

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



جامعة بجاية  
Tasdawit n Bgayet  
Université de Béjaïa

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

## Mémoire de Fin de Cycle

En Vue de L'obtention du Diplôme Master Professionnelle  
En Informatique

Option

*Administration et Sécurité des Réseaux*

Thème

---

### Routage Hiérarchique avec Optimisation de Consommation d'Energie dans les Réseaux de Capteurs Sans-Fil.

---

*Présenté par :* M<sup>r</sup>. BELLAL Juba  
M<sup>r</sup>. MOUSSAOUI Adel

Devant le jury composé de :

Président	D <sup>r</sup> AMROUN Kamel	Maitre de conf. A	U. A/Mira Béjaïa.
Rapporteur	D <sup>r</sup> AMAD Mourad	Maitre de conf. A	U. A/Mira Béjaïa.
Examineur	M <sup>r</sup> BEDJOU Khaled	Chargé de Recherche	Cerist.
Examinatrice	M <sup>lle</sup> CHEKLAT Lamia	Doctorante LMD	U. A/Mira Bejaïa.

*Promotion 2015/2016*

# Remerciements

Ah les remerciements ! Ça permet à chaque étudiant de remémorer les moments importants qui ont ponctué ses années d'étude, ainsi que les personnes qui l'ont accompagné et encouragé. Ces années d'étude ont été l'occasion pour nous de rencontrer de nombreuses personnes qui nous ont beaucoup apporté du point de vue scientifique, mais également de partager des amitiés fortes et inoubliables.

Tout d'abord, nous remercions **Dieu** le tout puissant pour nous avoir donné la force et l'intelligence nécessaire à ce modeste travail et nous remercions aussi nos parents de nous avoir bien élevé.

Nous tenons à remercier très chaleureusement les membres du jury. Merci à **M<sup>r</sup> AMROUNE**, qui nous a fait l'honneur de présider ce jury. Nos pensées vont également à **M<sup>r</sup> BEDJOU** et **M<sup>lle</sup> CHEKLAT**, qui nous ont fait l'honneur de participer à ce jury et pour l'intérêt qu'ils ont porté à ce mémoire.

Nous tenons à dire toute nos gratitudes à notre promoteur **M<sup>r</sup> AMAD Mourad**. Merci ! Pour nous avoir encadré au sens propre du terme, puis pour nous avoir fait confiance et pour nous avoir accompagné et conseillé au cours de cette année.

Et 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.

De même, nous remercions ceux qui ont contribué à ce travail de près ou de loin, soit avec leur support, leur amitié ou leur amour.

# Dédicace

Je dédie ce modeste travail :

A mes très chers parents

A Mon cher père pour m'avoir épaulé, encouragé à reprendre les études et motivé dans les moments les plus difficiles

A ma mère pour sa patience et tous ses efforts

A mon grand-père et ma grand-mère que Dieu le Tout Puissant les gardes en bon santé

A mes frères et soeurs : pour tous leur encouragements

A mon binôme **Juba** avec qui j'ai partagé des belles années d'études et avec qui j'ai eu l'honneur de les finir.

A ma très chère amie **Djidja**.

A mes amies

Et toute personne que je connais et qui me sont chers et tous ceux qui m'aiment . . . . .

***Adel***

En mémoire de mon grand-père. . .  
Je rends grâce à Dieu de m'avoir donné le courage et la volonté.  
Ainsi que la conscience d'avoir pu terminer mes études.  
A mes très chers : A celui qui m'a toujours appris comment  
réfléchir avant d'agir, à Celui qui m'a soutenu tout au long de  
ma vie scolaire, à Celui qui n'a jamais épargner un effort.  
Pour mon bien, Mon cher père Larbi et ma chère mère Dalila. A  
mon frère et ma sœur. A toute ma famille grande et petite.  
A tous mes amis les plus sincères. A mon binôme **Adel** avec qui  
j'ai partagé des bon moments  
A ma très chère amie **Djidji**.  
Et a tous mes amis de résidence : chambres **K215** avec qui j'ai  
passé des nuits inoubliable et partager des souvenirs.

***Juba***



# Table des matières

Table des matières	v
Table des figures	viii
Liste des abréviations	ix
Liste des algorithmes	x
Liste des tableaux	xi
Introduction générale	1
<b>1 Généralités sur les réseaux de capteurs sans fil (<i>RCSF</i>)</b>	<b>3</b>
1.1 Introduction	3
1.2 Réseaux sans fil	4
1.2.1 Réseaux Ad Hoc	4
1.3 Applications des réseaux Ad hoc	4
1.4 Réseaux de capteurs sans fil ( <i>RCSFs</i> )	5
1.4.1 Capteur	5
1.4.2 Architecture de base d'un capteur	6
1.4.3 Définition des RCSFs	7
1.4.4 Architecture d'un RCSF	7
1.4.5 Caractéristique des RCSFs	9
1.5 Type de communications	10
1.6 Pile protocolaire	10
1.7 Domaines d'application	11
1.8 Besoins et facteurs de conception d'un RCSF	14
1.8.1 Tolérance aux fautes, adaptabilité et fiabilité	14
1.8.2 Gestion et consommation d'énergie	15
1.8.3 Routage Intelligent	15
1.8.4 Sécurité	15
1.8.5 Coût de fabrication	15

1.8.6	Grande échelle . . . . .	16
1.8.7	Consommation d'énergie dans les RCSFs . . . . .	16
1.9	Comparaison réseaux ad hoc et réseaux de capteurs . . . . .	17
1.10	Conclusion . . . . .	17
<b>2</b>	<b>Le routage hiérarchique dans les RCSFs</b>	<b>19</b>
2.1	Introduction . . . . .	19
2.2	Principales classes des protocoles de routage . . . . .	20
2.2.1	Routage à plat . . . . .	20
2.2.2	Routage géographique ( <i>basé sur la localisation</i> ) . . . . .	21
2.2.3	Routage hiérarchique . . . . .	22
2.3	Caractéristique d'un protocole de routage hiérarchique . . . . .	22
2.4	Protocoles de routage hiérarchique pour les ( <i>RCSFs</i> ) . . . . .	25
2.4.1	LEACH ( <i>Low Energy Adaptive Clustering Hierarchy</i> ) . . . . .	25
2.4.2	LEACH-C ( <i>Low Energy Adaptive Clustering Hierarchy Centralized</i> ) . . . . .	27
2.4.3	PEGASIS ( <i>Power-Efficient Gathering in Sensor Information Systems</i> ) . . . . .	28
2.4.4	HEED ( <i>Hybrid, Energy-Efficient, Distributed approach</i> ) . . . . .	28
2.4.5	MHEED ( <i>Multi-Hop Hybrid Energy Efficient Distributed clustering</i> ) . . . . .	30
2.4.6	<b>EEPSC (Energy Efficient Protocol With Static Clustering)</b> . . . . .	<b>31</b>
2.4.7	SOP ( <i>Self-Organizing Protocole</i> ) . . . . .	32
2.5	Tableau comparatif pour les protocoles de routage hiérarchique étudiés . . . . .	33
2.6	Conclusion . . . . .	34
<b>3</b>	<b>Protocole de routage hiérarchique EEPSCZ (<i>Energy Efficient Protocol With Static Clustering Zones</i>)</b>	<b>35</b>
3.1	Introduction . . . . .	35
3.2	Proposition . . . . .	35
3.2.1	Motivation . . . . .	35
3.2.2	Hypothèses . . . . .	36
3.2.3	Principe générale de fonctionnement . . . . .	37
3.3	Conclusion . . . . .	48
<b>4</b>	<b>Simulation sur MATLAB</b>	<b>49</b>
4.1	Introduction . . . . .	49
4.2	Simulation et analyse de performances . . . . .	49
4.3	Aperçu sur les simulateurs des réseaux de capteurs . . . . .	50

4.4	Choix de MATLAB . . . . .	50
4.5	Métriques de performances . . . . .	51
4.6	Etapas de réalisation . . . . .	52
4.7	Modèle de simulation . . . . .	54
4.7.1	Description du système . . . . .	54
4.7.2	Modèle d'énergie . . . . .	54
4.7.3	Variable descriptive du système . . . . .	55
4.8	Evaluation de performances . . . . .	56
4.8.1	Moyenne d'énergie consommée . . . . .	57
4.8.2	Durée de vie du réseau . . . . .	58
4.9	Conclusion . . . . .	59
	<b>Conclustion Générale</b>	<b>60</b>
	<b>Bibliographie</b>	<b>62</b>

# Table des figures

1.1	<i>Exemple de capteur sans fil [5]. . . . .</i>	5
1.2	<i>Architecture de base d'un capteur [26]. . . . .</i>	6
1.3	<i>Architecture générale d'un réseau de capteurs sans fil [8]. . . . .</i>	8
1.4	<i>Architecture générale d'un réseau de capteurs sans fil. . . . .</i>	10
1.5	<i>Réseaux de capteurs militaire [26]. . . . .</i>	12
1.6	<i>Réseau de capteur pour l'agriculture [26]. . . . .</i>	13
1.7	<i>Réseau de capteur pour la surveillance médicale [5]. . . . .</i>	13
2.1	<i>Classes de protocoles de routage. . . . .</i>	19
2.2	<i>Algorithme de routage LEACH [8]. . . . .</i>	25
3.1	<i>Formation des niveaux. . . . .</i>	38
3.2	<i>Désignation des CH_T. . . . .</i>	39
3.3	<i>Election des CHs. . . . .</i>	43
3.4	<i>Ordonnancement TDMA [13]. . . . .</i>	45
4.1	<i>Étapes de simulation. . . . .</i>	51
4.2	<i>Déploiement des nœuds. . . . .</i>	52
4.3	<i>Modèle d'énergie [11]. . . . .</i>	54
4.4	<i>Moyenne d'énergie consommée. . . . .</i>	56
4.5	<i>Durée de vie moyenne du réseau. . . . .</i>	57

