JOIN\_MSG* ( $id, E_r$ ) au CH plus proche, Puis chaque CH va créer un calendrier TDMA qui est incluse dans le *Schedule\_Msg*. Ce *Schedule\_Msg* indique pour chaque CM quand il peut transmettre ces données au CH pour réduire la consommation d'énergie.

## 2. Transmission de données

**Communication intra-cluster** : Les CMs détectent et collectent les données locaux de l'environnement et les envoient aux CHs.

**Communication inter-cluster** : il suit la topologie d'arbre de routage sur l'ensemble de CHs (une communication à saut multiple).

Dans cette transmission il existe 2 types de CH : Soit un CH ou un *CH\_enfant*. Un *CH\_enfant* envoie ses données directement à la SB. La sélection de CH enfant est la suivante : Déterminer un seuil de distance euclidienne *DIST\_TH*, Si la distance de CH  $< DIST\_TH$  ; Il envoie ses données directement à la SB et l'a définit comme son prochain saut Sinon il est un CH normal, il communique la SB via un routage multi-sauts.

**Construction de l'arbre de routage** : Elle est construite selon les CHs :

- A. Chaque CH diffuse un message de *ROUTE\_MSG* ( $id, E_r, Cl, d(ch, SB)$ ) dans son rayon de transmission  $R_r$  (pour assurer la connectivité de tous les CHs).
- B. Chaque CH choisit son prochain saut en fonction de *ROUTE\_MSG* reçues.
- C. Si ,  $d(CH_i, SB) < DIST\_TH$  Le  $CH_i$  choisit la SB comme son prochain saut Sinon, il choisit un  $CH_j$  (voisin) dont  $ER_j > ER_i$  et  $CL_i > CH_j$  et  $CH_j$  n'est pas plus éloigné de la SB, comme son prochain saut, Donc la formule de relais :

$$relay(S_i, S_j) = \beta \frac{E_{cur}(S_j)}{E_{max}(S_j)} + (1 - \alpha)CL, \quad (2.19)$$

**Algorithm 5** ECDC

---

- 1: **Entrées** :  $C(S_i)$ ,  $ND(S_i)$ ,  $E_{cur}$ ,  $S_{id}$ ,  $d(CH_i, SB)$ ,  $DIST\_TH$ ,  $O(S_i)$ ,  $\alpha$ ,  $T2$ ,  $Vr$
- 2: **Sorties** :  $CL_p$ ,  $E_{ave}$ ,  $t_i$ ,  $E_{relay}$
- 3: envoyer  $Node\_Msg$
- 4: calculer son importance de couverture  $CL_p$

$$CL_p(S_i) = \frac{|C(S_i)| - |O(S_i)|}{|U_{i=1}^n C(S_i)|}, \quad (2.20)$$

- 5: Calculer  $E_{ave}$

$$E_{ave}(S_i) = \frac{1}{|ND(S_i)|} \sum_{j=1}^{|ND(S_i)|} E_{cur}(S_j), \quad (2.21)$$

- 6: Calculer  $t_i$

$$t_i = \begin{cases} \left[ \alpha \frac{E_{ave}(S_i)}{E_{cur}(S_i)} + (1 - \alpha)CL \right] T2Vr, & E_{cur}(S_i) \geq E_{ave}(S_i), \\ T2Vr, & E_{cur}(S_i) < E_{ave}(S_i), \end{cases} \quad (2.22)$$

- 7: **if** ( $(t_i = 0)$  et (non réception de  $Head\_Msg$ )) **then**
- 8:     envoyer  $Head\_Msg$
- 9:     recevoir  $Join\_Msg$
- 10:    créer un calendrier TDMA
- 11:    envoyer le calendrier TDMA aux CMs
- 12:    agréger les données
- 13:    envoyer  $Route\_Msg$  ( $id, Er, Cl, d(S_i, SB)$ )
- 14:    **if** ( $d(S_i, SB) < DIST\_TH$ ) **then**
- 15:        envoyer directement à la SB
- 16:    **else**
- 17:        Calculer  $E_{relay}$

$$relay(S_i, S_j) = \beta \frac{E_{cur}(S_j)}{E_{max}(S_j)} + (1 - \alpha)CL, \quad (2.23)$$

- 18:        Envoyer par le nœud routeur choisit
  - 19:    **end if**
  - 20: **else**
  - 21:     quitter la compétition
  - 22:     envoyer  $Join\_Msg$
  - 23:     collecter les données et les envoyer aux CHs
  - 24: **end if**
-

### 2.3.5 Energy Neutral Clustering protocol (ENC)

ENC utilise un nouveau mécanisme qui est le groupe de CHs (CHG) qui permet à un cluster d'utiliser plusieurs CHs pour partager une charge élevée de trafic [45].

- **Fonctionnement du protocole ENC :**

ENC est divisé en 3 phases :

1. **Initialisation** : le réseau doit être regroupé en  $K$  clusters, où la valeur de  $K$  est défini et contrôlé par l'utilisateur. Ainsi, plusieurs nœuds seront sélectionnés au hasard pour être des nœuds centraux de la génération initiale des clusters. Un nœud central (CN) est le nœud de capteur qui est situé au centre géographique de cluster engendré. Afin de sélectionner ces CNs, une  $P_k$  constante est affectée à chaque capteur, qui est calculé par :

$$P_k = \frac{K}{N}, \quad (2.24)$$

où  $P_k$  désigne la probabilité qu'un capteur soit sélectionné en tant que CN. Chaque capteur générera alors un nombre aléatoire compris entre 0 et 1. Un capteur deviendra un CN s'il génère un nombre inférieur à  $P_k$ .

Le reste des nœuds de capteur (qui génèrent des nombres supérieurs à  $P_k$ ) deviendront les membres initiaux du cluster ( $CM\_initial$ ).

2. **Formation de clusters** : Un CN commencera à diffuser un message «CLUSTER».

Un  $CM\_initial$  qui reçoit un seul message «CLUSTER» avant la date limitée TC, rejoindra le CN qui a envoyé ce message «CLUSTER». Un  $CM\_initial$  qui reçoit plus d'un message "CLUSTER" avant la fin du temps TC comparera la puissance du signal des messages "CLUSTER" reçus. Il choisira de rejoindre le CN à partir duquel la force du signal reçu est la plus forte. À partir de l'instant TC, les  $CM\_initial$  enverront des messages «JOIN» (contenant l'ID du nœud émetteur et récepteur; le champ «EnergyBudget»; le champ «DistanceToBS»; le champ «DistanceToCH») à une instance temporelle choisie au hasard avant l'échéance de TC + TJ. À la réception de ces messages «JOIN», un CN vérifiera s'il est le destinataire prévu de ce message en vérifiant la valeur sous le champ "ID

de cluster" de ce message "JOIN". Si ce CN est le destinataire prévu, il enregistrera les valeurs sous les champs contenus dans le message 'JOIN'. Sinon, ce dernier sera rejeté.

- 3. Finalisation :** Au moment  $TC + TJ$ , les messages 'JOIN' envoyés par tous les  $CM\_initial$  sont censés être reçus par les CNs correspondants, ce qui signifie qu'un CN pourra déterminer les  $CM\_initial$  qui se sont enregistrés auprès de lui-même. Et donc un cluster  $k$  estimera alors sa taille  $N_g^k$ . Ce CN choisira ensuite  $N_g^k - 1$   $CM\_initial$  qui sont plus proches de ce CN, pour former le CHG. Notez que ce CN deviendra également un nœud CHG. Les autres  $CM\_initial$  du cluster seront choisis en tant que  $CM\_final$ , qui transmettront leurs informations détectées au CHG. après avoir déterminé les rôles de tous les nœuds de capteur à l'intérieur du cluster, un CN diffusera un message SCHEDULE(CHG-Node ou  $CM\_final$ ) avant l'échéance de  $TC + TJ + TS$ . Lors de la réception de ce message, un capteur vérifie les informations contenues dans ce message et trouve son propre rôle. Ensuite, il choisi pour être un CM final allumera son émetteur-récepteur radio et transmettra ses informations détectées pendant la période de réveil spécifiée dans le message «SCHEDULE» et éteindra sa radio dans le cas contraire. Pour un capteur qui est choisi pour être un nœud CHG, il se réveillera et agira comme un CH actif pendant la période de réveil spécifiée dans le message 'SCHEDULE' et se mettra en veille dans le cas contraire.

### Algorithm 6 ENC

---

- 1: **Entrées** :  $K, N$
- 2: **Sorties** :  $P_k, N_g^k$
- 3: Calculer sa distance à la station de base
- 4: générer un nombre aléatoire  $P$  entre 0 et 1
- 5: Calculer  $P_k$

$$P_k = \frac{K}{N} \quad (2.25)$$

- 6: **if** ( $P < P_k$ ) **then**
  - 7:   devenir CN (Center Node)
  - 8:   **if** (Temps courant  $< T_c$ ) **then**
  - 9:     envoyer le message CLUSTER( $CN_i.id$ )
  - 10:   **else**
  - 11:     **if** ( $T_c < \text{Temps courant} < T_c + T_j$ ) **then**
  - 12:       CID = l'ID dans le message JOIN
  - 13:       **if** (monID=CID) **then**
  - 14:         enregistrer les informations contenues dans le message JOIN
  - 15:       **else**
  - 16:         rejeter le message JOIN
  - 17:       **end if**
  - 18:     **else**
  - 19:       **if** (Temps courant =  $T_c + T_j$ ) **then**
  - 20:         calculer la taille de CHG  $N_g^k$
  - 21:         choisir  $N_g^k - 1$  nœuds CM comme CHG (les plus proches de CN)
  - 22:         monRole= CHG
  - 23:         envoyer un message SCHEDULE
  - 24:       **end if**
  - 25:     **end if**
  - 26:   **end if**
  - 27: **else**
  - 28:   devenir un  $CM\_initial$
-

---

```
29:   if (Temps courant < Tc) then
30:       recevoir le message CLUSTER
31:       if (la force de signal de ce message CLUSTER est plus fort que les précédents messages CLUSTER) then
32:           monCh=l'ID dans le message CLUSTER
33:           calculer ma distance au CH madistaCH
34:       end if
35:   else
36:       if (Tc < Temps courant < Tc+Tj) then
37:           envoyer un message JOIN(monID, monCH, madistaCH, monER)
38:       else
39:           if (Temps courant= Tc+Tj) then
40:               recevoir un message SHEDULE
41:               if (Ce nœud est programmé en tant que CHG) then
42:                   monRole = CHG
43:               else
44:                   monRole = CM_final
45:                   envoyer les données à CHG
46:               end if
47:           end if
48:       end if
49:   end if
50: end if
51: monRole = CM_final
52: envoyer les données à CHG
```

---

### 2.3.6 Balanced Clustering Algorithm with Distributed Self-Organization (DSBCA)

- **Fonctionnement du protocole DSBCA :**

Le protocole DSBCA[38] est divisé en 3 phases :

1. **Phase de sélection de CH :** Le nœud déclencheur  $U_t$  calcule sa densité et sa distance de la station de base pour déterminer le rayon de cluster  $k$ , et devient le CH temporaire. Pour calculer  $k$  :

$$K = \text{floor}[\beta D(U_t)/D_k(U_t)] \quad (2.26)$$

où  $D_{(U)}$  est la distance entre la station de base et U.

$D_k(U)$  est la densité de connectivité du nœud U.

$\beta$  est un paramètre du capteur déterminés par les applications spécifiques des RCSFs, et floor est le calcul d'arrondi.

$D_{(U)}$  peut être calculé comme suit :

$$D_U = 10^{\frac{|RSSI-A|}{10-n}} \quad (2.27)$$

Dans cette phase, le nœud  $U_t$  déclenche le processus de mise en cluster et envoie des messages HELLO à ses voisins K-sauts [38]. Les voisins K-sauts utilisent la formule suivante pour calculer le poids respectif :

$$W(u) = \phi * P[D_k(u)] + \varphi * P\left[\frac{R_e(u)}{E(u)}\right] - \gamma * P[H(u)], \quad (2.28)$$

$$0 \leq \phi, \varphi, \gamma \leq 1, \gamma < \phi + \varphi < 1.$$

où  $\phi$ ,  $\varphi$ ,  $\gamma$  comme les facteurs d'effet sont définies par l'application spécifique,  $R_e(u)$  est l'énergie résiduelle du nœud u,  $E(u)$  est le l'énergie initiale du nœud u,  $H(u)$  est le temps du nœud u être élu en tant que CH. De cette façon, nous réduisons les perspectives de u être élu en tant que CH pour équilibrer la consommation globale d'énergie.

Puis le nœud avec le poids le plus élevé deviendra CH. A partir de là, les nœuds CH diffusent *Head\_message* pour ses voisins k-sauts pour les informer qu'il est devenue CH.