# Liste des abréviations

<b>CDMA</b>	<b>C</b> ode <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess.
<b>CH</b>	<b>C</b> luster <b>H</b> ead.
<b>CH_T</b>	<b>C</b> luster <b>H</b> ead <b>T</b> emporaire.
<b>CSMA</b>	<b>C</b> arrier <b>S</b> ense <b>M</b> ultiple <b>A</b> ccess.
<b>EEPSC</b>	<b>E</b> nergy <b>E</b> fficient <b>P</b> rotocol <b>S</b> tatic <b>C</b> lustering.
<b>EEPSCZ</b>	<b>E</b> nergy <b>E</b> fficient <b>P</b> rotocol with <b>S</b> tatic <b>C</b> lustering <b>Z</b> ones.
<b>FDMA</b>	<b>F</b> requency <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess.
<b>GAF</b>	<b>G</b> eographic <b>A</b> daptive <b>F</b> idelity.
<b>GPS</b>	<b>G</b> lobal <b>P</b> osition <b>S</b> ystem.
<b>HEED</b>	<b>H</b> ybrid <b>E</b> nergy- <b>E</b> fficient <b>D</b> istributed approach.
<b>LBNL</b>	<b>L</b> awrence <b>B</b> erkeley <b>N</b> ational <b>L</b> aboratory.
<b>LEACH</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchy.
<b>LEACH-C</b>	<b>L</b> ow <b>E</b> nergy <b>A</b> daptive <b>C</b> lustering <b>H</b> ierarchy <b>C</b> entralized.
<b>MAC</b>	<b>M</b> edium <b>A</b> ccess <b>C</b> ontrol.
<b>MHEED</b>	<b>M</b> ulti- <b>H</b> op <b>H</b> ybrid <b>E</b> nergy <b>E</b> fficient <b>D</b> istributed clustering.
<b>NAM</b>	<b>N</b> etwork <b>A</b> nimator.
<b>NS-2</b>	<b>N</b> etwork <b>S</b> imulator <b>2</b> .
<b>OPNET</b>	<b>O</b> ptimum <b>N</b> etwork <b>P</b> erformance.
<b>OSI</b>	<b>O</b> pen <b>S</b> ystems <b>I</b> nterconnection.
<b>PDA</b>	<b>P</b> ersonnele <b>D</b> igitale <b>A</b> ssistant.
<b>PEGASIS</b>	<b>P</b> ower- <b>E</b> fficient <b>G</b> athering in <b>S</b> ensor <b>I</b> nformation <b>S</b> ystems.
<b>QoS</b>	<b>Q</b> uality of <b>S</b> ervice.
<b>RCSF</b>	<b>R</b> éseau de <b>C</b> apteur <b>S</b> ans <b>F</b> il.
<b>SB</b>	<b>S</b> tation de <b>B</b> ase.
<b>SOP</b>	<b>S</b> elf <b>O</b> rganizing <b>P</b> rotocole.
<b>TDMA</b>	<b>T</b> ime <b>D</b> ivision <b>M</b> ultiple <b>A</b> ccess.

# Liste des algorithmes

<i>Algorithme 1</i> . . . . .	<b>42</b>
<i>Algorithme 2</i> . . . . .	<b>45</b>

# Liste des tableaux

2.1	Classification des protocoles de routage hiérarchique. . . . .	32
4.1	Variable descriptive du système. . . . .	55

# Introduction générale

Les progrès récents dans le domaine de la micro-fabrication et des technologies de communication sans fil, ont donné naissance à une nouvelle génération de réseaux, il s'agit des réseaux de capteurs sans fil (*RCSF*).

Un réseau de capteurs sans fil est un type particulier des réseaux ad hoc, il est constitué d'un grand nombre de dispositifs physiques appelés capteurs, pouvant relever des données de l'environnement surveillé (*température, vibration, mouvement des objets, etc*) et de les transmettre vers un centre de traitement. Les applications de ce type de réseaux sont très diverses, elles incluent le domaine militaire, le contrôle du trafic routier, les espaces sanitaires, le diagnostic industriel et autres.

Les capteurs sont dotés de batteries de taille très réduite, à faible autonomie et qui ne sont pas remplaçables. De ce fait, le réseau doit accomplir sa mission tout en conservant l'énergie des nœuds. Cette dernière est consommée dans trois fonctions : la capture, le traitement et la communication. Plusieurs facteurs interviennent dans ces fonctions on cite : le type d'applications, le protocole de routage, la politique d'accès au canal de transmission, etc.

Le but d'un réseau de capteurs est défini par la capacité de faire router une grande quantité d'informations à une station de base, en assurant une longue période de vie du réseau. Le routage est l'une des problématiques majeures d'un réseau de capteurs sans fil, car la communication des données capturées est effectuée en saut, jusqu'à ce que ces données arrivent à la destination.

*L'objectif de notre travail est de traiter le problème du routage dans les réseaux de capteurs, en minimisant la consommation d'énergie et améliorer la durée de vie du réseau. Pour cela, nous avons proposé un nouveau protocole de routage hiérarchique à basse consommation d'énergie, dans le but de prolonger la durée de vie du réseau.*

Ce mémoire est composé de quatre chapitres :

Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil : leurs définitions, leurs architectures, leurs caractéristiques, communication dans les réseaux de capteur sans fil, domaines d'application, et les besoins et facteurs de conception dans un RCSF.

Le deuxième chapitre est consacré au routage hiérarchique dans les RCSFs. Les principales classes des protocoles de routage, caractéristique d'un protocole de routage hiérarchique, ainsi qu'une étude de quelques protocoles de routage hiérarchique sont présentées.

Le troisième chapitre est consacré au protocole de routage amélioré, proposé comme solution à la problématique de routage à basse consommation d'énergie dans les réseaux de capteurs sans fil tout en détaillant son principe et son fonctionnement.

Le quatrième chapitre est consacré à simuler le protocole de routage amélioré, sur MATLAB et faire une analyse des résultats obtenus.

Enfin, notre mémoire s'achève par une conclusion générale résumant les grands points qui ont été abordés ainsi que des perspectives que nous souhaitons accomplir prochainement.

# Généralités sur les réseaux de capteurs sans fil (*RCSF*)

---

## 1.1 Introduction

Les progrès dans le domaine de l'électronique miniaturisée et les communications sans fil ont donné naissance à des composants capables de prélever des grandeurs environnementales, physiologiques etc. Ces composants sont appelés des nœuds capteurs et ils ont la capacité de s'auto-organiser pour former un réseau de capteurs sans fil (*RCSF*).

Les RCSFs permettent de faciliter le suivi et le contrôle à distance de l'environnement physique avec une meilleure précision. Ils peuvent aussi être déployés pour exploiter diverses applications (*environnementales, militaires, médicales, etc.*). En outre, un réseau de capteurs est constitué généralement d'un grand nombre de nœuds capteurs car ces derniers sont sujets à pannes accidentelles ou intentionnelles. Chaque nœud est composé principalement d'un ou plusieurs capteurs, d'une unité de traitement et d'un module de communication. Ces nœuds communiquent entre eux selon une certaine topologie du réseau afin d'acheminer les informations à un centre de contrôle distant de la zone de leur déploiement. La mise en place d'un RCSF pose de nombreux problèmes parmi lesquels le routage des informations vers la station de base via les différents nœuds du réseau. Dans cette optique plusieurs contributions ont été proposées dans la littérature. Ces contributions visent à minimiser la consommation d'énergie ceci afin d'optimiser l'autonomie des nœuds qui constituent le réseau et par suite garantir une longue longévité pour le réseau entier.

Dans ce chapitre, nous présentons les réseaux de capteurs sans fil avec un plan méthodologique que nous avons adopté. Nous commençons par les réseaux sans fil plus exactement les réseaux Ad Hoc, définition d'un capteur, son architecture, et voir comment former un réseau de capteurs sans fil. Ensuite, la communication dans les réseaux de capteur, les domaines d'application des réseaux de capteurs sans fil, ainsi que les besoins et facteurs de conception dans un RCSF, et une petite comparaison entre les réseaux ad hoc et les réseaux de capteurs et on terminera le chapitre par une conclusion.

## **1.2 Réseaux sans fil**

Ils existes différents types de réseaux sans fil, parmi eux on trouve le réseaux Ad Hoc .

### **1.2.1 Réseaux Ad Hoc**

Un réseau sans-fil ad hoc est formé d'un ensemble d'hôtes qui s'organisent seuls et de manière totalement décentralisés, formant ainsi, un réseau autonome et dynamique ne reposant sur aucune infrastructure filaire. Ces hôtes peuvent être fixes ou mobiles. Selon ces hypothèses, tout ensemble d'objets, munis d'une interface de communication adéquate, est spontanément susceptible de former un tel réseau. Aucune infrastructure n'étant disponible, ces objets ont donc à découvrir dynamiquement leur environnement. Un réseau ad hoc étant avant tout un réseau sans fil, les objets se communiquent entre eux par le biais d'une interface radio. Ces communications sont donc soumises aux phénomènes physiques qui régissent les ondes radio, telle qu'une forte atténuation du signal avec la distance. Ainsi, seul les hôtes suffisamment proches les uns des autres sont capables de communiquer directement ensemble, et les communications de longue distance doivent s'effectuer par le biais d'un mécanisme nommé multi-sauts : cela signifie simplement que certains objets doivent relayer les messages de proche en proche jusqu'à ce que leur acheminement soit effectué. L'utilisation d'une antenne radio omnidirectionnelle implique également qu'un message envoyé par un émetteur quelconque est reçu par tous les récepteurs suffisamment proches de lui.

## **1.3 Applications des réseaux Ad hoc**

Les applications ayant recours aux réseaux ad hoc couvrent un très large spectre, incluant les applications militaires et de tactique, les bases de données parallèles, l'enseignement à distance, les systèmes de fichiers répartis, la

simulation distribuée interactive et plus simplement les applications de calcul distribué.

D'une façon générale, les réseaux ad hoc sont utilisés dans toute application où le déploiement d'une infrastructure réseau filaire est trop contraignant, soit parce qu'il est difficile à mettre en place, soit parce que la durée d'installation du réseau ne justifie pas de câblage à demeure [15].