*Head\_message* inclut l'ID du nœud de tête de cluster (HID), l'ID du nœud d'envoi (SID) et le nombre de sauts du CH (HD).

Lorsqu'un nœud voisin reçoit *Head\_message*, même s'il se trouve déjà dans un cluster, il envoie *Join\_message* au CH pour demander à se joindre.

2. **Phase de construction des clusters** : DSBCA définit la taille de cluster. Le nombre des nœuds de cluster ne peut pas dépasser cette taille pour éviter la formation des grands clusters. Lorsque le CH reçoit *Join\_message* envoyé par le nœud ordinaire, il accepte un nouveau membre et mettre à

## Chapitre 2. Etat de l'art sur les protocoles de routages dans les RCSFs42

jour le nombre de nœuds de cluster si la taille de cluster est inférieure à la taille prédéfinie ou rejeter la demande. Si le nœud rejeté a déjà un CH, le processus de clustering cesse. Sinon, il trouve un autre cluster approprié à rejoindre.

3. **Phase de cycle** : Le cluster est stable pendant un certain temps jusqu'à ce que le processus de la réélection de CH est déclenchée en  $T(k)$ . Le CH rassemble les poids de tous les nœuds membres, puis sélectionne le nœud avec le poids le plus élevé en tant que CH suivant. De cette façon, les coûts de communication sont diminués.

- **Avantages et inconvénients de DSBCA**

### **Avantages**

1. Le DSBCA génère des clusters plus équilibrés en énergie donc l'énergie est réduite.
2. Il évite de créer des Clusters excessifs avec une grande quantité nœuds.

### **Inconvénients**

1. Un cluster peut entraîner une consommation d'énergie excessive dans la communication.
2. Si le nombre de nœuds dans un cluster venait à dépasser le seuil fixé, cela entraînerait un sur-coût supplémentaire et réduirait ainsi la durée de vie du réseau.

**Algorithm 7** DSBCA

---

- 1: **Entrées** :  $\beta, D(U_t), D_k, U_t, H(u), E(u), R_e(u), \phi, \varphi, \gamma$
- 2: **Sorties** :  $W(u)$
- 3: calculer  $K$

$$K = \text{floor}[\beta D(U_t)/D_k(U_t)] \quad (2.29)$$

- 4: envoyer un message HELLO pour ses voisins a K-sauts
- 5: calculer le poids

$$W(u) = \phi * P[D_k(u)] + \varphi * P\left[\frac{R_e(u)}{E(u)}\right] - \gamma * P[H(u)], \quad (2.30)$$

$$0 \leq \phi, \varphi, \gamma \leq 1, \gamma < \phi + \varphi < 1.$$

- 6: le nœud ayant le poids le plus élevé devient CH
  - 7: ce nœud envoie un message *Head\_message* [l'ID de CH(HID),l'ID de nœud destinataire(SID),le nombre des sauts de CH (HD)]à ses voisins à K-sauts pour les informer qu'il est CH
  - 8: lorsque un nœud reçoit *Head\_message* , même s'il est déjà dans un cluster , il envoie *JOIN\_MSG* au CH pour demander son appartenance au cluster
  - 9: les nœud qui ne reçoivent pas *Head\_message* dans un temps  $T(w)$ ( $T(w) < T(k)$ ) ,il sera lui même CH (parce que le HEAD-MSG est limite aux noeuds voisins a K-sauts )
  - 10: limiter la taille de cluster
  - 11: lorsque le CH reçoit *JOIN\_MSG* d'un nœud ordinaire , il va regarder la taille de cluster pour savoir s'il va accepter un nouveau membre ou pas
  - 12: le nœud qui a envoyé *JOIN\_MSG* a un CH et qu'il est plain (il refuse son appartenance) il va chercher un autre CH s'il ne possède pas deja un
  - 13: chaque nœud membre contient une table qui sauvegarde des informations de cluster (HID,HD,SID et autres informations)
  - 14: un nœud peut mettre a jour une table
  - 15: un processus de sélection de CH est déclenche a  $T(k)$  , le CH compare les poids de tous les membres , et choisit le nœud ayant le plus grand poids pour être CH
-

### 2.3.7 Energy-Efficient Intra-Cluster Routing Algorithm (EIR)

Le protocole EIR est proposé pour équilibrer la consommation d'énergie des nœuds membres NMs pendant un temps limité, ce qui conduit à l'amélioration de la couverture du réseau [21].

- **Fonctionnement de protocole EIR :**

Il est divisé en 3 phases dont :

1. **Calcul de distance optimal :** Dans cette phase, la distance optimale entre chaque NM et SM est déterminée de telle manière que la consommation d'énergie des NMs soit équilibrée. Cette phase est lancée lorsque SM reçoit les informations requises des NM.

L'équation suivante est le cas parfait pour chaque NM, qui conduit à équilibrer l'énergie restante du nième nœud avec NC :

$$d_{toMS}(i) = \sqrt[n]{\frac{\sum_{j=1}^X (d_{topos}(i, j))^n}{X}} \quad (2.31)$$

Cependant, cela ne peut pas toujours être satisfait. Par conséquent, la phase suivante vise à déterminer les lieux de séjour pour chaque NM [21].

2. **Positions de séjour exclusives de MS :** Dans cette phase de l'EIR proposé, les positions de séjour optimaux de MS sont déterminées exclusivement pour chaque NM sur la base de la distance optimale calculée entre NM et MS. A cet effet, pour chaque NM, les positions de séjour de MS sont exclusivement sélectionnées de manière à ce que l'énergie de chaque nœud soit équilibrée avec la consommation d'énergie des identifiés CN.
3. **Positions de séjour optimale de MS :** Dans cette phase, les positions de séjour optimaux de MS sont déterminés en fonction des lieux de séjour exclusifs de MS. Le but d'employer le MS est de résoudre la consommation d'énergie déséquilibrée de NMs, qui est le résultat des distances inégales

## Chapitre 2. Etat de l'art sur les protocoles de routages dans les RCSFs45

entre NMs et CH, les valeurs sont attribuées aux NM en fonction de leurs distances à leur CH local. En fait, les nœuds situés plus loin de CH ont une valeur plus élevée par rapport aux autres. Par conséquent, la valeur la plus élevée est attribuée au CN qui a la distance la plus élevée de CH (Valeur (CN) = 1). Ensuite, la valeur des autres nœuds membres seront affectés comme suit :

$$value(i) = \frac{d_{toCH}(i)}{d_{toCH}(CN)} \quad (2.32)$$

La valorisation des nœuds vise à générer les lieux de séjour. Par conséquent, à ce stade, GA évalue les emplacements (x, y) selon les positions de séjour exclusives optimisées en seconde phase et les valeurs attribuées aux NM. La fonction de fitness du GA proposée dans cette phase est l'équation suivante :

$$F = \min \left( \sum_{i=1}^m \left( value(i) \cdot \sum_{j=1}^X (d_{toBestpos}(i, j) - d_{toOptimalPos}(i, j))^2 \right) \right) \quad (2.33)$$

où  $m$  est le nombre de nœuds de capteurs situés sur le cluster.

$d_{toOptimalPos}$  est la distance entre NMs et les positions générées par GA dans cette phase.

$d_{toBestpos}$  désigne la distance entre le  $i$  ème MN et ses positions de séjour suggérées qui sont calculés dans la deuxième phase. De plus,  $X$  est le nombre de lieux de séjour dans un cluster.

---

**Algorithm 8** EIR

---

```
1: Entrées :  $E_r, S_{id}, m$ 
2: Sorties : CN,  $d_{toMS}(i), E1(i), F$ 
3: générer la position de MS
4: clusterID= l'id de cluster auquel il appartient le MS
5: MS envoie un paquet qui contient (clusterID , Pos(x,y))
6: for  $i$  allant de 1 a  $N$  do
7:   if (la position de  $i$  est dans le CluserID) then
8:      $i$  envoie un paquet d'informations ( $i.RE$  ,  $i.pos$ ) a MS
9:   end if
10: end for
11: CN = argmin (RES( $i$ ))
12: for  $i$  allant de 1 a  $m$  do
13:   calculer la distance au MS  $d_{toMS}(i)$ 
14:   calculer  $E1(i)$ 
15: end for
16: for  $i$  allant de 1 a  $m$  do
17:   calculer Bestpos( $i$ )
18: end for
19: for  $i$  allant de 1 a  $m$  do
20:   attribuer les valeurs aux nœuds
21: end for
22: calculer la position optimale F
```

---

### 2.3.8 A Cluster-Head Rotation Approach for Sensor Networks (LAR-CH)

LAR-CH est une approche de rotation de CH qui prend en compte la charge énergétique actuelle du CH pour estimer un seuil dynamique pour la rotation CH. Dans LAR-CH, chaque CH estime d'abord le nombre de rounds de communication  $CH_{ERound}$  qu'il peut terminer avec succès avant de mourir. Si la valeur de  $CH_{ERound}$  est supérieur à 1, le CH termine  $CH_{ERound} - 1$  rounds en tant que CH et ensuite il déclenche la rotation du CH. Si la valeur de  $CH_{ERound}$  est inférieur ou égal à 1, le CH déclenche la division de cluster [44].

• **Fonctionnement de protocole LAR-CH :**

Le protocole de clustering proposé comporte quatre phases :

1. **Formation initiale de clusters :** Initialement, tous les nœuds envoient leurs identifiants et informations de localisation à la SB qui divise ces nœuds en clusters et sélectionne les CHs en fonction des informations de localisation seulement, l'Er des nœuds n'est pas prise en compte car tous les nœuds commencent initialement par la même énergie et perdre très peu d'énergie pour envoyer leur identifiants et Informations de localisation à la SB. Le nœud à proximité du centre de gravité du cluster est choisi comme CH dans chaque cluster car il minimise la distance de la communication intra-cluster.

Ensuite, la SB diffuse ces informations à tous les nœuds du réseau. Si l'ID d'un nœud correspond à l'ID d'un des CH, le nœud prend la responsabilité du CH; autrement, il lit le calendrier TDMA pour la transmission de données et passe en mode veille jusqu'à ce que son seuil de transmission vient.

2. **Approche de rotation CH :** Chaque CH estime d'abord le nombre de rounds qu'il peut terminer avec succès avant de mourir. Pour cela, il mesure d'abord la charge énergétique pour un seul round de communication  $E_{L1R}$  en utilisant l'équation suivante :

$$E_{L1R} = CH_{(E, Rl, m-1)} + CH_{(E, Al, m)} + CH_{(E, Tl, d(SB))} \quad (2.34)$$

Où  $CH_{(E, Rl, m-1)}$  représente la consommation d'énergie de CH lors de la réception d'un paquet de données de  $l$  bits de  $(m - 1)$  CM nœuds.

$CH_{(E, Al, m)}$  représente la consommation d'énergie de CH en agréger son paquet de données de  $l$  bits avec les paquets reçus de membres associés.

$CH_{(E, Tl, d(SB))}$  représente la consommation d'énergie du CH en transmettant  $l$  bits agrégés de Long paquet de données à la SB.  $E_{L1R}$  est l'énergie totale requise par CH pour réussir un cycle de communication.

$d(SB)$  représente la distance CH à la SB.

$m$  représente le nombre total de nœuds dans le cluster.

Pour estimer le nombre attendu de rounds qu'il peut réussir avant de mourir, le CH utilise l'équation ci-dessous :

$$CH_{ERounds} = \left\lfloor \left( \frac{E_{RCH}}{E_{LIR}} \right) \right\rfloor \quad (2.35)$$

$CH_{ERounds}$  représente le nombre attendu de rounds qu'un CH peut réussir avant de mourir et  $E_{RCH}$  est l'énergie résiduelle du CH.

Après avoir estimé les  $CH_{ERounds}$ , si la valeur de  $CH_{ERounds}$  est supérieur à 1, CCH (CH Courant) termine  $CH_{ERounds} - 1$  rounds et puis il déclenche la rotation de CH. En rotation de CH, CCH sélectionne un NCH (nouveau CH) pour le prochain round en se basant sur l'Er et la distance intra-cluster.

Chaque membre du cluster envoie son état énergétique actuel et son emplacement au CH correspondant. CCH sélectionne un ensemble de membres de telle sorte que l'énergie résiduelle des membres sélectionnés est supérieure à l'énergie moyenne des membres du cluster. Le CCH sélectionne un nœud membre comme CH dans l'ensemble de CM tel que la distance au membre sélectionné soit moins que le reste. Après avoir sélectionné le NCH, CCH transfère son état actuel (informations sur les membres et calendrier TDMA) au NCH et devient un nœud normal du cluster. Sinon, CCH appelle à la division du cluster et divise la charge du cluster en deux sous-clusters.