## 1.4 Réseaux de capteurs sans fil (*RCSFs*)

Le réseaux de capteurs sans fil ce compose de différents élément, on trouve :

### 1.4.1 Capteur

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale et de la communiquer à un centre de contrôle via une station de base. Les capteurs sont dotés d'une batterie, capables de communiquer entre eux et de surveiller une grande variété de phénomènes ambiants, notamment : la température, l'humidité, la pression, le taux de bruits, la présence ou pas des certains types d'objets, et d'autres caractéristiques, tel que la vitesse, la direction et le volume d'un objet donné et de les transformer en données numériques : afin de les communiquer par ondes radio à travers le réseau vers la station de base (*SB*). Son intégration est une tâche difficile à réaliser en tenant compte de certaines contraintes : par exemple l'espace mémoire, la consommation énergétique, etc. [2].



FIGURE 1.1 – *Exemple de capteur sans fil [5].*

### 1.4.2 Architecture de base d'un capteur

Un capteur est composé de quatre éléments principaux :

- **Unité d'acquisition** : composée d'un capteur qui obtient des mesures sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui convertit l'information relevée et la transmet à l'unité de traitement.
- **Unité de traitement** : composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique (*TinyOS par exemple*). Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communication qui permettent de faire collaborer un capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.
- **Unité de communication** : unité responsable de toutes les émissions et réceptions de données via un support de communication radio.
- **Batterie** : un capteur est muni d'une batterie pour alimenter tous ses composants. Cependant, à cause de sa taille réduite, la batterie dont il dispose est limitée et généralement irremplaçable. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs [5].

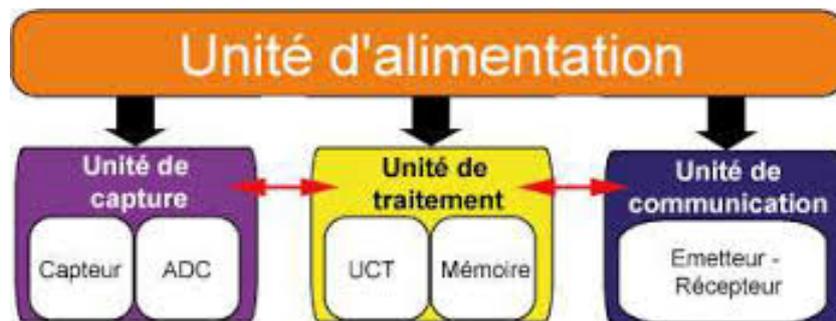


FIGURE 1.2 – Architecture de base d'un capteur [26].

### 1.4.3 Définition des RCSFs

Les réseaux de capteurs sans fil (*RCSF*) sont un type particulier de réseau Ad-hoc, dans lesquels les nœuds sont des capteurs intelligents. Ils se composent généralement d'un grand nombre de capteurs communicants entre eux via des liens radio pour le partage d'information et le traitement coopératif. Dans ce type de réseau, les capteurs échangent des informations par exemple sur l'environnement pour construire une vue globale de la région contrôlée, qui est rendue accessible à l'utilisateur externe par un ou plusieurs nœud(s). Les données collectées par ces capteurs sont acheminées directement ou via les autres capteurs de proche en proche à un « *point de collecte* », appelé station de base (*SINK*). Cette dernière peut être connectée à une machine puissante via internet ou par satellite.

Les capteurs sont déployés d'une manière aléatoire dans une zone d'intérêt, et une station de base, située à l'extrémité de cette zone, est chargée de récupérer les données collectées par les capteurs. Lorsqu'un capteur détecte un événement pertinent, un message d'alerte est envoyé à la station de base par le biais d'une communication entre les capteurs. Les données collectées sont traitées et analysées par des machines puissantes [5].

### 1.4.4 Architecture d'un RCSF

Un réseau de capteurs est constitué essentiellement de plusieurs nœuds capteurs, d'un nœud Sink et un centre de traitement des données [27].

- **Nœuds** : sont des capteurs, leur type, leur architecture et leur disposition géographique dépendent de l'exigence de l'application en question. Leur énergie est souvent limitée puisqu'ils sont alimentés par des piles.
- **Sink** : c'est un nœud particulier du réseau. Il est chargé de la collecte des données issues des différents nœuds du réseau. Il doit être toujours actif puisque l'arrivée des informations est aléatoire. C'est pourquoi son énergie doit être illimitée. Dans un réseau de capteur sans fils plus ou moins large et à charge un peu élevée, on peut trouver deux sinks ou plus pour alléger la charge.
- **Centre de traitement des données** : c'est le centre vers lequel les données collectées par le sink sont envoyées. Ce centre a le rôle de regrouper les données issues des nœuds et les traiter de façon à en extraire de l'information utile exploitable. Le centre de traitement peut être éloigné du sink, alors les données doivent être transférées à travers un autre réseau, c'est pourquoi on introduit une passerelle

entre le sink et le réseau de transfert pour adapter le type de données au type du canal (comme c'est illustré dans la figure 1.3).

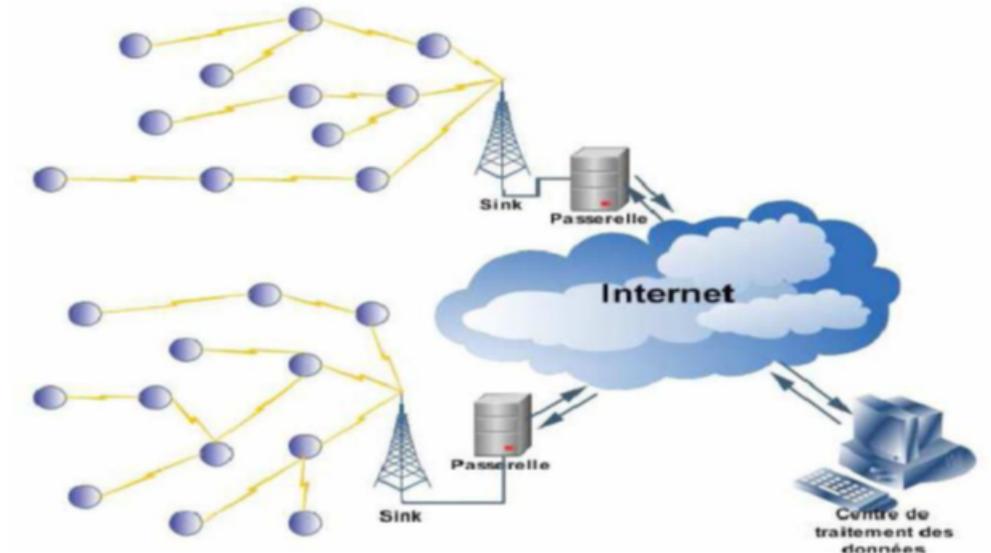


FIGURE 1.3 – Architecture générale d'un réseau de capteurs sans fil [8].

- **Type des nœuds** : dans un réseau de capteurs il existe deux types de nœuds : nœud source et nœud sink. Un nœud source est n'importe quelle entité dans le réseau qui peut fournir de l'information, c'est à dire un simple nœud capteur. Un nœud sink est l'entité où les données sont récupérées. Il y a essentiellement trois types de sink :
  - Un nœud appartenant au réseau comme n'importe quel autre nœud.
  - Une entité extérieure au réseau. Pour ce deuxième cas, le sink peut être un dispositif extérieur, par exemple, un ordinateur portatif ou un PDA interagissant avec le réseau.
  - Une passerelle vers un autre réseau tel qu'internet, où la demande de l'information vient d'un certain centre de traitement lointain.

### 1.4.5 Caractéristique des RCSFs

Parmi les caractéristiques les plus importantes d'un réseau de capteurs, nous citons [23] :

- **Durée de vie limitée** : les nœuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des régions géographiques éloignées. Par conséquent recharger ou remplacer leurs batteries devient quasiment impossible.

- **Ressources limitées** : habituellement les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces nœuds. En conséquence, la capacité de traitement et de mémoire est très limitée.

- **Topologie dynamique** : la topologie des réseaux de capteurs change d'une manière fréquente et rapide, car les nœuds capteurs peuvent être déployés dans des environnements hostiles (*par exemple un champ de bataille*), la défaillance d'un nœud capteur peut donc être très probable. De plus, les nœuds capteurs et les nœuds finaux où ils doivent envoyer l'information capturée peuvent être mobiles.

- **Agrégation des données** : dans les réseaux de capteurs, les données produites par les nœuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une approche répandue consiste à agréger les données au niveau des nœuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données.

- **Scalabilité** : les réseaux de capteurs engendrent un très grand nombre de capteurs, ils peuvent atteindre des milliers voir des millions de capteurs. Le défi à relever par les RCFS est d'être capable de maintenir leurs performances avec ce grand nombre de capteurs.

- **Bande passante limitée** : en raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits élevés.

- **Sécurité physique limitée** : cela se justifie par les contraintes et limitations physiques qui minimisent le contrôle des données transmises.

## 1.5 Type de communications

On cite deux types de communications :

- **Communication nœud capteur à un nœud capteur :**

Ce type de communication directe est employé pour des opérations locales, par exemple pendant le processus de clustering ou le processus de création de route.

- **Communication nœud capteur a un nœud intermédiaire :**

Les données capturées sont transmises d'un capteur à un nœud intermédiaire. Ce type de communication est unicast.

## 1.6 Pile protocolaire

La pile protocolaire utilisée par la station de base ainsi que par tous les capteurs du réseau comprend 5 couches (*une couche application, une couche de transport, une couche de réseau, une couche de liaison de données et une couche physique*) qui ont les mêmes tâches que celles du modèle OSI, et 3 niveaux ou plans (*un plan de gestion d'énergie, un plan de gestion de mobilité et un plan de gestion des taches*).

La figure 1.4 schématise les différentes couches de la pile protocolaire [6].