3. ***Division de cluster*** : La division de cluster est un processus au niveau du cluster local exécuté par le CCH. L'algorithme de division de cluster se compose des quatre étapes suivantes :
  - CCH choisit deux membres m1 et m2 parmi ses CM associés comme CH de sorte que m1 et m2 aient la distance de séparation maximale.
  - Regroupez chacun des membres restants du cluster actuel avec m1 ou m2, selon le plus proche.
  - CCH informe à la fois m1 et m2 de leur statut CH et rejoint également le sous-cluster le plus proche.

— Les m1 et m2 entament leur métier de CH comme les autres CH du réseau.

Le coût de la division en clusters est le même que celui de la rotation de CH. Dans la division de cluster, au lieu de choisir un NCH comme dans la rotation de CH, le CCH sélectionne deux NCH et les autres membres rejoignent le CH le plus proche.

4. ***Transmission de données*** : La phase de transmission des données implique la transmission des données des capteurs au SB. Dans cette phase, le CH reçoit des données de CM associés selon le calendrier TDMA. Une fois les données reçues de tous les CM associés, CH agrège ses paquets de données avec les paquets reçus et envoie le paquet agrégé à la SB. Le protocole LAR-CH utilise CDMA pour lutter contre les interférences radio causées par les clusters voisins.

---

**Algorithm 9** LAR-CH

---

- 1: **Entrées** :  $E_{R-CH}$ ,  $E_{L1R}$
- 2: **Sorties** :  $CH_{ERounds}$
- 3: envoyer son ID et ses informations de localisation à la station de base
- 4: la station de base divise le réseau en clusters et sélectionne dans chaque cluster un nœud CH
- 5: la station de base envoie les informations de division de réseau et les CH aux nœuds de réseau
- 6: **if** (l'ID de nœud est celui choisit comme CH) **then**
- 7:     être CH
- 8:     calculer  $CH_{ERounds}$  (c'est le nombre des rends qui peut accomplir avant que son energie se termine

$$CH_{ERounds} = \left\lfloor \left( \frac{E_{R-CH}}{E_{L1R}} \right) \right\rfloor \quad (2.36)$$

- 9:     **if** ( $CH_{ERounds} > 1$ ) **then**
  - 10:         terminer les  $CH_{ERounds} - 1$  tours
  - 11:         déclencher la rotation de CH.
  - 12:         CCH sélectionne le NCH en se basant sur l'énergie résiduelle
  - 13:         CCH envoie ses informations courantes (les informations des CMs et TDMA ) à NCH et il devient CM
  - 14:     **else**
  - 15:         CCH choisit deux CMs m1 et m2 ayant une distance de séparation maximale
  - 16:         grouper les CMs avec m1 ou m2 (avec le plus proche)
  - 17:         CCH informe m1 et m2 de leurs nouveaux états CH et rejoint l'un des sous-cluster (le plus proche)
  - 18:         m1 et m2 commence leurs travail en tant que CH
  - 19:     **end if**
  - 20: **else**
  - 21:     lire le TDMA pour savoir son temps alloue pour la transmission de données et se mettre en veille
  - 22:     transmettre les donnees vers les CHs dans le temps alloue pour la transmission
  - 23: **end if**
-

### 2.3.9 Probability Driven Unequal Clustering Mechanism for WSN (PRODUCE)

PRODUCE est un algorithme de clustering qui élimine le problème de point chaud. Il maximise la durée de vie du réseau et le temps de couverture dans ce dernier. Ce protocole utilise des probabilités pour la construction des clusters de tailles inégaux et la géométrie stochastique pour le routage inter-cluster. Le clustering inégal organise les clusters de plus petite taille près de SB et de plus grande taille plus loin de SB. Cela conduit à ce que les CHs proches de la SB se concentrent davantage sur leurs rôles en tant que relais et que les CHs plus éloignés des SB peuvent se concentrer davantage sur communication intra-cluster qui se traduit par l'élimination de problème de point chaud. Ce protocole de clustering inégal basé sur les probabilités équilibre les résultats de consommation d'énergie et maximise la durée de vie du réseau et le temps de couverture, en particulier dans le réseau où la densité des nœuds est élevée [36].

### 2.3.10 Energy Driven Unequal Clustering (EDUC)

EDUC est un algorithme distribué qui réduit la consommation d'énergie et évite le problème de point chaud dans les RCSFs hétérogènes. Cet algorithme gère efficacement la consommation d'énergie des nœuds dans un cluster afin de réduire l'épuisement énergétique.

EDUC comporte deux phases : la phase de construction des clusters et la phase de collecte de données. La phase de construction du cluster comprend la sous phase de compétition CH et la sous phase de formation du cluster. Chaque nœud ne sert de CH qu'une seule fois pendant toute la durée de vie du réseau. Les CH préparent le calendrier TDMA pour leurs membres du cluster pour éviter les collisions intra-cluster. Les CH sont élus au hasard et le niveau d'énergie est calculé avec précision pour la rotation du CH. Les CH transmettent les données directement à la SB [57].

### **2.3.11 Energy Efficient Unequal Clustering (EEUC)**

EEUC est une approche hybride largement utilisée pour l'application de collecte de données dans les RCSFs. Pour éviter les problèmes de points chauds, le réseau est divisé en clusters de différentes tailles, et le routage à sauts multiples est impliqué dans la transmission des données vers la SB. La taille du cluster plus proche de SB est plus petit, ce qui réduit la consommation d'énergie du trafic intra-cluster et économiser plus d'énergie pour le routage multi-saut, La taille du cluster est directement proportionnelle à la distance de SB. L'EEUC est une méthode distribuée où les CHs sont élus par compétition localisée et les CHs finaux sont choisis en fonction de leurs énergies résiduelles [36].

### **2.3.12 Unequal Low Energy Adaptive Clustering Hierarchy (Unequal LEACH)**

LEACH c'est le premier algorithme de clustering développé qui élit CH en fonction de la probabilité. En raison de difficultés pratiques dans LEACH, LEACH amélioré avec une phase de configuration plus pratique est proposé. L'amélioration de LEACH construit des clusters basé sur la pondération adaptative à la demande (AOW) qui est un compromis entre l'énergie résiduelle et l'énergie totale et la plage de concurrence. Les clusters sont de tailles inégales et les CHs transmettent directement les données à la SB sans aucun nœud intermédiaire. La méthode Round Robin est utilisée pour la sélection de CH et la tranche de temps round née est prédéterminée. En phase d'état stable, les membres du cluster utilisent le calendrier TDMA pour transmettre des données au CH et CH utilise le calendrier CSMA pour transmettre les données reçues et agrégées à la SB [46].

### **2.3.13 Multi-hop Routing Protocol with Unequal Clustering (MRPUC)**

Cette approche distribuée vise à égaliser la consommation d'énergie dans tous les nœuds en cas de problème de point chaud et à transmettre les données à la SB

avec des relais pour réduire la consommation d'énergie. MRPUC sélectionne le noeud avec plus d'énergie résiduelle comme CH et les nœuds rejoignent le cluster en tant que membres du cluster dans lesquels le CH contient l'énergie résiduelle maximale et une distance moindre à la SB [23].

### **2.3.14 Partition Energy Balanced and Efficient Clustering Scheme (PEBECS)**

PEBECS résout le problème de point chaud et répartit uniformément la charge tout au long du réseau. La sélection de CH se fait par un algorithme heuristique basé sur l'énergie résiduelle, le degré du nœud et la distance à la SB. Un algorithme de compétition des CH est introduit pour attribuer le nombre de membres du cluster dans chaque cluster. Le CH dans la partition près de SB est plus petit en taille pour réduire le trafic intra-cluster et économiser de l'énergie pour le trafic inter-cluster. Il équilibre efficacement la consommation d'énergie, améliore l'involutivité et allonge la durée de vie du réseau significativement [36].

### **2.3.15 Energy Aware Fuzzy Unequal Clustering algorithm (EAUCF)**

EAUCF est une approche distribuée utilisée pour allonger la durée de vie du réseau comme EEUC. Dans EAUCF, les CHs provisoires sont choisis au hasard et l'énergie résiduelle est utilisée pour élire les CHs finaux. L'EAUCF se concentre principalement sur la détermination du rayon de compétition des CH pour répartir uniformément la charge, ce qui permet d'éliminer le problème des points chauds [3].

### **2.3.16 Improved Fuzzy Unequal Clustering algorithm (IFUC)**

IFUC est une approche distribuée qui réduit la consommation d'énergie. Il évite les problèmes de points chauds et allonge également la durée de vie du réseau. La logique floue est utilisée pour élire les CHs et déterminer le rayon de cluster.

Si un CH provisoire ne trouve pas un autre CH dans sa plage , il est sélectionné comme CH final [42].

### 2.3.17 Energy Balanced Unequal Clustering for Wireless Sensor Networks (EBUC)

EBUC est un protocole de clustering inégal centralisé couramment utilisé dans les applications de collecte de données périodiques. Il utilise l'algorithme PSO pour sélectionner les CHs candidats et divise le réseau en clusters de différentes tailles ; Le protocole de routage multi-sauts est basé sur un algorithme gourmand ; l'énergie résiduelle et la distance à la SB sont utilisées pour choisir un nœud relais. Cet algorithme réduit le taux de nœuds morts et allonge la durée de vie du réseau [23].

### 2.3.18 Genetic Algorithm based Energy-Efficient Adaptive Clustering Hierarchical Protocol (GAEEP)

GAEEP utilise l'algorithme génétique (GA) pour déterminer le nombre et la position des CH à réduire la consommation d'énergie. Toute l'opération subit de nombreux rounds ; chaque round se compose de deux phases à savoir la phase de configuration et phase d'état stable. Dans la phase de configuration, SB exécute GA et détermine le nombre optimal de CH et la position de CH. Lorsqu'un nœud est situé beaucoup plus près de SB que de tout CH, le nœud transmet directement les données à la SB. Chaque CH utilise la planification TDMA et attribue des créneaux aux membres de son cluster pour éviter les collisions dues à communication intra-cluster. Pour réduire les collisions entre clusters, les CHs utilisent le code CDMA. GAEEP c'est plus économe en énergie et fiable que LEACH, SEP, ERP, LEACH-GA et DEU dans les réseaux homogènes et hétérogènes [2].

### 2.3.19 Energy Degree Distance Unequal Clustering Algorithm (EDDUCA)

L'algorithme EEDUCA est proposé pour réduire la consommation d'énergie. Il améliore la durée de vie du réseau et élimine également le problème de point chaud. EDDUCA comporte trois phases : la phase de formation de clusters, Phase de sélection de CH et phase de transmission de données. En phase de formation de clusters, le triangle «Sierpinski» divise le réseau en clusters inégales. La taille des clusters est la même lorsque la distance entre CH et SB est la même. Dans la phase de sélection de CH, les CHs sont sélectionnés en fonction de l'énergie résiduelle, de degré du nœud et de la distance à la SB. Un poids de nœud est calculé et le nœud avec un poids minimal est élu CH. Chaque nœud transmet des données à CH et CH transmet à son cluster supérieur jusqu'à ce qu'il atteigne la SB. Il fonctionne bien et allonge la durée de vie du réseau [25].

### 2.3.20 Unequal Clustering and Connected Graph Routing Algorithm (UCCGRA)

UCCGRA est une approche distribuée qui améliore l'efficacité énergétique selon deux méthodes : élection des CHs et routage de cluster. Le schéma de vote est utilisé pour construire une taille inégale des clusters, les plus petits clusters sont construits près de SB pour réduire le trafic intra-cluster et pour l'élimination du problème de point chaud. La sélection du CH est principalement basée sur la topologie, l'énergie résiduelle et la puissance d'émission. Pour la communication multi-sautle routage basé sur un graphe connecté qui utilise l'emplacement géographique des nœuds. UCCGRA répartit efficacement la charge et réduit la consommation d'énergie. Il atteint une durée de vie maximale du réseau que UCRA, EEUC et HEED [54].

### 2.3.21 Unequal Clustering Scheme (UCS)

Le but du protocole UCS est de maximiser la durée de vie du réseau. La position des CH est prédéterminée avant le déploiement. Le cluster se compose de nœuds dans

la région de Voronoi avec SB au centre. UCS est un réseau à deux couches avec les clusters dans la même couche sont égaux dans la taille. Le CH transmet les données à BS via deux sauts. UCS atteint l'amélioration de 10 à 30% par rapport aux protocoles de clustering égaux [28].

### 2.3.22 Coverage-preserving clustering protocol (CPCP)

Dans CPCP, les nœuds de capteur communiquent directement avec leurs nœuds de CH élus, tandis que le routage des données des nœuds CHs vers la SB est effectué sur des chemins à sauts multiples à l'aide des capteurs

- Fonctionnement de protocole CPCP :

Le protocole CPCP [51] se compose de six phases comme décrit ci-dessous :

#### 1. *Phase de mise à jour des informations*

La première phase du CPCP est réservée à la mise à jour des informations sur les énergies restantes des nœuds. Chaque nœud de capteur diffuse un paquet de mise à jour avec des informations sur son énergie restante à tous ses voisins dans la plage  $R_{sense}$ . Afin de réduire les collisions de paquets, les nœuds utilisent des interruptions aléatoires avant d'envoyer les paquets de mise à jour. Lors de la réception des informations de mise à jour de tous les voisins, chaque nœud calcule son coût en fonction de la couverture ( $C_{mw}$  ou  $C_{ws}$ ). En supposant que les nœuds de capteur sont statiques, les nœuds voisins doivent échanger leurs informations de localisation une seule fois, au début de la durée de vie du réseau. Si le coût de redondance de la couverture  $C_{cc}$  ou le coût sensible à l'énergie  $C_{ea}$  sont utilisés, cette phase de mise à jour des informations peut être ignorée, car ces mesures de coût ne dépendent pas des énergies restantes des nœuds voisins.

#### 2. *Phase d'élection des CHs*

Au début de cette phase, chaque capteur détermine son «temps d'activation» un temps proportionnel à son coût actuel (c.-à-d.  $C_{mw}$ ,  $C_{ws}$ ,  $C_{cc}$  ou  $C_{ea}$ ). Chaque capteur doit attendre l'expiration de son «temps d'activation», si pendant le «temps d'activation», un nœud n'entend pas de message d'annonce de tout autre nœud de capteur, à l'expiration de son

«temps d'activation», il se déclare comme un nouveau CH, en envoyant une annonce à tous les nœuds de la plage  $R_{cluster}$ . Le message d'annonce contient des informations sur l'emplacement du nouveau CH. Après avoir reçu ce dernier, tous les nœuds de capteur de la gamme  $R_{cluster}$  quittent la compétition. Chaque nœud de capteur conserve une table de tous les nœuds principaux de cluster à partir desquels il a reçu le message d'annonce jusqu'à présent, ainsi que la distance à chaque nœud principal de cluster. Ces informations sont utilisées ultérieurement par le nœud du capteur pour décider de son appartenance au cluster. Cela peut rarement arriver que deux nœuds de capteur avec les mêmes coûts et dans chaque la gamme  $R_{cluster}$  des autres se déclarent simultanément être de nouveaux CH. ce conflit peut être résolu en donnant la priorité au nœud avec l'énergie restante la plus élevée.

3. ***Phase de mise à jour de l'itinéraire*** Les nœuds principaux du cluster envoient leurs données sur des chemins à sauts multiples vers le récepteur. Pour obtenir ces itinéraires, le nœud récepteur génère d'abord un message de découverte d'itinéraire qui est diffusé à travers le réseau. Lors de la réception du message diffusé, chaque nœud capteur introduit un retard proportionnel à son coût avant de transmettre le message de découverte d'itinéraire aux nœuds de la plage  $R_{bc}$ . De cette façon, un message arrive à chaque nœud le long du chemin de coût minimum souhaité. Le coût cumulé du chemin de routage du puits vers le nœud obtenu dans cette phase est appelé coût de routage sensible à la couverture du nœud, comme décrit dans l'équation suivante :

$$C_{final}(S_i) = \sum_{S_j, S_k \in P(S_i, S_{dst})} C_{link}(S_j, S_k), \quad (2.37)$$

4. ***Phase de formation des clusters*** Dans la quatrième phase du CPCP, chaque nœud principal non cluster décide de rejoindre le nœud principal de cluster le plus proche. Les nœuds capteurs envoient de courts JOIN-messages à leurs nœuds de tête de cluster sélectionnés. Ces JOIN-messages servent à reconnaître qu'un nœuds deviendra membre du cluster pour la

prochaine ronde. Notez qu'il n'y a aucune restriction sur le nombre de membres du cluster, uniquement sur la zone du cluster, et donc tout les nœuds qui souhaitent rejoindre une CH, et se trouvent dans la plage de cluster  $R_{cluster}$ , peut le faire.

5. ***Phase d'activation du capteur*** Dans la cinquième phase, un sous-ensemble de nœuds de capteur est sélectionné pour effectuer la tâche de détection pour le prochain cycle, tandis que le reste des nœuds de capteur s'endorment. Les nœuds actifs sélectionnés fournissent une couverture complète sur le champ surveillé pendant ce cycle de communication. chaque nœud de capteur s'assigne un retard d'activation qui est proportionnel à son coût d'application actuel (c'est-à-dire  $C_{mw}$ ,  $C_{ws}$ ,  $C_{cc}$  ou  $C_{ea}$ ). De cette façon, les nœuds de capteurs à moindre coût ont de meilleures chances de devenir des capteurs actifs dans le prochain cycle de communication. Chaque nœud de capteur attend ensuite cette période de temps avant de décider s'il restera éveillé lors de la prochaine communication. En attendant l'expiration de son délai d'activation, le nœud du capteur peut recevoir les messages ACTIVATION de ses nœuds voisins, qui ont des retards d'activation plus faibles, s'ils décident de devenir actifs lors du prochain cycle de communication. Sinon, il s'éteint pour le cycle à venir. Tous les nœuds capteurs du réseau conjointement participer à la phase d'activation, quel que soit le cluster auquel ils appartiennent. Cela élimine l'activation redondante des nœuds de capteur aux frontières des clusters, ce qui peut se produire lorsque l'activation des nœuds est effectuée dans chaque cluster indépendamment.
6. ***Phase de communication des données*** Une fois les grappes formées et les capteurs actifs sélectionnés, la phase de communication des données commence là où les nœuds de capteurs actifs collectent périodiquement des données et les envoient aux nœuds CHs. Les nœuds CHs regroupent les données des membres du cluster et acheminent les paquets de données agrégés sur les chemins multi-niveau prédéterminés vers la SB.

---

**Algorithm 10** CPCP

---

- 1: chaque nœud envoie un paquet avec ses informations sur son énergie résiduelle vers ses nœuds voisins
  - 2: chaque nœud calcule son cout lié à la couverture
  - 3: chaque nœud détermine son temps d'activation "ACTIVATION TIME"
  - 4: attendre la fin du temps d'activation
  - 5: **if** (le nœud ne reçoit pas un message d'annonce) **then**
  - 6:     devenir CH en envoyant un message d'annonce dans un rayon  $R_{cluster}$
  - 7:     envoyer un message Route-Discovery dans le réseau
  - 8:     après la réception de ce message , chaque nœud ajoute son cout avant qu'il envoie le message pour obtenir le chemin avec le cout minimum souhaité
  - 9:     agréger les données et les envoyer vers la station de base
  - 10: **else**
  - 11:     le nœud va être exclu de la compétition
  - 12:     chaque nœud contient une table de tous les CHs qui a reçu de leurs part un message d'annonce avec leurs distances
  - 13:     chaque nœud choisit à joindre le CH le plus proche
  - 14:     chaque nœud a un délais d'activation proportionnel à son cout, les nœuds ayant des petits cout passent à l'état ACTIF
  - 15:     **if** (sa zone est complètement couverte) **then**
  - 16:         le nœud passe à l'état SOMMEIL
  - 17:     **else**
  - 18:         le nœud reste ACTIF
  - 19:     **end if**
  - 20:     collecter les données et les transmettre aux CH
  - 21: **end if**
-

- **Avantages de CPCP**

**Avantages :**

1. Ce protocole réduit l'énergie de transmission de données.
2. Augmente la durée de vie du réseau.

### 2.3.23 Hybrid Energy Efficient Distributed Clustering (HEED)

HEED est un protocole de routage hiérarchique, basé sur un algorithme de clustering distribué. Il permet de prolonger la durée de vie du réseau. La sélection des CHs est donc basée sur l'énergie résiduelle de chaque nœud et chaque nœud appartient à exactement un seul cluster et peut communiquer directement avec son CH [55].

- **Fonctionnement de protocole HEED :**

L'algorithme est divisé en trois phases :

1. ***Phase d'initialisation***

Le protocole définit d'abord un premier pourcentage de CHs auprès de tous les nœuds. La valeur de ce pourcentage  $C_{prob}$  est utilisée pour éviter la collision. Chaque nœud définit sa probabilité  $CH_{prob}$  de devenir un CH comme suit :

$$CH_{prob} = C_{prob} \times \frac{E_{residuel}}{E_{max}} \quad (2.38)$$

$E_{residuel}$  est l'énergie résiduelle.

$E_{max}$  est l'énergie maximale du nœud.

2. ***Phase de répétition***

Pendant cette phase, chaque nœud passe plusieurs itérations jusqu'à ce qu'il trouve le CH auquel il peut transmettre avec le minimum d'énergie. S'il n'est à la portée d'aucun CH, le nœud lui-même choisit d'être un CH et envoie un message d'annonce à ses voisins les informant du changement de statut. Enfin, chaque nœud double sa valeur  $CH_{prob}$  et va à la prochaine itération de cette phase. Il arrête l'exécution de cette phase lorsque

son  $CH_{prob}$  atteint 1. Par conséquent, il existe deux types de statuts que pourrait annoncer un nœud à ses voisins :

- (a) Statut provisoire : le nœud devient un CH provisoire si son  $CH_{prob}$  est inférieur à 1. Il peut devenir un nœud normal à une itération plus tard s'il trouve un CH atteignable avec un coût minimal.
- (b) Statut final : le nœud devient définitivement un CH si son  $CH_{prob}$  à 1.

### 3. *Phase de finalisation*

Pendant cette phase, chaque nœud prend une décision finale sur son statut. Il choisit soit un CH qui atteindra avec un coût minimal ou se prononce lui-même en tant que CH.

- **Avantages et inconvénients de HEED**

#### **Avantages**

1. prolonger la durée de vie du réseau en distribuant l'énergie de la communication et le nombre de CHs d'une façon uniforme et donc produire des clusters compact tout en minimisant la charge des messages de contrôle.
2. Afin de former les clusters, les nœuds ont uniquement besoin d'information locale (nœuds voisins).

#### **Inconvénients**

1. Ce protocole se focalise plutôt sur la prolongation de la durée de vie du réseau, plutôt que sur l'ensemble des besoins d'un réseau de capteurs.
2. HEED souffre de surcharge car il a besoin de plusieurs itérations pour former les clusters.

### **2.3.24 Unequal Hierarchical Energy Efficient Distributed Clustering (UHEED)**

Fondé sur les principes de HEED, UHEED est un protocole de clustering, qui crée des clusters à tailles inégales selon la distance entre un CH et la SB : plus le CH est loin de la SB, plus son rayon de compétition sera plus large, et ainsi la taille de des clusters sera plus grande comparé à celle des cluster formé près de la SB. L'objectif

principal de ce protocole est réduire la consommation de l'énergie au niveau des CHs qui se situe tout près de la SB. A la différence de HEED, UHEED utilise une formule pour calculer le rayon de compétition afin de créer des clusters de tailles inégales [20].

$$R_c(S_i) = \left[ 1 - c \frac{d_{max} - d(S_i, SB)}{d_{max} - d_{min}} \right] R_{comp}^0 \quad (2.39)$$

$R$  : c'est la valeur maximum du rayon de la concurrence.

- **Avantages et inconvénients de UHEED**

- Avantages**

1. Prolonger la durée de vie du réseau.
2. Construit des cluster de différents tailles en fonction de la distance du CH a la SB.

- Inconvénients**

1. Le même rayon de compétition utilisés par les CHs donc on aura le problèmes de point chaud.

### 2.3.25 Hybrid Unequal Clustering with Layering protocol (HUCL)

Il existe deux types de protocoles de clustering dans les RCSF : statiques et dynamiques. Tous deux présentent différents avantages mais aussi certains inconvénients, tel que déséquilibre énergétique pour le clustering statique et une surcharge due aux messages de contrôle pour le clustering dynamique. Pour optimiser la durée de vie du réseau, un nouveau protocole qui hybride les avantages des deux clustering est proposé : HUCL. [41].

- **Fonctionnement du protocole HUCL :**

- Le réseau dans HUCL est divisé en couche et dans chaque couche on trouve des clusters de tailles inégales selon la distance des nœuds par rapport à la SB (distance minimale implique un clusters de taille petite) c'est à dire la SB diffuse un message 'HELLO' avec un niveau de puissance minimal

donc les nœuds recevant ce message forment la couche 1 et ces derniers calculent leur distance par rapport à la SB en utilisant la force du signal reçu. Ensuite, SB augmente le niveau de puissance et diffuse à nouveau le message 'HELLO' pour former la couche 2 et ainsi de suite.