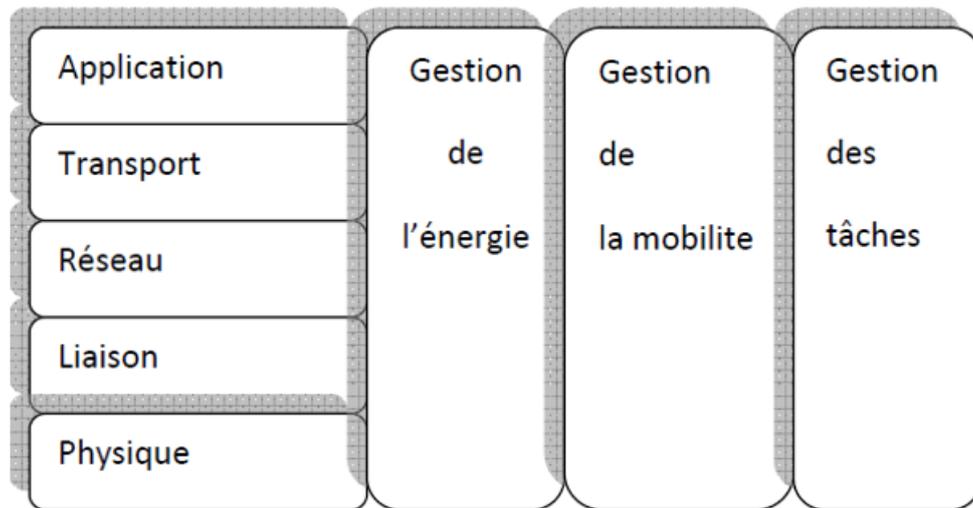


FIGURE 1.4 – Architecture générale d'un réseau de capteurs sans fil.

- **Couche application** : elle représente le niveau le plus proche de l'utilisateur, elle concerne toutes les applications implémentées dans un réseau de capteur sans fil, ces applications doivent permettre à l'utilisateur d'interagir avec les RCSFs à travers les différentes interfaces.

- **Couche transport** : elle gère le flux de données c'est-à-dire tout ce qui concerne le transport de données, leur découpage en paquets, le contrôle de flux, la conservation d'ordre des paquets et de la gestion des erreurs de transmission.

- **Couche réseau** : elle prend soin de router les données fournies par la couche transport avec optimisation de consommation énergétique.

- **Couche liaison de données** : le protocole MAC (*Media Access Control*) de la couche liaison assure la gestion de l'accès au support physique, elle est responsable du multiplexage du flux de données, elle assure une connexion fiable selon la topologie du réseau de capteur.

- **Couche physique** : elle assure la transmission et la réception des données au niveau bit, autrement dit elle est responsable de la sélection de la fréquence porteuse, de la détection du signal de la modulation.

## 1.7 Domaines d'application

Les RCSFs peuvent s'avérer utile dans de nombreuses applications, parmi lesquelles en peut citer :

- **Application militaire** : les RCSFs ont joué un rôle crucial et décisif pour l'application militaire comme par exemple lors de la gestion des forces, des équipements et des armes, il s'agit là de capteurs minuscules qui leur sont attachés pour permettre la surveillance permanente par le chef des troupes. Les RCSFs sont aussi utilisés pour la surveillance dans les champs de batailles afin de fournir des renseignements concernant leurs emplacements, leur nombre, leurs mouvements, et l'identité des soldats et des véhicules [6].



FIGURE 1.5 – Réseaux de capteurs militaire [26].

- **Gestion de stock** : on pourrait imaginer devoir stocker des denrées nécessitant un certain taux d'humidité et une certaine température. Dans ces applications, le réseau doit pouvoir collecter ces différentes informations et alerter en temps réel si les seuils critiques sont dépassés.

- **Contrôle de la pollution** : des capteurs au-dessus d'un emplacement industriel offrent la possibilité de détecter et de contrôler des fuites de gaz ou de produits chimiques. Ces applications permettent de donner l'alerte en un temps record et de pouvoir suivre l'évolution de la catastrophe [14].

- **Agriculture** : des nœuds peuvent être incorporés dans la terre et on peut interroger le réseau sur l'état du champ et déterminer par exemple les secteurs les plus secs afin de les arroser en priorité. On peut aussi imaginer équiper des troupeaux de bétail de capteurs pour connaître en tout temps, leur position ce qui éviterait aux éleveurs d'avoir recours à des chiens de berger.



FIGURE 1.6 – Réseau de capteur pour l'agriculture [26].

- **Surveillance médicale** : en implantant sous la peau de mini capteurs vidéo, on peut recevoir des images d'une partie du corps en temps réel sans aucune chirurgie. On peut ainsi surveiller la progression d'une maladie ou la reconstruction d'un muscle.

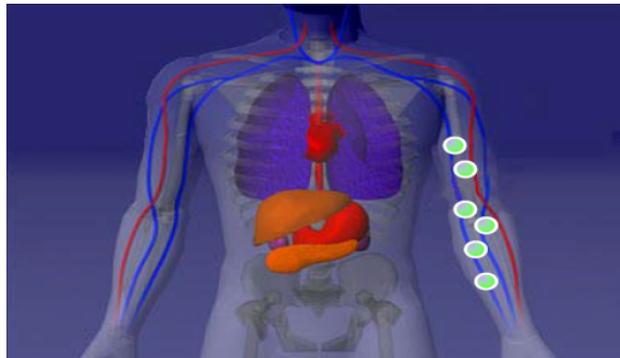


FIGURE 1.7 – Réseau de capteur pour la surveillance médicale [5].

- **Surveillance de barrages** : on peut inclure sur les parois des barrages des capteurs qui permettent de calculer en temps réel la pression exercée. Il est donc possible de réguler le niveau d'eau si les limites sont atteintes. On peut aussi imaginer inclure des capteurs entre les sacs de sables formant une digue de fortune.

La détection rapide d'infiltration d'eau peut servir à renforcer le barrage en conséquence. Cette technique peut aussi être utilisée pour d'autres constructions tels que ponts, voies de chemins de fer, routes de montagnes, bâtiments et autres ouvrages d'art.

## 1.8 Besoins et facteurs de conception d'un RCSF

Dans ce qui suit, on présentera les besoins de base et les facteurs de conception des réseaux de capteurs sans fil.

### 1.8.1 Tolérance aux fautes, adaptabilité et fiabilité

Les réseaux de capteurs sont requis pour fonctionner en s'adaptant aux changements environnementaux que les capteurs contrôlent. La fiabilité est la capacité de maintenir les fonctionnalités de réseau de capteurs sans la moindre interruption qui sera due à l'échec du nœud capteur. Ce dernier peut échouer en raison du manque d'énergie, de dommages physiques, de problèmes de communication, d'inactivité, ou d'interférence environnementale. Le réseau devrait pouvoir détecter l'échec d'un nœud et s'organiser, se reconfigurer et récupérer des échecs de nœud sans desserrer aucune information.

### 1.8.2 Gestion et consommation d'énergie

La source d'énergie est l'un des composants les plus importants d'un nœud capteur. Elle peut être généralement représentée par une simple batterie à faible autonomie d'énergie. Au-delà de l'endroit inaccessible avec moins de contrôle et d'existence humaine, les sources d'énergie jouent un rôle critique dans la survie des nœuds capteurs. Ainsi, l'énergie devrait être intelligemment divisée selon le besoin, sur les tâches de captage, de calcul, et de communication. Les capteurs peuvent être mis en veille lorsqu'ils sont inactifs. Un certain nombre de recherches courantes se concentrent sur la conception de protocoles et d'algorithmes power-aware (*consommation d'énergie minimale*) pour les réseaux de capteurs sans fil.

### 1.8.3 Routage Intelligent

Les protocoles de routage doivent être adaptatifs à la flexibilité des RCSFs (*auto-configurant*). L'information devrait être persistante malgré les change-

ments des nœuds du réseau. En outre, les algorithmes de routage devraient être intelligents pour choisir les sauts et les pas de distance miniums pour le transfert des données avec un faible coût d'énergie.

#### **1.8.4 Sécurité**

Pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein du réseau. Les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées de ces derniers. De plus, l'absence d'une protection physique des nœuds capteurs ainsi que la nature des liens sans fil, rend le réseau vulnérable aux attaques malveillantes.

#### **1.8.5 Coût de fabrication**

A cause de leur grande échelle, le coût de fabrication d'un nœud capteur doit être très réduit. Ainsi, le coût global du réseau ne doit pas être supérieur à celui d'un réseau classique afin de pouvoir justifier son intérêt.

#### **1.8.6 Grande échelle**

Les RCSFs sont généralement déployés avec un grand nombre de capteurs qui peut atteindre le million. Ceci peut engendrer des problèmes de communication et de contrôle qui nécessitent des protocoles capables de les gérer.

#### **1.8.7 Consommation d'énergie dans les RCSFs**

La première étape dans la conception de système énergétique de capteurs consiste à analyser les caractéristiques de consommation d'énergie d'un nœud de capteur sans fil. Cette analyse systématique de l'énergie d'un nœud capteur est extrêmement importante pour identifier les problèmes dans le système énergétique pour permettre une optimisation efficace. L'énergie consommée par un capteur est principalement due aux opérations suivantes : la détection, le traitement et la communication [11].

- **Energie de capture** : les sources de consommation d'énergie des nœuds pour les opérations de détection ou de capture sont : l'échantillonnage, la conversion analogique-numérique, le traitement de signal et l'activation de la sonde de capture [25].

- **Energie de traitement** : l'énergie de traitement est composée de deux sortes d'énergie : l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (*en exécutant un logiciel*). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.
- **Energie de communication** : l'énergie de communication se décline en trois parties : l'énergie de réception, l'énergie de l'émission et l'énergie en état de veille. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance ; quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur.

## 1.9 Comparaison réseaux ad hoc et réseaux de capteurs

Les réseaux de capteurs sans fil sont apparentés aux réseaux ad hoc. En effet, ces deux types de réseaux ont de nombreux points communs :

- Réseaux sans infrastructure,
- Architecture décentralisée,
- Autonomie,
- Utilisation d'ondes radio pour communiquer.

Les points de différence entre les deux réseaux sont [9] :

- La densité des nœuds déployés est beaucoup plus importante dans les réseaux de capteurs,
- Les nœuds capteurs ont des capacités limitées en énergie et en mémoire,
- La topologie des réseaux de capteurs change très fréquemment,
- Dans un réseau de capteurs, la communication entre les nœuds se fait par diffusion et non pas point à point,
- Les capteurs peuvent ne pas avoir un identifiant global à cause du grand nombre de nœuds.