- les rounds du HUCL sont divisés en deux phases, la phase d'installation qui est à son round divisée en cinq sous-phases et la phase de transmission.

### **Phase d'installation**

1. Découverte des nœuds voisins :

Chaque nœud diffuse un  $Neighbor_{Msg}$  aux autres nœuds selon son rayon  $r$  et il reçoit au même temps le même message ; puis il met à jour sa table des nœuds voisins. Ensuite il calcule un temps d'attente aléatoire  $Tw(S_i)$  selon l'équation ci-dessous :

$$Tw(S_i) = \frac{E_{init}(S_i)}{E_{rem}(S_i)} \times \frac{1}{\alpha|NL(S_i)|} \times T_2 \times R, \quad (2.40)$$

$E_{init}$  est l'énergie initiale.

$E_{rem}$  est l'énergie résiduelle du nœud  $S_i$ .

$|NL(S_i)|$  est le nombre de voisins du nœud  $S_i$ .

$R$  est un nombre aléatoire compris entre 0, 1 et 0, 2.

$\alpha$  est le nombre de fois qu'un nœud élu comme CH.

$T_2$  : le temps nécessaire pour la deuxième sous phase.

2. Compétition des CHs :

Si le  $Tw(S_i)$  de nœud  $S_i$  est expiré, et ce dernier n'a reçu aucun  $HEAD - Message$  ; il s'annonce qu'il est un CH en diffusant le  $HEAD_{Message}$  selon le rayon  $R_c$  calculer selon l'équation ci-dessous :

$$R_c(S_i) = \left[1 - c \frac{d_{max} - d(S_i, SB)}{d_{max} - d_{min}}\right] RL_{max} \quad (2.41)$$

$RL_{max}$  est le rayon maximal de compétition dépendant de la couche.

$d_{max}$  et  $d_{min}$  sont la distance maximale et minimale entre la SB et les nœuds.

$C$  est un facteur compris entre 0 et 1.

Si non, il devient un CM et met à jour sa table des CHs pour trouver le

CH le plus proche.

3. Formation de clusters :

une fois les CMs ont mis à jour leur table des CH, ils choisissent le CH le plus proche en lui envoyant un *Join – Msg* en suite à la réception *Join – Msg* les CHs mettent à jour leur liste des CMs.

4. Optimisation des clusters :

dans cette étape les clusters sans membres (CH seul) et les clusters à un seul membre (CH + un nœud membre) sont supprimés et se combinent avec les clusters les plus proches.

5. Construction du chemin de transmission vers la SB :

Dans cette partie, chaque CH construit le chemin de transmission des données vers la SB comme suit :

Tous les CHs de la couche 1 diffusent *Route – Msg* à la couche suivante jusqu'à ce que le message arrive à la couche la plus proche de la SB. Ensuite, un *Route – Reply – Msg* sera diffusé dans le sens inverse.

### **Phase de transmission des données**

Dans cette phase, les CHs diffusent une planification TDMA à leurs membres. Chaque CH agrège les données collectées par les membres de son cluster et les transmet au CH de la couche supérieure, le CH de la couche supérieure reçoit les données de l'autre canal et vérifie si les données sont déjà compressées. Si oui, il transmet les données à la couche supérieur suivante. Dans le cas contraire, il effectue une compression de données dans le réseau et transmet les données compressées à la couche supérieure.

### **2.3.26 Tableau comparatif des protocoles de clustering in-égales**

Protocole	Taille des clusters	Intra-communication	Inter-communication	Type de noeuds	Méthode	Election des CHs	Objectif
EADUC	Égale	1-saut	M-sauts	Ho/Hé	Distribuée	déterministe	Max la durée de vie
EADUCA	Égale	1-saut	M-sauts	Ho/Hé	Distribuée	déterministe	Max la durée de vie
UCR	Inégale	1-saut	M-sauts	Ho	Distribuée	déterministe	Max la durée de vie +Eq
ECDC	Égale	1-saut	M-sauts	Ho	Distribuée	déterministe	Max la durée de vie +Eq
ENC	Égale	1-saut	M-sauts	Ho	Distribuée	Hybride	Max la durée de vie
DSBCA	Inégale	M-sauts	M-sauts	Ho	Distribuée	Hybride	Max la durée de vie +Eq
EIR	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
LAR-CH	Égale	1-saut	M-saut	Ho	Distribuée	déterministe	Max la durée de vie
PRODUCE	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
EDUC	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie
EEUC	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie
EAUCF	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie+Eq
ULEACH	Inégale	1-saut	M-saut	Ho	Distribuée	Déterministe	Max la durée de vie
MRPUC	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
PEBECs	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
IFUC	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
EBUC	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
GAEEP	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
EDDUCA	Inégale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
UCCGRA	Inégale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie +Eq
UCS	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie
CPCP	Égale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie + Eq
HEED	Égale	1-saut	M-sauts	Ho	Distribuée	Hybride	Max la durée de vie
UHEED	Inégale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie
HUCL	Inégale	1-saut	M-saut	Ho	Distribuée	Hybride	Max la durée de vie + Eq

Tableau 2.2 – Tableau comparatif des protocoles de clustering inégaux.  
 Hé :Hétérogène. Ho :Homogène. M-saut :Multi-sauts. Eq :Équilibrage de charge

## **2.4 Conclusion**

Ce chapitre nous a permis de voir quelques protocoles de routage assurant l'efficacité énergétique dans les réseaux de capteurs, leur étude nous a suscité à les comparer. Dans le chapitre suivant, nous allons introduire notre protocole de routage hiérarchique dans les RCSFs dont appelé le protocole de clustering inégale (Unequal Clustering Protocol (UCP)).

# Le protocole UCP (Unequal Clustering Protocol)

## 3.1 Introduction

L'un des défis rencontrés lors de la conception des protocoles de routage, est le prolongement de la durée de vie du réseau ainsi que l'économie d'énergie. Dans ce chapitre nous proposons un protocole de routage hiérarchique (UCP) pour un réseau de capteurs sans fil qui assure le prolongement de la durée de vie du réseau. Le protocole proposé est une hybridation entre les protocoles HUCL, AEADUC et EADUC en améliorant la formule du calcul du  $R_c$ .

## 3.2 Hypothèses

Nous considérons un réseau de capteurs sans fil de superficie  $M * M$  composé de  $N$  nœuds, la solution proposée tient compte des hypothèses suivantes :

- Le déploiement des nœuds est aléatoire et tous les nœuds sont identiques en terme d'énergie initiale et de capacité de stockage, ces capacités sont limitées.
- Chaque nœud contient un identifiant et un rayon.
- Utilisation des rayons de communication différents pour avoir des clusters inégaux afin de résoudre le problème des points chauds. Le rayon de communication dépend de la distance entre le nœud et la SB, de l'énergie résiduelle et du nombre des nœuds voisins.
- Les CHs envoient les données à la SB.
- La station de base possède une réserve d'énergie illimitée, donc tous les nœuds sont à portée de la SB.

### 3.3 Description de la proposition

#### 3.3.1 Proposition

Notre proposition est une hybridation de 3 protocoles : EADUC, AEADUC et HUCL. Nous avons utilisé la formule de calcul du temps d'attente de protocole HUCL, la formule de calcul de  $E_{relay}$  du protocole EADUC et la formule de calcul de rayon de communication de protocole AEADUC en améliorant cette dernière.

• **Amélioration de la formule de EADUC Amélioré :**

La formule du rayon de compétition  $R_c$  dans ce protocole est la suivante :

$$R_c = \left[ 1 - \alpha \frac{d_{max} - d(s_j, SB)}{d_{max} - d_{min}} - \beta \left( 1 - \frac{E_r}{E_{max}} \right) + \gamma \left( 1 - \frac{S_j(nb)}{nb_{max}} \right) \right] R_{max} \quad (3.1)$$

$d(s_j, SB)$  est la distance entre le nœud  $j$  et  $SB$ .

$S_j(nb)$  est le nombre des nœuds voisins par rapport au nœud  $j$ .

$nb_{max}$  est la valeur max des nœud voisins.

D'après les informations données dans [26].  $\alpha$   $\beta$  et  $\gamma$  sont égales à 1, et le rayon de communication doit être inférieur à  $R_{max}$  et supérieur à 0. Nous avons considéré que :  $d_{max}=289$  m,  $d_{min}=50$  m,  $E_{max}=2j$ ,  $R_{max}=100m$ ,  $nb_{max}=99$ ,  $E_{min}=0.1j$ ,  $nb_{min}=1$ .

En prenant en compte les valeurs d'extrémité(max et min), on aura 8 cas à étudier :

1. le premier cas : on prend les valeurs maximales de  $d(s_i, SB)$  ,  $E_r$  et  $N = S_j(nb)$  ; le  $R_c = 100$  m
2. le deuxième cas : on prend les valeurs minimales de  $d(s_i, SB)$  ,  $E_r$  et  $N = S_j(nb)$  ; le  $R_c = 3.98$  m
3. le troisième cas : on prend la valeur maximale de  $d(s_i, SB)$ , et les valeurs minimales de  $E_r$  et  $N = S_j(nb)$  ; le  $R_c = 103m > R_{max}$ .
4. le quatrième cas : on prend la valeur minimale de  $d(s_i, SB)$  , la valeur maximale de  $E_r$  et la valeur minimale de  $N = S_j(nb)$  ; le  $R_c = 99m$

5. le cinquième : on prend les valeurs minimales de  $d(s_i, SB)$  et  $E_r$  et la valeur maximale de  $N = S_j(nb)$  ; le  $R_c = -95\text{m} < 0$
6. le sixième cas : on prend les valeurs maximale de  $d(s_i, SB)$  et  $E_r$  et la valeur minimale de  $N = S_j(nb)$  ; le  $R_c = 198 \text{ m} > R_{max}$ .
7. le septième cas : on prend la valeur minimale de  $d(s_i, SB)$ , et les valeurs maximales de  $E_r$  et  $N = S_j(nb)$  sont grands ;le  $R_c = 0\text{m}$
8. le huitième cas : on prend la valeur maximale de  $d(s_i, SB)$  , la valeur minimale de  $E_r$  et la valeur maximale de  $N = S_j(nb)$  est grand ;le  $R_c = 5\text{m}$

### **Résolution au problème**

Pour trouver dans tous les cas possibles :  $0 \leq R_c \leq R_{max}$ .

nous avons résolu ce problème mathématiquement :

nous avons mis une condition que  $d(s_i, SB) \geq d_{min}$  .

on remplace :

$$\frac{d_{max}-d(s_j,SB)}{d_{max}-d_{min}}, \left(1 - \frac{E_r}{E_{max}}\right) \text{ et } \left(1 - \frac{S_j(nb)}{nb_{max}}\right) \text{ respectivement par } x, y \text{ et } z.$$

$$1 - \alpha x - \beta y + \gamma z > 0$$

$$x = y = z = 1 \implies 1 - \alpha - \beta + \gamma > 0 \implies 1 + \gamma > \alpha + \beta \quad (3.2)$$

donc l'intervalle est :  $[0, \alpha + \beta \leq 1 + \gamma]$

en remplaçant  $\alpha = \beta = \gamma = 0.5$  ,on aura  $1 < 1.5$  c'est à dire la condition de l'intervalle est vérifiée, donc on fait nos calcule et on aura les résultats suivants :

$d_{max}=289 \text{ m}$ ,  $d_{min}=50 \text{ m}$ ,  $E_{max}=2\text{j}$ ,  $R_{max}=100\text{m}$ ,  $nb_{max}=99$ ,  $E_{min}=0.1\text{j}$ ,  $nb_{min}=1$ .

En prenant en compte les valeurs d'extrémité(max et min), on aura 8 cas à étudier :

1. le premier cas : on prend les valeurs maximales de  $d(s_i, SB)$  ,  $E_r$  et  $N = S_j(nb)$  ; le  $R_c = 100 \text{ m}$
2. le deuxième cas : on prend les valeurs minimales de  $d(s_i, SB)$  ,  $E_r$  et  $N = S_j(nb)$  ; le  $R_c = 51.99 \text{ m}$

3. le troisième cas : on prend la valeur maximale de  $d(s_i, SB)$ , et les valeurs minimales de  $E_r$  et  $N = S_j(nb)$ ; le  $R_c = 101\text{m} > R_{max}$ .
4. le quatrième cas : on prend la valeur minimale de  $d(s_i, SB)$ , la valeur maximale de  $E_r$  et la valeur minimale de  $N = S_j(nb)$ ; le  $R_c = 50\text{m}$
5. le cinquième : on prend les valeurs minimales de  $d(s_i, SB)$  et  $E_r$  et la valeur maximale de  $N = S_j(nb)$ ; le  $R_c = 2.5\text{m}$
6. le sixième cas : on prend les valeurs maximale de  $d(s_i, SB)$  et  $E_r$  et la valeur minimale de  $N = S_j(nb)$ ; le  $R_c = 149.49\text{ m} > R_{max}$ .
7. le septième cas : on prend la valeur minimale de  $d(s_i, SB)$ , et les valeurs maximales de  $E_r$  et  $N = S_j(nb)$  sont grands; le  $R_c = 50\text{m}$
8. le huitième cas : on prend la valeur maximale de  $d(s_i, SB)$ , la valeur minimale de  $E_r$  et la valeur maximale de  $N = S_j(nb)$  est grand; le  $R_c = 52.5\text{m}$

### **Remarque**

Dans le 6<sup>e</sup> et le 3<sup>e</sup> cas, nous avons trouvé que le  $R_c = 149$  et  $101$  respectivement qui est supérieur au  $R_{max}$ , dans ce cas la formule du  $R_c$  dans EADUC Amélioré est incorrecte.

### **Preuve**

$$1 - \alpha x - \beta y + \gamma z > 0$$

on prend  $x = y = 0$  on aura  $1 + \gamma z > 1$  (3.3)

donc forcément le  $R_c$  grandit

la formule correcte est :

$$R_c = \left[ 1 - \alpha \frac{d_{max} - d(s_j, SB)}{d_{max} - d_{min}} - \beta \left( 1 - \frac{E_r}{E_{max}} \right) - \gamma \left( 1 - \frac{S_j(nb)}{nb_{max}} \right) \right] R_{max} \quad (3.4)$$

dans l'intervalle  $[0,1]$  avec  $0 < \alpha + \beta + \gamma \leq 1$ .

## **3.3.2 Fonctionnement du protocole**

Le protocole se déroule en tours, chaque tour est composé de deux phases : la phase d'installation du réseaux et la phase de transmission de données

## 1. Phase d'installation du réseau :

cette phase est subdivisée en trois sous phases, sont :

- (a) **Sous-phase de collecte d'information des nœuds voisins** : Chaque nœud envoie un message *Neighbor\_Msg* qui contient l'Id de nœud et son  $E_r$  dans un rayon  $r$ , au même temps, il reçoit des messages *Neighbor\_Msg* de ses nœuds voisins, à la fin de cette phase, chaque nœud connaît le nombre de ces nœuds voisins. Chaque nœud calcule son temps d'attente  $T_i$  en appliquant la formule suivante :

$$T_i = \frac{E_{init}}{E_r} \times \frac{1}{p * |NB_i|} \times T2 \times V_r \quad (3.5)$$

$NB_i$  : c'est le nombre de nœuds voisins.

$E_r$  : c'est l'énergie résiduelle de nœud

$E_{init}$  : c'est l'énergie initiale .

$V_r$  : est une valeur réelle distribuée aléatoirement dans l'intervalle  $[0.9, 1]$  qui est introduite pour réduire la probabilité que deux nœuds envoient *Head\_Msg* en même temps.

$p$  : le nombre de fois que le nœud  $i$  élus comme CH.

$T2$  : le temps nécessaire pour la deuxième sous phase.

---

**Algorithm 11** Collecte des informations des nœuds voisins (pour chaque nœud su réseau)

---

**Entrées** :  $NB_i, E_r, E_{init}, V_r$

**Sorties** :  $T_i$

**Début**

Envoyer *Neighbor\_Msg*

Compter le nombre de ses nœuds voisins  $NB_i$

Générer le  $V_r$

Calculer le temps d'attente  $T_i$

**Fin**

---

- (b) **Sous-phase de compétition des CHs** : Dans cette phase, si le nœud ' $S_i$ ' ne reçoit pas de *Head\_Msg* lorsque le temps  $T_i$  se termine, il envoie

le message *Head\_Msg* pour avvertir qu'il sera un CH pour le tour courant dans le rayon de compétition  $R_{c_i}$ , sinon il quitte la compétition.

Chaque nœud calcule son propre rayon de compétition  $R_c$ , afin de générer des clusters inégales. La formule du  $R_c$  du protocole proposé est la suivante :

$$R_c = [1 - \alpha \frac{d_{max} - d(s_j, SB)}{d_{max} - d_{min}} - \beta(1 - \frac{E_r}{E_{max}}) - \gamma(1 - \frac{S_j(nb)}{nb_{max}})]R_{max} \quad (3.6)$$

$d(s_j, SB)$  est la distance entre le nœud  $j$  et SB.

$S_j(nb)$  est le nombre des nœuds voisins par rapport au nœud  $j$ .

$nb_{max}$  est la valeur max des nœud voisins.

$\alpha$ ,  $\beta$  et  $\gamma$  : sont les facteurs pondérés dans  $[0,1]$  tel que :  $\alpha + \beta + \gamma \leq 1$ .

$R_{max}$  : c'est la valeur maximale du rayon de compétition.

---

**Algorithm 12** Compétition des CHs(pour chaque nœud du réseau)

---

**Entrées :**  $T_i$ ,  $d(s_j, SB)$ ,  $nb_{max}$ ,  $R_{max}$ ,  $\alpha$ ,  $\beta$  et  $\gamma$

**Sorties :**  $R_c$ ,  $Role_i$

**Début**

$Role_i=0$  \*/le nœud est en état normal(non CH)

**if** ( $(t_i = 0)$  et (non réception de *Head\_Msg*)) **then**

    Calculer  $R_c$

    envoyer *Head\_Msg* dans le rayon  $R_c$

$Role_i=1$  \*/le nœud devient en état CH

**else**

    Quitter la compétition

$Role_i=0$

**end if**

**Fin**

---

- (c) **Sous-phase de formation de cluster** : Chaque nœud non CH choisit le CH le plus proche et lui envoie le message *Join\_Msg* ( contient l'id et l'énergie résiduelle de ce nœud).

Chaque CH crée une liste de programmation de nœud selon la réception du message *Join\_Msg*, et envoie la liste de programme aux membres du

cluster par diffusion du message *SCHedule\_Msg*.

## 2. Phase de transmission de données

les membres du clusters collectent les données locales de l'environnement et envoient les données collectées aux CHs, Ce processus est appelé communication intra-cluster. Les membres du cluster communiquent directement avec les CHs, tout comme le protocole LEACH.

Communication inter-cluster :les CHs reçoivent et agrègent les données de leurs membres de cluster, puis envoient les données agrégées aux nœuds de saut suivants. On introduit une distance *DIST\_TH*, si  $d(S_i, SB) < DIST\_TH$  alors le nœud fait une transmission directe vers la station de base Sinon le nœud  $S_i$  construit une liste des nœuds candidats pour effectuer la transmissions à partir de ces nœuds voisins et calcule leurs paramètre  $E_{relay}$  suivant la formule suivante :

$$E_{relay} = d^2(s_i, s_j) + d^2(s_j, SB) \quad (3.7)$$

Le nœud  $S_i$  choisit le nœud ayant l'énergie résiduelle la plus élevée et une valeur de  $E_{relay}$  la plus petite.

---

### Algorithm 13 Phase de transmission (pour chaque CH)

---

**Entrées :**  $d(s_i, s_j)$ ,  $d(s_j, SB)$ ,  $DIST - TH$

**Sorties :**  $E_{relay}$

**Début**

**if** ( **then**  $d(s_i, SB) < DIST - TH$ )

transmettre les données directement vers la SB

**else**

Calculer  $E_{relay}$

$$E_{relay} = d^2(s_i, s_j) + d^2(s_j, SB) \quad (3.8)$$

Chaque nœud  $i$  choisit le nœud  $j$  ayant le  $E_{relay}$  le plus petit et l' $E_r$  la plus grande

Transmettre les données par le nœud  $j$

**end if**

**Fin**

---

## 3.4 Simulation et évaluation des performances

### 3.4.1 Le choix de MATLAB

MATLAB (abréviation de MATrix LABoratory) c'est un système interactif dont les variables et les éléments de la base de données ne nécessitent pas de dimensionnement.

MATLAB utilise plusieurs Tool-boxes (Boîtes à outils), qui sont des collections complètes des fonctions spécifiques à un domaine d'application donné : Traitement du signal, traitement d'images, télécommunications et utilise aussi des fonctions prédéfinies.

MATLAB est aussi un langage de programmation de haut niveau dans lequel nous trouvons les concepts des langages de programmation modernes tel que : le pascal, C++.

Tous ces avantages ont rendu MATLAB un langage de programmation très utilisé par les chercheurs.

Pour l'implémentation de notre proposition on a utilisé MATLAB 18.0[1].

### 3.4.2 Simulation

#### 3.4.2.1 Le modèle énergétique

Pour cette simulation, nous allons utiliser le modèle énergétique de LEACH.

$E_{Tx}$  c'est l'énergie utilisée pour la transmission

$E_{Rx}$  c'est l'énergie utilisée pour la réception

$$E_{Tx}(l, d) = \begin{cases} l * E_{elec} + l * E_{fs} * d^2 & \text{si } d < d_0, \\ l * E_{elec} + l * E_{mp} * d^4 & \text{si } d \geq d_0, \end{cases} \quad (3.9)$$

$$E_{Rx}(l) = l * E_{elec}, \quad (3.10)$$

$E_{elec}$  est énergie consommée par bit pour la transmission et la réception. Les CHs doivent aussi faire l'agrégation de données puis les transmettre vers le nœud relais, l'énergie consommée pour l'agrégation de données est  $E_{DA}$ .

### 3.4.2.2 Paramètres de simulation

nous avons considéré que les nœuds utilisés dans la simulation sont homogènes, ont la même quantité d'énergie initiale et que le réseau est déployé sur une zone carrée de dimension 200 x 200(m<sup>2</sup>). Le tableau ci-dessous montre les paramètres utilisés :

Description	Symbole	Valeur
Nombre de nœuds	N	100, 200, 300
Energie initiale des nœuds	$E_r$	2 J
Energie consommée par l'amplificateur pour une petite distance	$E_{fs}$	$10 * 10^{-12}$
Energie consommée par l'amplificateur pour une grande distance	$E_{mp}$	$13 * 10^{-14}$
Energie de transmission et de réception par bit	$E_{elec}$	$50 * 10^{-9}$
Taille des paquets de données	l	50000 bits
Energie d'agrégation	$E_{DA}$	$5 * 10^{-9}$
La valeur maximale de rayon de communication	$R_{max}$	50 m

Tableau 3.1 – Paramètres de simulation

### 3.4.2.3 Résultats de simulation

#### Déploiement du réseau du protocole UCP :

La figure ci-dessous montre le déploiement de 100 nœuds dans une zone de 200 x 200 d'une manière aléatoire.

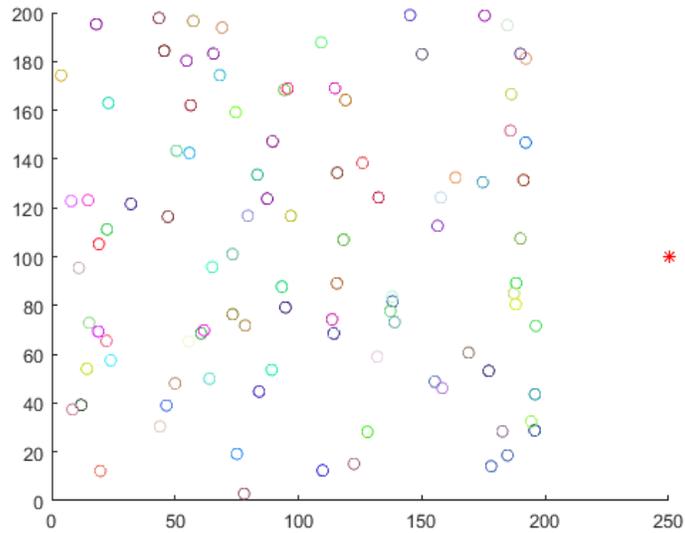


FIGURE 3.1 – Déploiement du réseau

Nous avons fait une comparaison entre trois protocoles EADUC, EADUC Amélioré et UCP.

Pour obtenir de bon résultats de la comparaison, on va utiliser les mêmes dimensions du réseau(200\*200) et le même emplacement de la station de base. Les performances des trois protocoles sont évaluées en se basant sur les mesures suivantes :

- **Quantité d'énergie** : dans les RCSFs, la plupart des approches visent à minimiser la consommation d'énergie et prolonger la durée de vie du réseau. La contrainte d'énergie est le paramètre le plus étudié pour la validation.
- **Durée de vie** : La durée de vie du réseau est devenue la caractéristique clé pour évaluer les réseaux de capteurs d'une manière spécifique à l'application.
- **Scalabilité** : représente la continuité du fonctionnement du protocole avec les mêmes performances même en augmentant le nombre de nœuds du réseau (passage à l'échelle).