## **1.10 Conclusion**

L'intérêt pour les réseaux de capteurs s'accroît considérablement, ils sont devenus actuellement des éléments indispensables dans plusieurs domaines tout en exploitant de multiples technologies de communication.

Les différentes utilisations possibles de ces réseaux démontrent la diversité et elle sont fortement dépendantes du domaine et des objectifs envisagés. A chaque application correspond une architecture du réseau et une implémentation. Ainsi, le déploiement de telle infrastructure ou topologie, à la fois dynamique et reconfigurable pose de grands défis auxquels il faut répondre.

Par exemple la durée de vie limitée de batterie du capteur, où chaque nœud est alimenté par une source d'énergie généralement irremplaçable et limitée, pose un grand problème sur la durée de vie totale du réseau en entier. Pour cela, plusieurs chercheurs essayent d'intégrer des mécanismes qui permettent de prolonger la durée de vie de réseau. Ils consacrent une grande importance à la conservation d'énergie de la communication qui est la plus gourmande dans ce type de réseau, et alors ils développent par la suite des protocoles de routage appropriés. Le chapitre suivant est consacré en détail le routage hiérarchique dans les (*RCSFs*).

# Le routage hiérarchique dans les RCSFs

---

## 2.1 Introduction

Le réseau de capteurs est composé d'un nombre important de nœuds déployés dans des vastes zones dans lesquelles tous les nœuds sont connectés entre eux. En effet, l'échange de données est pris en charge par les communications multi-sauts. Toutefois, l'opportunité d'un protocole de routage particulier dépend principalement de la capacité des nœuds et des besoins de l'application visée.

Pour assurer le passage à l'échelle (*scalabilité*) des protocoles de routage hiérarchique et garantir plus d'économie en énergie afin d'augmenter la durée de vie des réseaux de capteurs sans fils, la hiérarchisation virtuelle des réseaux peut être adoptée. Cela consiste à regrouper les nœuds en groupes dits clusters où chaque cluster est représenté par un nœud appelé cluster Head (*CH*), dont le choix se base sur plusieurs critères à savoir l'identification, la distance vers la station de base, l'énergie résiduelle, etc. Cette technique est appelée clusterisation.

Dans ce chapitre, nous abordons en premier lieu les principales classes des protocoles de routage dans les RCSFs, dans la seconde partie nous nous intéressons aux différentes approches proposées pour le routage hiérarchique de données sur les réseaux de capteurs sans fil.

## 2.2 Principales classes des protocoles de routage

La section structure du réseau présente trois classes principales de protocoles de routage dédiés aux RCSF, à savoir les protocoles utilisant le routage plat, le routage géographique ou le routage hiérarchique qui sont représentés dans la figure 2.1.

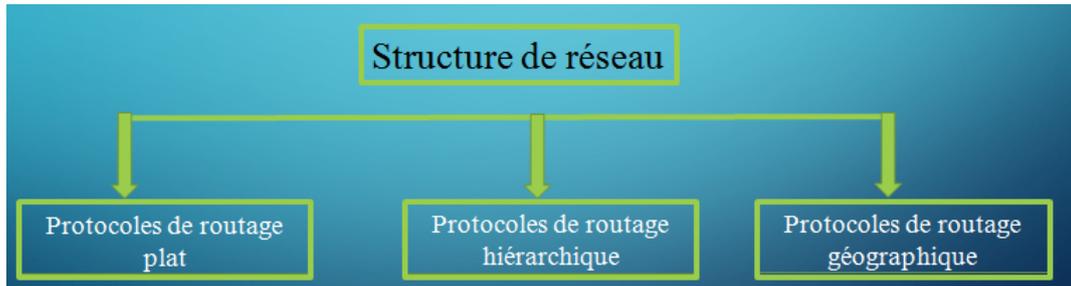


FIGURE 2.1 – Classes de protocoles de routage.

### 2.2.1 Routage à plat

Dans une structure plate du réseau, les nœuds jouent le même rôle et coopèrent ensemble pour réaliser la capture et le routage des données. En raison du grand nombre de nœuds, il n'est pas possible d'assigner un identificateur global à chaque nœud, de ce fait le routage centré donné a vu le jour. Dans ce type de routage, la station de base propage généralement ses requêtes dans tout le réseau, chaque requête possède la forme suivante [7, 17] : « *Trouver moi des données qui satisferont certaines condition* ». La station de base attend par la suite les données requises auprès des capteurs localisés dans les régions de capture sélectionnées par ces requêtes.

On peut citer comme exemple :

- FLOODING

La technique du FLOODING (*inondation*) [12] est un mécanisme classique où chaque nœud capteur transmet, par diffusion les paquets reçus à ses voisins même si ces derniers ont déjà reçu ces paquets d'une autre source ; les données seront diffusées de la même façon jusqu'à ce que le réseau soit inondé. La technique de FLOODING est une technique réactive qui ne nécessite pas une maintenance coûteuse de la topologie du réseau, elle utilise des algorithmes simples pour la découverte des routes. Cette simplicité mène à un majeur

inconvenient qui est la redondance ou la duplication des paquets. Le protocole gaspille de l'énergie, en envoyant des copies du même paquet à un même nœud, ce qui ne réduit en rien la perte de l'énergie.

La technique de FLOODING ne prend pas en compte les ressources d'énergie disponibles. Cependant, la diffusion des messages continue après la réception de ces derniers par la station de base. En effet, seul le nœud destinataire sait qu'il a reçu le message ; il y a donc un gâchis de la bande passante et des dépenses d'énergie inutiles.

### 2.2.2 Routage géographique (*basé sur la localisation*)

Un routage est dit géographique lorsque les décisions d'acheminement sont basées sur la position des nœuds. Ce type de routage utilise les informations sur l'emplacement ou la localisation des nœuds capteurs [4]. Généralement ces informations sont nécessaires pour calculer la distance entre deux nœuds donnés, et estimer des lors la puissance nécessaire pour émettre les paquets. La distance entre deux nœuds voisins peut être estimée à partir de la puissance du signal en réception. Les coordonnées des nœuds voisins sont obtenues aussi par échange d'information entre eux [4]. La position des nœuds peut être récupérée directement en utilisant des systèmes de localisation, tels que le GPS (*Global Position System*) [24]. Cette technique de routage consiste à router l'information vers le nœud dont on cherche après avoir obtenu la zone dans laquelle il se trouve. Nous présentons dans ce qui suit le protocole GAF (*Geographic adaptive fidelity*) basé sur la localisation géographique.

- GAF

Dans [31], GAF est un protocole basé sur la localisation géographique, efficace en consommation d'énergie. Il est conçu pour les réseaux ad hoc et les réseaux de capteurs. L'information de localisation utilisée dans GAF pourrait être fournie à l'aide d'un GPS. L'idée principale de GAF est de conserver l'énergie par la mise en veille des nœuds inutiles dans le réseau sans affecter le niveau de fidélité du routage. La fidélité est définie par une connectivité sans interruption entre les nœuds communicants.

GAF consiste à diviser (*partitionner*) la zone déploiement en parties zones formant des grilles virtuelles. Dans chaque zone, les nœuds collaborent entre eux afin de jouer différents rôles. Les nœuds utilisent leur GPS pour connaître leur position dans la grille virtuelle. Les nœuds associés au même point dans

la grille sont considérés équivalent en termes de cout de routage des paquets. Au niveau de chaque grille, les nœuds élisent parmi eux un seul nœud pour rester à l'état actif et les autres passent à l'état sommeil pour une certaine durée. Par ailleurs, le protocole GAF peut substantiellement augmenter la durée de vie du réseau au fur et à mesure que le nombre de nœuds augmente.

### 2.2.3 Routage hiérarchique

Le routage hiérarchique ou le routage par groupement (*cluster-based*), initialement proposé dans les réseaux câblés, est une technique bien connue avec des fonctionnalités qui résolvent les problèmes liés à la surcharge de la station de base due à la densité du réseau.

Les protocoles du routage hiérarchiques sont chargés, généralement, de confier des rôles à des nœuds du réseau, d'établir des groupes de capteurs (*clusters*), et de définir la manière dont les nœuds décident du chef de groupe (*cluster-head*) à joindre.

Les nœuds choisis comme étant des cluster-head sont des nœuds à énergie élevée. Ils peuvent être utilisés pour traiter et envoyer l'information. Les nœuds à faible énergie peuvent être employés pour exécuter la tâche de la capture à proximité de la cible [3].

## 2.3 Caractéristique d'un protocole de routage hiérarchique

Un protocole de routage hiérarchique doit spécifier plusieurs tâches mais tout d'abord quelques définitions s'imposent.

- **Clustering** : le clustering est une technique pour partitionner le réseau en groupes (*Clusters*), sachant que pour chaque groupe est désigné un leader (*Cluster Head*), ce dernier communique avec les membres de son groupe et les Cluster Heads des autres groupes. De cette manière, l'opération de clustering contribue considérablement à l'économie de l'énergie, à la réduction de la complexité des protocoles de routage, et à la résistance au facteur d'échelle, en plus de l'agrégation de données qui permet d'éliminer la redondance de données et de n'envoyer que les informations utiles [1].

- **Cluster** : le cluster est un ensemble de nœud qui forme l'unité d'organisation d'un réseau de capteur, la nature dense de ces réseaux exige la décomposition en cellules afin de simplifier les tâches de communication et reprendre aux différentes contrainte [1].

- **Cluster Head** : nécessaire pour organiser l'activité des clusters, il représente le chef du groupe surnommé aussi le leader, ses tâches sont diversifiées tel que l'organisation de la communication inter-cluster et intra-cluster, l'agrégation de données, il est élu par les autres nœuds ou bien pré-assignés par le concepteur de réseau, il peut être ordinaire comme les autres nœuds ou bien doter de plus d'énergie [1].

- **Station de base** : elle se situe au niveau supérieur de la hiérarchique d'un réseau de capteur, elle fournit une connexion entre de réseau et l'utilisateur finale [1].