Le meilleur protocoles est celui qui minimise la consommation d'énergie et prolonge la durée de vie du réseau et assure la scalabilité .

### 1. Quantité d'énergie résiduelle

La figure ci-dessus représente l'énergie résiduelle des nœuds par rounds, la différence de dissipation d'énergie des quatre protocoles EADUC, AEADUC et HUCL avec le protocole proposé UCP.

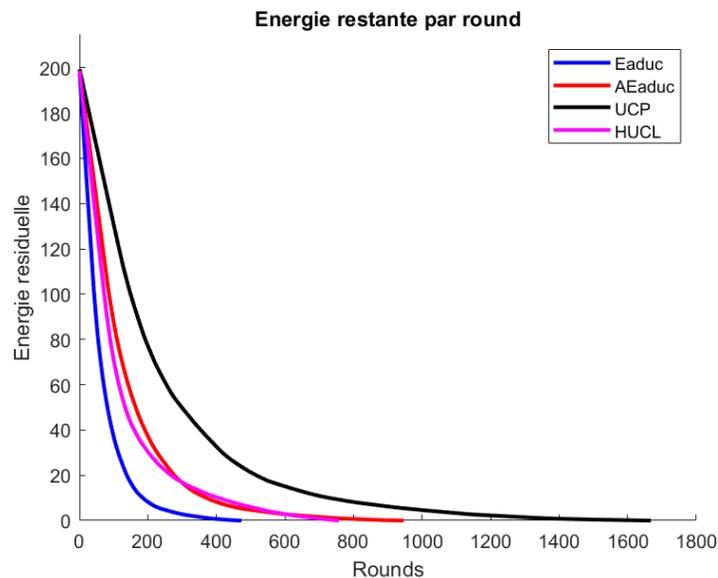


FIGURE 3.2 – Energie résiduelle par rounds

D'après les résultats de la simulation, on remarque que dans le protocole EADUC, l'énergie de tous les nœuds de réseau s'épuise au round 500.

Dans HUCL l'énergie de tous les nœuds de réseau s'épuise au round 900.

Dans AEADUC l'énergie de tous les nœuds de réseau s'épuise au round 1110

par contre dans le protocole UCP l'énergie de tous les nœuds de réseau s'épuise au round 1879.

Donc, on peut dire que le protocole proposé UCP surpasse les trois autres protocoles EADUC, AEADUC, et HUCL et cela est dû à l'hybridation de ces derniers. Le protocole UCP économise l'énergie par :

L'utilisation de la formule de calcul de rayon de communication de AEADUC

qui prend en considération la distance entre le nœud et la SB, l'énergie résiduelle de nœud et le nombre de nœuds voisins ce qui permet de créer des clusters inégales .

Le choix de nœud relais de la même manière que le protocole EADUC, ce qui permet d'économiser l'énergie par la transmission multi-sauts.

Et enfin, l'utilisation de temps d'attente de HUCL qui ne nécessite pas beaucoup de calculs, il se contente par l'utilisation de l'énergie résiduelle de nœud, l'énergie initiale et le nombre des nœuds voisins.

## 2. Durée de vie

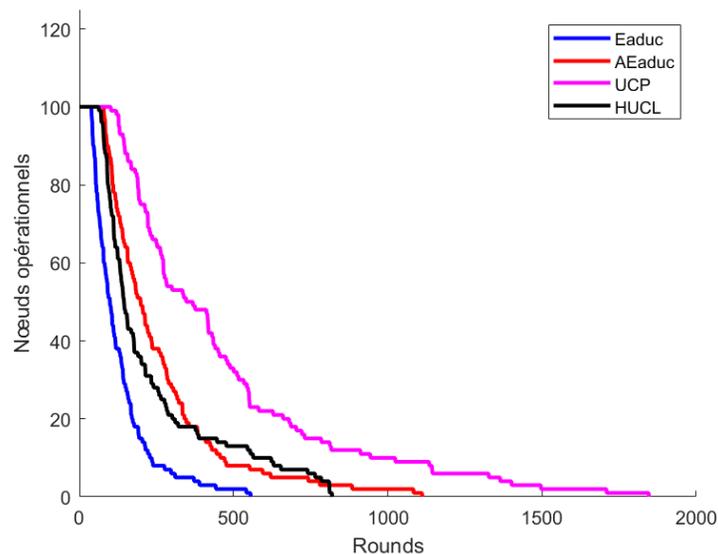


FIGURE 3.3 – Durée de vie de réseau

D'après les résultats de la simulation, on remarque que parmi les quatre protocoles comparés, le protocole proposé UCP est plus performant en terme de durée de vie.

On voit que le protocole EADUC dure jusqu'au round 500, le protocole AEADUC et HUCL durent jusqu'au round 1110 et 900 respectivement par

contre que le protocole UCP dure jusqu'au round 1879.

### 3. Scalabilité

La Figure 3.4 et le Tableau 3.2 représentent une comparaison entre les trois protocoles EADUC, AEADUC, HUCL et UCP quant à la durée de vie et ce, entre le déploiement du réseau et la mort du premier nœud.

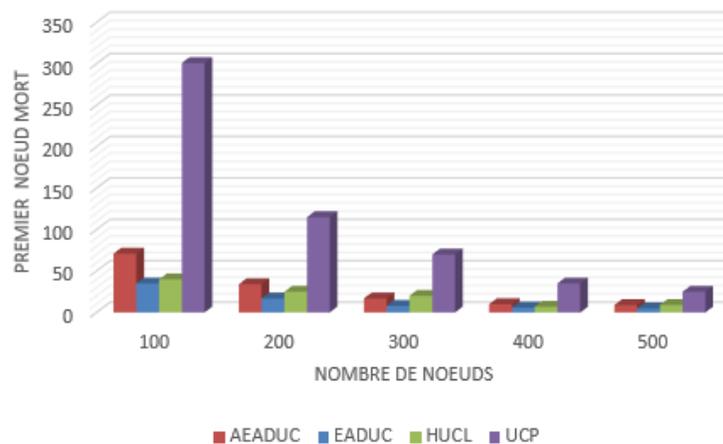


FIGURE 3.4 – Durée jusqu'à épuisement d'énergie du premier nœud en fonction du nombre de nœuds

On remarque dans la Figure 3.4 que plus le nombre de nœuds augmente, plus la durée jusqu'à la mort du premier nœud engendrée par les trois protocoles EADUC, AEADUC, HUCL et UCP diminue, jusqu'à la mort de tous les nœuds. Le protocole qu'on propose ici est plus efficace, quand le nombre de nœuds est compris entre 100 et 500, car il génère moins de messages de contrôle par rapport à EADUC, AEADUC et HUCL.

Protocole	Nombre de nœuds				
	100	200	300	400	500
EADUC	35	17	8	6	5
AEADUC	71	34	17	10	9
HUCL	40	25	20	7	9
UCP	301	115	70	35	215

Tableau 3.2 – Épuisement d'énergie du premier nœud du réseau en fonction du nombre de nœuds.

La Figure 3.5 et le Tableau 3.3 représentent une comparaison entre les trois protocoles EADUC, AEADUC, HUCL et UCP quant à la durée de vie et ce, entre le déploiement du réseau et la mort de la moitié soit 50% des nœuds en fonction de leur nombre initial.

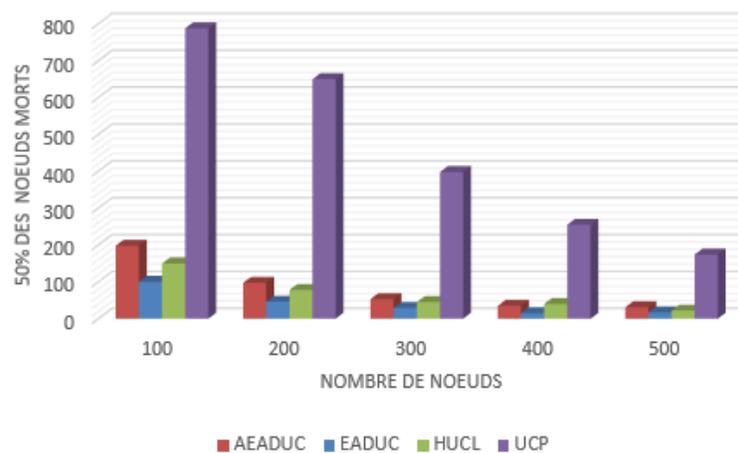


FIGURE 3.5 – Durée jusqu'à l'épuisement d'énergie de 50% des nœuds en fonction du nombre de nœuds

Protocole	Nombre de nœuds				
	100	200	300	400	500
EADUC	100	46	29	15	17
AEADUC	198	97	53	35	31
HUCL	150	78	45	40	22
UCP	788	650	398	255	174

Tableau 3.3 – Épuisement d'énergie de 50% de nœuds en fonction du nombre de nœuds déployés.

La Figure 3.6 et le Tableau 3.4 représentent une comparaison entre les trois protocoles EADUC, AEADUC, HUCL et UCP quant à la durée de vie et ce, entre le déploiement du réseau et la mort du dernier nœud en fonction de leur nombre de nœuds.

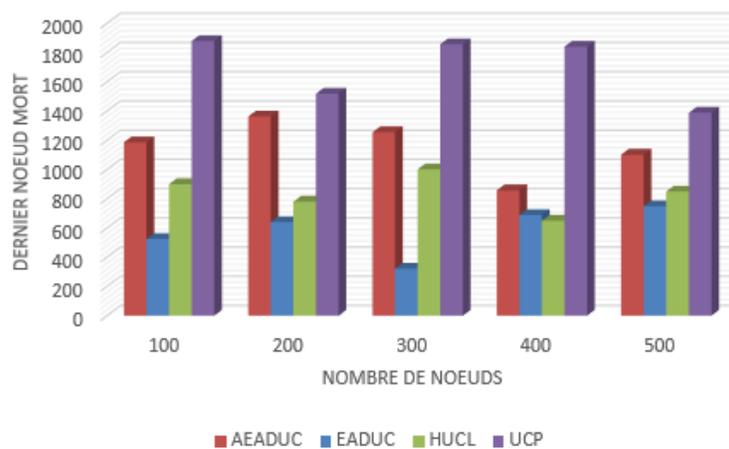


FIGURE 3.6 – Durée jusqu'à l'épuisement d'énergie du dernier nœud en fonction du nombre de nœuds

Protocole	Nombre de nœuds				
	100	200	300	400	500
EADUC	525	641	322	689	749
AEADUC	1186	1363	1256	858	1102
HUCL	900	780	1002	650	850
UCP	1879	1519	1857	1841	1389

Tableau 3.4 – Épuisement d'énergie du dernier nœud du réseau en fonction du nombre de nœuds.

### 3.5 Conclusion

Dans ce chapitre, nous avons présenté un nouveau protocole de routage hiérarchique UCP, dans les réseaux de capteurs sans fil. Après l'analyse des résultats de simulation faite, Nous pouvons conclure que notre proposition UCP est plus performante que les trois protocoles EADUC, AEADUC et HUCL en termes de l'énergie consommée par rounds ainsi que la prolongation de la durée de vie du réseau et de performance.

# Conclusion et perspectives

Ils existent plusieurs protocoles de routage dans la littérature qui ont comme objectif principale la diminution de la consommation d'énergie et l'augmentation de la durée de vie de réseau. Concevoir un protocole de routage qui est plus performant que les protocoles existants dans la littérature pose un défi pour les chercheurs.

Dans ce mémoire, nous avons commencé par quelques définitions sur les RCSFs et la notion de clustering. Ensuite nous avons fait un état de l'art qui résume tous les articles qu'on a synthétisé, en se basant sur ce dernier (état de l'art), nous avons arrivé à proposer notre protocole de clustering inégale appelé UCP et cela en se basant sur trois protocoles déjà cités dans l'état de l'art qui sont : EADUC, AEADUC, et HUCL.

Le principe de notre protocole est le suivant : nous avons amélioré la formule de calcul de rayon de compétition du protocole AEADUC pour éviter les rayons de compétition négatifs et les grands rayons qui dépassent le rayon maximal. Par la suite, nous avons pris la formule de calcul de temps d'attente de protocole HUCL, la formule améliorée de AEADUC et la formule de calcul de Erelay de protocole EADUC. Le protocole proposé offre une meilleure utilisation de la ressource énergétique et surpasse en performances les protocoles existants.

Quelques avenues de recherche se dessinent à la suite de ce présent travail : Simuler sur des simulateurs tels que OMNET, OPNET, NS3, etc. Expérimenter UCP sur une plate-forme réelle de capteurs.