- **Election des Cluster Head** : le nœud Cluster Head consomme plus d'énergie par rapport aux autres nœuds de réseau. Le Cluster Head coordonne le fonctionnement des nœuds membres de son cluster et agrège leurs données, de ce fait, il dissipe plus d'énergie créant un déséquilibre de la distribution de l'énergie sur le réseau. Pour pallier à ce problème, une rotation de ce rôle de Cluster Head est organisée au sein du cluster ou bien au sein du réseau entier. La rotation est effectuée périodiquement ou bien en fonction de la consommation de l'énergie du nœud de Cluster Head [19].

- **Communication intra-cluster** : la communication entre nœuds Cluster Head et les autres nœuds membres du cluster peut se faire, soit en un seul saut soit, en plusieurs sauts. Dans le cas d'une communication directe (*en un seul saut*), les paquets de données sont envoyés directement au Cluster Head. Cela suppose que les nœuds membres soient capables d'atteindre le CH en utilisant une transmission assez puissante pour une bonne réception de données.

Ce type de communications engendre une consommation importante d'énergie si la distance entre le CH et les autres nœuds est grande. Pour réduire la consommation de l'énergie, une communication en plusieurs sauts, de petites distances, est utilisée, dans ce cas chaque membre du cluster envoie ses données au plus

proche membre de son cluster jusqu'à l'aboutissement au CH. Ce type de communication est souvent utilisé pour réduire le nombre de collisions.

On trouve aussi des techniques employées au niveau MAC pour garantir un accès équitable et sans erreurs, à savoir, CDMA (*Code Division Multiple Access*) en utilisant un code pour chaque cluster, TDMA (*Time Division Multiple Access*) en allouant pour chaque nœud du cluster un temps spécifique (*dit Frame*) pour envoyer ses données, ou bien FDMA (*Frequency Division Multiple Access*) où chaque nœud utilise une fréquence spécifique pour l'envoi de données [30].

- **Communication inter-cluster** : les Cluster Heads communiquent avec la station de base directement, soit en deux ou plusieurs sauts via des nœuds appelés généralement des « *Nœud Gateway* ». Ces nœuds peuvent être des CHs ou bien des nœuds membres d'un cluster. L'utilisation de la consommation en multi-sauts permet de réduire la consommation d'énergie et d'augmenter la scalabilité du réseau.

- **Niveau d'énergie de données** : selon le type des capteurs utilisés, l'agrégation de données peut se faire à chaque nœud du réseau ou bien uniquement au niveau des Cluster Heads. L'agrégation de données permet de réduire la taille des données échangées entre les nœuds, et par conséquent réduire l'énergie dépensée. Plusieurs techniques d'agrégation sont utilisées à savoir : des fonctions élémentaires comme la somme, la moyenne, l'écart type, etc. ou bien des fonctions plus complexes spécifiques aux applications utilisées.

- **Qualité de service (*Qos*) et les opérations temps-réel** : dans certaines applications, la donnée doit être délivrée rapidement après sa capture, sinon elle n'est plus utile et par conséquent, la latence définie pour la livraison des données est une autre condition pour les applications qui sont soumises sous des contraintes de temps. Cependant, dans plusieurs applications la conservation d'énergie, qui est directement liée à la donnée envoyée. Pendant que l'énergie s'épuise, le réseau exige de réduire la qualité des résultats afin de réduire la diminution d'énergie dans les nœuds, et par conséquent augmenter la durée de vie du réseau [20].

## 2.4 Protocoles de routage hiérarchique pour les (*RCSFs*)

Quelques exemples sur les protocoles de routage hiérarchique :

### 2.4.1 LEACH (*Low Energy Adaptive Clustering Hierarchy*)

LEACH est un protocole de routage hiérarchique introduit par heinzelman et al. Ce protocole se base sur la clustérisations dynamique. Au départ, LEACH sélectionne au hasard quelques nœuds capteurs au tant que chefs de cluster et tourne ce rôle d'une manière uniforme pour répartir la charge entre les capteurs et prolonger la durée de vie du réseau. Dans [11], le pourcentage des nœuds capteurs qui doivent agir au tant que chefs de groupe est égale à 5%, ces clusters agrègent les données transmises par ses membres et envoient ces données à la station de base, pour cette raison, les clusters head ont besoin plus d'énergie que les autres nœuds. L'opération de Leach est divisée en tours, ou chaque tour commence avec une phase d'installation suivie d'une phase de communication :

- **Phase d'installation**

Chaque capteur choisit lui-même d'être un chef de groupe avec une probabilité  $P$  qui est choisie en fonction du nombre de clusters  $K$  et du nombre de nœud capteurs  $N$  dans le réseau.

De même pour les prochaines chois des itérations suivantes. Ce protocole vérifie si le nœud n'été pas une tête de groupe dans les plus récents tours. Puis chaque capteur choisit un nombre aléatoire,  $r$ , entre 0 et 1. Si ce nombre aléatoire est inférieur à une valeur de seuil  $T(n)$ , le nœud devient un CH pour le cycle actuel.

Le seuil est défini comme suit :

$$T(n) = \frac{P}{1 - p * (r \bmod \frac{1}{P})} \text{ (si } n \in G) \dots \dots (2.1)$$

Avec :

- P : pourcentage désiré de CHs.
- r : itération actuelle.
- G : ensemble des nœuds qui ont été sélectionnés comme CH durant les dernières  $(1/P)$  itérations.

Une fois que les CHs sont choisies, ils envoient un message d'annonce au reste des nœuds dans le réseau qu'ils sont les nouveaux chefs de cluster.

Après avoir reçu cette annonce (*ADV*) contenant l'ID du nœud et un en-tête qui distingue ce message d'annonce. Chaque nœuds non tête décident du groupe auquel ils veulent appartenir. Cette décision est basé sur l'intensité du signal d'annonce, puis il informe le CH qu'il appartient à son groupe, et cela ce fait par un message *Join-REQ* contenant l'ID du nœud et l'ID du CH. A la réception de tous les messages à partir des nœuds qui serait certainement inclus dans le cluster, le CH crée une planification TDMA afin d'attribuer pour chaque nœuds une tranche de temps ou il peut transmettre. Ce calendrier sera diffusé à tous les nœuds du cluster.

• **Phase de communication**

Selon le calendrier designer par le CH, les nœuds capteurs peuvent commencer la détection et la transmission de données à leur chef, et ce dernier agrège ces données avant de les envoyer à la station de base.

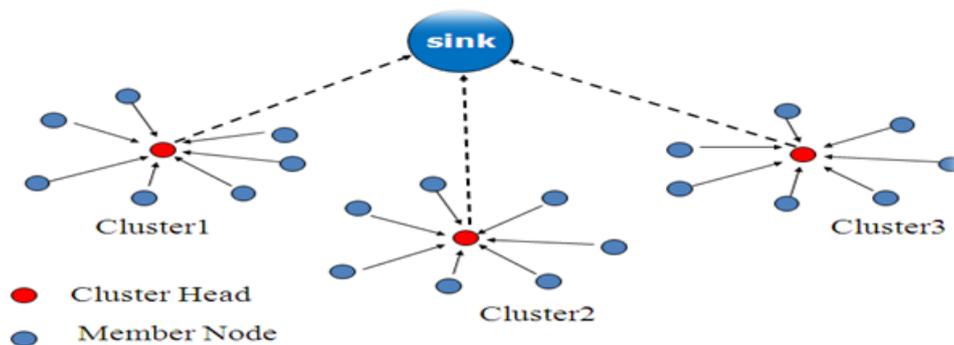


FIGURE 2.2 – Algorithme de routage LEACH [8].

**Avantage**

- La consommation d'énergie est partagée sur l'ensemble des nœuds prolongeant ainsi la durée de vie du réseau.

**Inconvénient**

- Les CHs les plus éloignés de la station de base meurent rapidement par rapport à ceux qui sont proches de la station.

### 2.4.2 LEACH-C (*Low Energy Adaptive Clustering Hierarchy Centralized*)

LEACH-C est une variante de LEACH, a été conçue pour répondre au problème de sélection aléatoire du CH dans LEACH. Elle a été proposée par les mêmes auteurs de LEACH dans [10]. Cette variante utilise une architecture centralisée pour choisir les Cluster Head, la phase d'état stable de LEACH-C est comme suite :

- Lors de la phase d'installation de LEACH-C, les nœuds informent la SB par l'envoi d'information sur leur position par GPS et leur niveau d'énergie.
- La station de base désigne les Cluster Head en se basent sur la moyenne des niveaux d'énergie, les nœuds ayant un niveau d'énergie inférieur à la moyenne ne peuvent pas être des Cluster Head dans l'itération courante.
- Enfin, la station de base diffuse un message contenant les identificateurs des Cluster Head.
- Le fonctionnement des autres opérations est similaire à celui de LEACH.

**Avantage**

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

**Inconvénient**

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

### 2.4.3 PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*)

L'idée principale de PEGASIS est de former une chaîne entre les nœuds. Dans ce protocole, un nœud peut seulement communiquer avec son voisin le plus proche, et doit attendre son tour pour pouvoir transmettre à la station de base. Le nœud capteur doit trouver son voisin le plus proche par l'envoi d'un signal et l'atténuation graduelle de ce dernier, jusqu'à ce qu'il soit reçu par un seul nœud. De ce fait la chaîne sera formée, elle doit contenir automatiquement des nœuds qui sont les plus proches les uns des autres, et qui forment un chemin vers la station de base.

Les données collectées sont transmises, et agrégées, d'un nœud à un autre jusqu'à ce qu'elles arrivent à un cluster-head. Ce dernier les transmet, à son tour, à la station de base. La sélection de ce cluster-head se base sur plusieurs paramètres prédéterminés, tel que le rapport  $(Signal)/(Bruit)$ .

Le cluster-head élu demeure seulement pour une période de temps, ensuite dans la prochaine période, un autre nœud sera choisi pour un autre round [5, 18].

#### **Avantage**

- Le changement régulier des cluster-heads par période diminue le risque de partitionnement du réseau.
- La communication entre les nœuds les plus proches réduit la bande passante.

#### **Inconvénient**

- Le choix d'un seul nœud responsable de la communication avec la station de base, à chaque période engendre, des coûts supplémentaires en énergies.

### 2.4.4 HEED (*Hybrid, Energy-Efficient, Distributed approach*)

Les auteurs de [32] ont proposé un algorithme de clustering distribué appelé HEED pour les réseaux de capteurs. Contrairement aux techniques précédentes, HEED ne fait aucune restriction sur la distribution et la densité des nœuds. Il ne dépend pas de la topologie du réseau ni de sa taille mais il suppose que les capteurs ont la possibilité de modifier leur puissance de transmission. HEED sélectionne les cluster-heads selon un critère hybride

regroupant l'énergie restante des nœuds et un second paramètre tel que le degré des nœuds. Il vise à réaliser une distribution uniforme des clusters heads dans le réseau et à générer des clusters équilibrés en taille. Un nœud  $u$  est élu comme cluster head avec une probabilité  $P_{ch}$  égale à

$$P_{ch} = \frac{C_{prob} \cdot E_n}{E_{total}} \dots \dots \dots (2.2)$$

Où :

- $E_n$  est l'énergie restante du nœud  $n$ .
- $E_{total}$  est l'énergie globale dans le réseau.
- $C_{prob}$  est le nombre optimal de clusters.

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

#### Avantages

- HEED prolonge 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.
- HEED n'indique aucune supposition sur la distribution ou la densité des nœuds, ainsi que leurs capacités.

#### Inconvénients

- Le fait, que le choix des CHs est une décision qui ne se base que sur des informations locales, des insuffisances dans la fonction du cout seront présentées telle le cas de la communication inter-clusters qui n'est pas prise en considération par cette fonction.
- Les clusters générés avec HEED ne sont pas tellement équilibrés en taille.

### 2.4.5 MHEED (*Multi-Hop Hybrid Energy Efficient Distributed clustering*)

MHEED est un protocole de routage hiérarchique, il a été proposé dans [22] pour améliorer le protocole HEED. Il se base sur la notion du multi saut pour résister à la scalabilité, et résoudre le problème de communication à un seul saut qui existe dans le protocole HEED.

Le protocole MHEED est constitué de plusieurs tours d'agglomération (*clustering*) successifs, ou chaque tour comporte deux phases :

- **Phase d'agglomération (*clustering*)** : faite selon le même principe décrit dans HEED. A la fin de cette phase, les chefs de grappes (*cluster heads*) sont élus et leurs grappes (*clusters*) sont formées.

- **Phase de communication** : la transmission de données se fait en deux étapes. Tout d'abord, les nœuds ordinaires transmettent de manière périodique leurs données à leurs chefs de grappe (*pendant les tranches de temps qui leur sont allouées*). A la réception de ces données, chaque chef va agréger les données de ces membres pour minimiser le nombre de paquets à transmettre. Il envoie le paquet agrégé à la station de base. Cette transmission est faite, de relais en relais, par le biais des chefs de grappes relais prédéterminés comme suit :

- **Dans la première variante**, appelée MHEEDB, chaque chef de grappe va garder un seul relais (*le meilleur*) parmi les chefs de grappes voisines.
- **Dans la seconde variante**, appelée MHEEDP, chaque chef de grappe va garder plusieurs relais vers la station de base parmi les chefs de grappes voisines. A chaque transmission de données, il choisit l'un d'entre eux (*de manière probabilisée*) pour faire suivre ses données ou celles reçues d'autres chefs plus éloignés l'ayant choisi pour relais. A la fin de chaque tour d'agglomération (*clustering*), le processus d'agglomération est déclenché de nouveau. De nouveaux chefs de grappes sont élus et de nouveaux relais sont recherchés parmi ces derniers pour acheminer les données.

#### Avantages

- MHEED résout le problème de scalabilité.
- MHEED permet de réaliser un équilibrage de charge entre les différentes CHs.

### Inconvénient

- Le problème des points chauds.

## 2.4.6 EEPSC (Energy Efficient Protocol With Static Clustering)

EEPSC est un protocole de routage hiérarchique basé sur le clustering statique qui a été proposé dans [33]. C'est une amélioration du protocole LEACH. Il se base sur le déploiement des nœuds, ainsi que la station de base se trouve sur l'une des extrémités du réseau. Ce protocole partitionne le réseau en clusters statiques, élimine la surcharge de clustering dynamique et utilise des clusters head temporaires pour équilibrer la charge de l'énergie entre les nœuds qui ont de grandes capacités d'énergie dans le but de prolonger la durée de vie des réseaux de capteurs sans fil.

Le principe du protocole EEPSC se divise en trois phases :

1. **Phase d'installation** : en premier lieu, la station de base diffuse  $k-1$  messages différents avec différentes puissances d'émission afin d'avoir  $k$  clusters. Par la diffusion du message  $k=1$ , les nœuds qui reçoivent ce message seront dans le même cluster, ils mettent le groupe à ID- $K$  et informent la station de base qu'ils sont les membres de cluster  $k$  par l'envoi du message Join-REQ en utilisant la méthode d'accès CSMA (*Carrier Sense Multiple Access*). De même pour la diffusion du message  $k=k-1$ . Après, les nœuds qui n'ont pas rejoint aucun clusters, collectent leur ID-cluster au message  $k$  et informent la station de base qu'ils sont les membres du dernier cluster. Par la suite, la station de base choisit aléatoirement un cluster head temporaire CH-T pour chaque cluster, et utilise le TDMA pour diviser le temps en tranches (*time slot*), chaque nœud du cluster aura une tranche de temps où il peut transmettre les données.
2. **Phase d'élection de clusters head (CHs)** : après la formation des clusters, au début de chaque tour, chaque nœud envoie son énergie résiduelle au CH-T du même cluster durant sa tranche de temps. Par la suite, le CH-T choisit le nœud qui a l'énergie résiduelle maximale comme cluster head (CH) pour le tour actuel afin de collecter et d'agrèger les données transmises à la station de base. De plus, il choisit le nœud qui a l'énergie résiduelle minimale comme cluster head temporaire pour le prochain tour en lui envoyant un message « round-start » incluant l'identifiant (ID) de cluster head du tour actuel.

3. **Phase de communication** : dans cette phase, chaque nœud du cluster envoie ses données au cluster head CH durant sa tranche du temps. Ensuite, le CH collecte et agrège les données. Après, il les envoie directement (*un saut*) à la station de base.

#### Avantages

- EEPSC utilise un clustering statique, ce qui élimine la surcharge de clustering dynamique.
- EEPSC utilise les CHs pour répartir la charge de l'énergie parmi les nœuds de capteurs.
- Prolonge la durée de vie du réseau.

#### Inconvénients

- Les CHs envoient directement leurs données à la station de base, ce qui peut épuiser un CH en une émission.
- Le critère de choix des CHs (*énergie résiduelle*), n'est pas toujours suffisant.
- EEPSC utilise la communication directe (*un seul saut*) vers la station de base, ce qui pose problème dans les grands réseaux.

### 2.4.7 SOP (*Self-Organizing Protocole*)

Ce protocole [29] basé sur une architecture composée de :

- **Station de base** : ils sont fixes et adressable.
- **Routeurs**.
- **Capteurs** : ils peuvent être hétérogènes, fixes ou mobiles, ainsi Ils surveillent l'environnement et renvoient les données vers la station de base via des routeurs, et ils sont identifiables via l'adresse du routeur auquel ils sont connectés et de basant sur sa position dans la hiérarchie.

L'auto-organisation des routeurs et la création des tables de routages s'effectué en quatre phases :

1. **Phase de découverte** : les capteurs se trouvent dans le voisinage de chaque routeur sont découverts.
2. **Phase d'organisation** : durant cette phase les groupes de capteurs sont constitués et fusionnés en formant une hiérarchie. Chaque nœud se voit assigner une adresse basée sur sa position dans la hiérarchie.
3. **Phase de maintenance** : mettre à jour les tables de routage et les niveaux d'énergie.

4. **Phase d'auto-réorganisation** : une réorganisation des groupes est effectuée dans le cas de partitionnement du réseau, ou de panne d'un nœud.

**Avantage**

- Coûts réduits pour le maintien d'une hiérarchie équilibrée et des tables de routage à jours.

**Inconvénient**

- La phase d'organisation nécessite un overhead supplémentaire.

## 2.5 Tableau comparatif pour les protocoles de routage hiérarchique étudiés

Pour comparer et analyser les différentes philosophies des protocoles de routage conçus pour les réseaux de capteurs sans fil, il est important d'utiliser des critères de classification appropriés pour pouvoir les distinguer. En effet, la classification permet aux concepteurs de mieux comprendre les caractéristiques de ces protocoles et de discerner les relations qui relient.

Protocoles / Critères	Classification	Mobilité	Agrégation de données	Qos	Scalabilité	Multi saut	Couche MAC
<b>LEACH</b>	Hiérarchique	Non	Distribué	Non	Limité	Non	TDMA-CDMA
<b>LEACH-C</b>	Hiérarchique	Non	Distribué	Non	Limité	Non	TDMA-CDMA
<b>PEGASIS</b>	Hiérarchique	Non	Non	Non	Limité	Non	TDMA
<b>HEED</b>	Hiérarchique	Non	Centralisé	Non	Bonne	Non	TDMA
<b>MHEED</b>	Hiérarchique	Oui	Non	Oui	Limité	Oui	/
<b>EEPSC</b>	Hiérarchique	Non	Centralisé	Non	Bonne	Non	TDMA
<b>SOP</b>	Hiérarchique	Possible	Oui	Non	Limité	Non	/
<b>EEPSCZ</b>	Hiérarchique	Non	Oui	Non	Bonne	Oui	TDMA

TABLE 2.1 – Classification des protocoles de routage hiérarchique.

## **2.6 Conclusion**

Tout au long de ce chapitre, nous avons effectué une étude exhaustive de quelques protocoles et approches de routage pour les RCSFs. L'analyse de ces protocoles nous a permis de mettre en relief les avantages et les inconvénients des techniques de routages adoptées par chacun d'eux, ce qui nous a inspiré pour mettre en œuvre une proposition pour le routage hiérarchique à basse consommation énergétique que nous allons présenter dans le chapitre suivant.