# Bibliographie

- [1] Cours pdf pour débuter en matlab. <https://www.cours-gratuit.com/cours-matlab/cours-pdf-pour-debuter-en-matlab>. consulté le 05/07/2020 à 1 :36.
- [2] Mohammed Abo-Zahhad, Sabah M Ahmed, Nabil Sabor, and Shigenobu Sasaki. A new energy-efficient adaptive clustering protocol based on genetic algorithm for improving the lifetime and the stable period of wireless sensor networks. *International Journal of Energy, Information and Communications*, 5(3) :47–72, 2014.
- [3] M Mehdi Afsar and Mohamed Younis. An energy-and proximity-based unequal clustering algorithm for wireless sensor networks. In *39th Annual IEEE Conference on Local Computer Networks*, pages 262–269. IEEE, 2014.
- [4] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer networks*, 38(4) :393–422, 2002.
- [5] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer networks*, 38(4) :393–422, 2002.
- [6] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks : a survey. *IEEE wireless communications*, 11(6) :6–28, 2004.
- [7] JN Al-Karaki, AE Kamal, and Raza Ul-Mustafa. On the optimal clustering in mobile ad hoc networks. In *First IEEE Consumer Communications and Networking Conference, 2004. CCNC 2004.*, pages 71–76. IEEE, 2004.
- [8] Saad A Alharthi and Princy A Johnson. Threshold sensitive heterogeneous leach protocol for wireless sensor networks. In *2016 24th Telecommunications Forum (TELFOR)*, pages 1–4. IEEE, 2016.
- [9] Sariga Arjunan and Sujatha Pothula. A survey on unequal clustering protocols in wireless sensor networks. *Journal of King Saud University - Computer and Information Sciences*, 31, 03 2017.
- [10] Battula Babu. Energy efficient leach-c protocol for wireless sensor network. 10 2013.

- 
- [11] Kaci Bader. Détection d'intrusions dans les réseaux de capteurs sans fil. *Master recherche*, 2.
- [12] Chahrazed BAGHDADLI and Zineb FEROUANI. *DEVELOPPEMENT D'UNE APPLICATION DEPLOYEE SUR UN RESEAU DE CAPTEURS SANS FIL SUPPORTANT 6LOWPAN*. PhD thesis.
- [13] BOUCHAKEL Siham BESSAIH Aldja. Routage et simulation dans les réseaux mobiles ad hoc. mémoire master, université de Béjaia, 2017.
- [14] Florent Brissaud, Dominique Charpentier, Anne Barros, and Christophe Bérenghier. Capteurs intelligents : Nouvelles technologies et nouvelles problématiques pour la sûreté de fonctionnement. 2008.
- [15] Chiara Buratti, Andrea Conti, Davide Dardari, and Roberto Verdone. An overview on wireless sensor networks technology and evolution. *Sensors*, 9(9) :6869–6896, 2009.
- [16] Yacine Challal. Réseaux de capteurs sans fils. *Cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France*, 17, 2008.
- [17] Guihai Chen, Chengfa Li, Mao Ye, and Jie Wu. An unequal cluster-based routing protocol in wireless sensor networks. *Wireless Networks*, 15(2) :193–207, 2009.
- [18] Imrich Chlamtac, Marco Conti, and Jennifer J-N Liu. Mobile ad hoc networking : imperatives and challenges. *Ad hoc networks*, 1(1) :13–64, 2003.
- [19] Hasna Dhehibi, Ahmed Zouinkhi, Olfa Kanoun, and Nabil Derbel. An enhancement of mod-leach protocol for wireless sensor network. In *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, pages 744–749. IEEE, 2017.
- [20] Enver Ever, Ravi Luchmun, Leonardo Mostarda, Alfredo Navarra, and Purav Shah. Uheed-an unequal clustering algorithm for wireless sensor networks. 2012.
- [21] Niayesh Gharaei, Kamalrulnizam Abu Bakar, Siti Zaiton Mohd Hashim, and Ali Hosseingholi Pourasl. Energy-efficient intra-cluster routing algorithm to enhance the coverage time of wireless sensor networks. *IEEE Sensors Journal*, 19(12) :4501–4508, 2019.

- [22] J Gnanambigai, Dr N Rengarajan, and K Anbukkarasi. Leach and its descendant protocols : A survey. *International Journal of Communication and Computer Technologies*, 1(3) :15–21, 2012.
- [23] Bencan Gong, Layuan Li, Shaorong Wang, and Xuejun Zhou. Multihop routing protocol with unequal clustering for wireless sensor networks. In *2008 ISECS international colloquium on computing, communication, control, and management*, volume 2, pages 552–556. IEEE, 2008.
- [24] Xin Gu, Jiguo Yu, Dongxiao Yu, Guanghui Wang, and Yuhua Lv. Ecdc : An energy and coverage-aware distributed clustering protocol for wireless sensor networks. *Computers and Electrical Engineering*, 40(2) :384 – 398, 2014.
- [25] Awatef Ben Fradj Guiloufi, Nejah Nasri, and Abdennaceur Kachouri. An energy-efficient unequal clustering algorithm using ‘sierpinski triangle’for wsns. *Wireless Personal Communications*, 88(3) :449–465, 2016.
- [26] Vrinda Gupta and Rajoo Pandey. An improved energy aware distributed unequal clustering protocol for heterogeneous wireless sensor networks. *Engineering Science and Technology, an International Journal*, 19(2) :1050 – 1058, 2016.
- [27] Said Harchi. *Un protocole de session dans les réseaux de capteurs sans fils*. PhD thesis, 2013.
- [28] Zhengyu He and Bo Hong. Dynamically tuned push-relabel algorithm for the maximum flow problem on cpu-gpu-hybrid platforms. In *2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*, pages 1–10. IEEE, 2010.
- [29] Oliver M Heckmann. *The competitive Internet service provider : network architecture, interconnection, traffic engineering and network design*. John Wiley & Sons, 2007.
- [30] Oliver M Heckmann. *The competitive Internet service provider : network architecture, interconnection, traffic engineering and network design*. John Wiley & Sons, 2007.
- [31] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, 1(4) :660–670, 2002.

- [32] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*, pages 10–pp. IEEE, 2000.
- [33] Mohammad Ilyas. *The handbook of ad hoc wireless networks*. CRC press, 2002.
- [34] Mohammad Ilyas. *The handbook of ad hoc wireless networks*. CRC press, 2017.
- [35] Dongyao Jia, Huaihua Zhu, Shengxiong Zou, and Po Hu. Dynamic cluster head selection method for wireless sensor network. *IEEE Sensors Journal*, 16(8) :2746–2754, 2015.
- [36] Jung-Hwan Kim, Chauhdary Sajjad Hussain, Wen-Cheng Yang, Dong-Sub Kim, and Myong-Soon Park. Produce : A probability-driven unequal clustering mechanism for wireless sensor networks. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops 2008)*, pages 928–933. IEEE, 2008.
- [37] Nabila LABRAOUI. *LA SÉCURITÉ DANS LES RÉSEAUX SANS FIL AD HOC*. PhD thesis, 2012.
- [38] Ying Liao, Huan Qi, and Weiqun Li. Load-balanced clustering algorithm with distributed self-organization for wireless sensor networks. *IEEE sensors journal*, 13(5) :1498–1506, 2012.
- [39] V Loscri, G Morabito, and Salvatore Marano. A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach). In *IEEE vehicular technology conference*, volume 62, page 1809. IEEE ; 1999, 2005.
- [40] L Malathi, RK Gnanamurthy, and Krishnan Chandrasekaran. Energy efficient data collection through hybrid unequal clustering for wireless sensor networks. *Computers & Electrical Engineering*, 48 :358–370, 2015.
- [41] L. Malathi, R.K. Gnanamurthy, and Krishnan Chandrasekaran. Energy efficient data collection through hybrid unequal clustering for wireless sensor networks. *Computers and Electrical Engineering*, 48 :358 – 370, 2015.
- [42] Song Mao, Chenglin Zhao, Zheng Zhou, and Yabin Ye. An improved fuzzy unequal clustering algorithm for wireless sensor network. *Mobile Networks and Applications*, 18(2) :206–214, 2013.

- [43] Raphaël Massin, Christophe J Le Martret, and Philippe Ciblat. Un algorithme de clusterisation distribué pour les réseaux ad hoc structurés. *25th French GRETSI*, 2015.
- [44] Rohit Pachlor and Deepti Shrimankar. Lar-ch : A cluster-head rotation approach for sensor networks. *IEEE Sensors Journal*, 18(23) :9821–9828, 2018.
- [45] S. Peng and C.P. Low. Energy neutral directed diffusion for energy harvesting wireless sensor networks. *Comput. Commun.*, 63(C) :40–52, June 2015.
- [46] Peng Ren, Jiansheng Qian, Leida Li, Zhikai Zhao, and Xiaobin Li. Unequal clustering scheme based leach for wireless sensor networks. In *2010 Fourth International Conference on Genetic and Evolutionary Computing*, pages 90–93. IEEE, 2010.
- [47] Roberto Riggio, Daniele Miorandi, and Imrich Chlamtac. Airtime deficit round robin (adrr) packet scheduling algorithm. In *2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, pages 647–652. iee, 2008.
- [48] Belkheyr Sahraoui. *la Géo-localisation dans les Réseaux de Capteurs sans Fil*. PhD thesis.
- [49] Hiral Shah and SR Bhoyar. Improved v-leach protocol in wireless sensor network with data security. *OSR J. Electron. Commun. Eng*, 9 :49–54, 2014.
- [50] Georgios Smaragdakis, Ibrahim Matta, and Azer Bestavros. Sep : A stable election protocol for clustered heterogeneous wireless sensor networks. Technical report, Boston University Computer Science Department, 2004.
- [51] Stanislava Soro and Wendi B. Heinzelman. Cluster head election techniques for coverage preservation in wireless sensor networks. *Ad Hoc Networks*, 7(5) :955 – 972, 2009.
- [52] M Usha and N Sankarram. A survey on energy efficient hierarchical (leach) clustering algorithms in wireless sensor network. In *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)*, volume 2, pages 601–609, 2014.

- 
- [53] Gayathri Venkataraman, Sabu Emmanuel, and Srikanthan Thambipillai. Size-restricted cluster formation and cluster maintenance technique for mobile ad hoc networks. Master's thesis, 2007.
- [54] Hui Xia, Rui-hua Zhang, Jia Yu, and Zhen-kuan Pan. Energy-efficient routing algorithm based on unequal clustering and connected graph in wireless sensor networks. *International Journal of Wireless Information Networks*, 23(2) :141–150, 2016.
- [55] Ossama Younis and Sonia Fahmy. Heed : a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*, 3(4) :366–379, 2004.
- [56] Jiguo Yu, Yingying Qi, Gang Wang, Qiang Guo, and Xin Gu. An energy-aware distributed unequal clustering protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 7(1) :202145, 2011.
- [57] Jiguo Yu, Yingying Qi, Gang Wang, Qiang Guo, and Xin Gu. An energy-aware distributed unequal clustering protocol for wireless sensor networks. *International Journal of Distributed Sensor Networks*, 7(1) :202145, 2011.

## RÉSUMÉ

Le clustering dans les réseaux de capteurs sans fil est un moyen efficace de structurer le réseau. Son but est d'identifier un cluster et de lui attribuer un cluster-head. Celui-ci sera responsable de l'agrégation de données. L'exécution de cette agrégation entraîne une augmentation de la consommation énergétique et une diminution de la durée de vie du noeud. Dans ce mémoire, nous introduisons UCP, un protocole de clustering inégal pour le routage dans les RCSFs, le principe de UCP est l'amélioration du protocole EADUC avec l'intégration de certains propriétés des deux protocoles AEADUC et HUCL. Nous avons simulé notre proposition avec notre simulateur développé sous MATLAB, les résultats de simulation montrent qu'UCP permet d'augmenter la durée de vie du réseau par rapport à EADUC, AEADUC et HUCL et économise l'énergie.

**Mots clés :** Clustering inégal, réseau de capteurs sans fil, Routage, Optimisation d'énergie, Durée de vie.

## ABSTRACT

Clustering in wireless sensor networks is an efficient way to structure the network. Its objective is to identify a subset of nodes in the network (a cluster) and assign him a leader (a cluster-head). This one will be in charge of specific tasks such as aggregation of data. Performing these additional tasks increases the energy consumption and decrease the life of the node. In this report, we introduce UCP, an unequal clustering protocol for routing in RCSFs, the principle of UCP is the improvement of the EADUC protocol with the integration of certain properties of the two protocols AEADUC and HUCL. We simulated our proposition with a simulator program developed with MATLAB, by appearing the results obtained with protocols already existed EADUC, AEADUC and HUCL, we can see that our proposition increase the network lifetime and conserves energy.

**Key words :** Unequal clustering, wireless sensor networks, Routing, Energy optimisation, Network lifetime.