# Protocole de routage hiérarchique EEPSCZ (*Energy Efficient Protocol With Static Clustering Zones*)

---

## 3.1 Introduction

La majorité des travaux de recherche menés actuellement dans le domaine des réseaux de capteurs se concentrent généralement sur le problème de conservation d'énergie. Le développement d'une technique efficace permettant d'économiser la ressource énergétique qui est un objectif primordial pour les réseaux de capteurs sans fil.

Dans ce chapitre, nous proposons notre contribution baptisée EEPSCZ (*Energy Efficient Protocol With Static Clustering Zones*), qui s'appuie sur le principe des clusterheads, ainsi que le regroupement des nœuds inspiré de l'algorithme EEPSC [33], tout en l'améliorant avec un partitionnement vertical.

## 3.2 Proposition

### 3.2.1 Motivation

Dans le but d'obtenir de meilleures performances dans les réseaux de capteurs en termes d'énergie consommée, il est primordial d'utiliser d'autres critères de choix concernant les cluster heads tel que l'énergie résiduelle,

la distance par rapport à la station de base pour une maximisation de la durée de vie des clusterheads. L'objectif consiste essentiellement à optimiser la consommation de d'énergie, afin d'éviter un effondrement du réseau de capteurs.

Les algorithmes de regroupement (*clustering*) pour un réseau de capteurs permettant la constitution des grappes (*clusters*) de capteur dont chacun est dominée par un cluster-head (*CH*). Ces CHs sont souvent nécessaires pour l'organisation des activités dans les clusters, leurs tâche ne se limite pas à l'agrégation de données, elle s'étend aussi à l'organisation de la communication intra-cluster et même inter-cluster. L'énergie consommée dans les échanges intra-cluster est proportionnelle au nombre de nœuds dans le cluster. Les algorithmes de clustering proposés tentent de former généralement des clusters de même taille, donc les clusterheads ont tendance à consommer la même quantité d'énergie pendant la phase intra-cluster de transfert de données.

Par contre, durant la phase de communication inter-cluster, les nœuds proches de la station de base consomment beaucoup plus d'énergie à cause de la charge du trafic de relais. En effet, les nœuds communiquent leurs données à la station de base avec un routage multi-sauts. Le modèle de trafic résultant est non-uniforme et surcharge les nœuds proches de la station de base : la charge des nœuds près de la station de base est plus élevée par rapport aux autres nœuds.

Ainsi, les clusterheads proches de la station de base consomment plus d'énergie et meurent plus vite que les autres clusterheads : problème des points chauds. Ce qui pourrait réduire la couverture de détection et conduire à la perte du réseau.

### 3.2.2 Hypothèses

Pour appliquer les propositions fait dans notre protocole, nous allons nous basé sur le modèle du réseau suivant :

- Les nœuds capteurs sont tous identiques (*même énergie initiale, même capacité de stockage et de traitement de données*),
- Les nœuds sont distribués sur la zone de capture aléatoirement et sont fixe,
- La station de base est vu comme une ressource non limité ni épuisable,
- La station de base n'est pas au centre de la zone de capture,

- La mort de chaque capteur n'est causée que par l'épuisement de son énergie,
- Chaque nœud possède un identifiant unique,
- La station de base est capable d'envoyer des signaux unidirectionnels,
- Les nœuds peuvent gérer leur puissance d'émission.

### 3.2.3 Principe générale de fonctionnement

Lors de l'étude du problème de routage, toutes ces contraintes sont à considérer, le problème devient plus complexe lorsque le nombre de nœuds déployés augmente. Il s'agit du problème de résistance au facteur d'échelle (*scalability*).

Dans le protocole [33], les auteurs se sont penchés sur un balayage du réseau pour construire des clusters, le même principe est utilisé dans notre proposition. Par contre, nous ajoutons au balayage cité précédemment, un balayage vertical afin de créer des clusters de taille identique (*même taille*).

EEPSC utilise un routage à un saut, ce qui n'est pas optimal (*facteur de scalabilité*), pour ce faire, nous utilisons le principe du multi saut pour acheminer les données des capteurs vers la station de base inspiré du protocole MHEED [22] étudié dans le chapitre précédent.

Notre proposition se caractérise par une stratégie d'acheminement multi sauts où les nœuds envoient les paquets vers la station de base en utilisant leurs voisins pour acheminer le paquet d'un niveau à un niveau plus bas jusqu'à atteindre la station de base. Pour ce la, nous avons adopté pour le principe de la recherche de relais [22], ce dernier prend en considération l'énergie résiduelle et la distance par rapport au nœud émetteur lors du choix du nœud relais.

Le fonctionnement de notre protocole est établi en trois phases, la phase d'installation (*configuration*) où tous les clusters sont formés puis la phase de sélection des CHs, les CHs sont désignés par les métriques ; énergie résiduelle et la distance de la station de base.

Finalement la phase de routage des données vers la station de base. Les différentes phases de l'algorithme de clustering proposé sont détaillées dans ce qui suit :

#### 1. Phase d'installation

Cette phase est exécutée une seule fois durant tout le processus de routage, afin de réaliser un regroupement (*clustering*) des nœuds de

capteurs. L'installation permet de structurer le réseau en clusters statiques (*fixes*) d'où nous évitons la surconsommation (*overhead*) lors du clustering dynamique. L'installation du réseau se réalise en deux étapes, la première est caractérisée par la formation des niveaux (*ou couches*) du réseau, la seconde est réalisée par une division en zones. A la fin de cette phase, notre réseau sera divisé en plusieurs clusters statiques.

- **Formation des niveaux**

La station de base utilise un mécanisme pour partitionner le réseau en niveau, pour ce faire, la SB envoie ( $k-1$ ) messages « *Hello* » différents et avec différentes puissances de signal, où  $k$  désigne le nombre de couches dans le réseau.

La puissance du signal est variée pour avoir des distances différentes à la SB, ce qui représente les différents niveaux suivant la formule suivante :

$$D = k * Ri \dots \dots (3.1)$$

Où :

K : Nombre de niveaux .

D : Distance entre un nœud et la Station de Base .

Ri : est la portée du signal d'un nœud capteur qui est supposé pareil identique pour tous les nœuds de capteurs du réseau. A la réception du message par les nœuds à portée, ils calculent la distance de la station de base avec l'affaiblissement du signal et mettent à jour leurs ID\_niveau à  $k$ .

La figure 3.1 représente les niveaux formés dans cette phase.

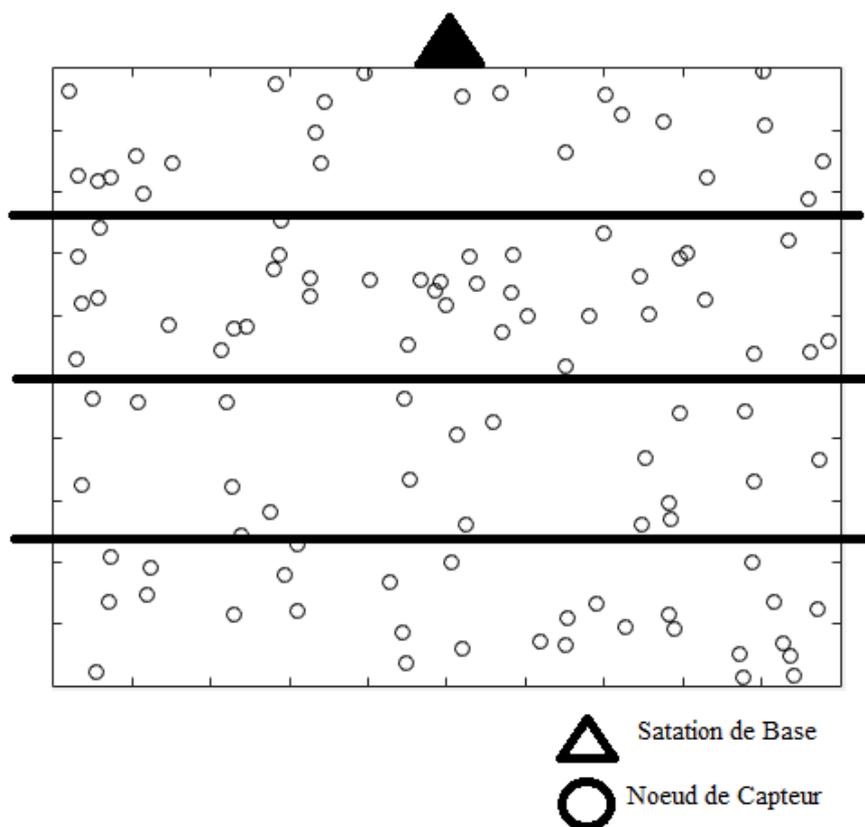


FIGURE 3.1 – Formation des niveaux.

A la fin de cette phase, le réseau est repartitionner en niveaux, où chaque nœud possède un ID\_niveau.

- **Formation des zones**

Une technique de découpage en zones est utilisée dans cette étape, une variable ID\_zone est assigné à chaque nœud capteur pour désigner sa zone ainsi les clusters du réseau sont formés.

La SB effectue un balayage vertical, pour cela la SB envoie  $(h-1)$  messages « Hello » avec différentes puissances de signal, désignant le nombre de zone dans le réseau, (la SB ne se trouve pas au milieu du réseau). La SB inclue dans ce paquet un identifiant pour chaque zone sur tous les niveaux du réseau.

A la réception du message diffusé par un nœud d'une couche il met à jour le champ ID\_zone, les clusters sont formés d'une façon à ce que chaque cluster est composé des nœuds avec un même ID\_niveau et ID\_zone. Après la détermination des clusters (*niveaux, zones*), la SB effectue une sélection aléatoire dans chaque cluster d'un CH\_T (*Cluster Head temporaire*) qui aura pour mission d'élire le CH principal durant la prochaine phase (*phase de sélection des CHs*), comme le montre la figure 3.2.

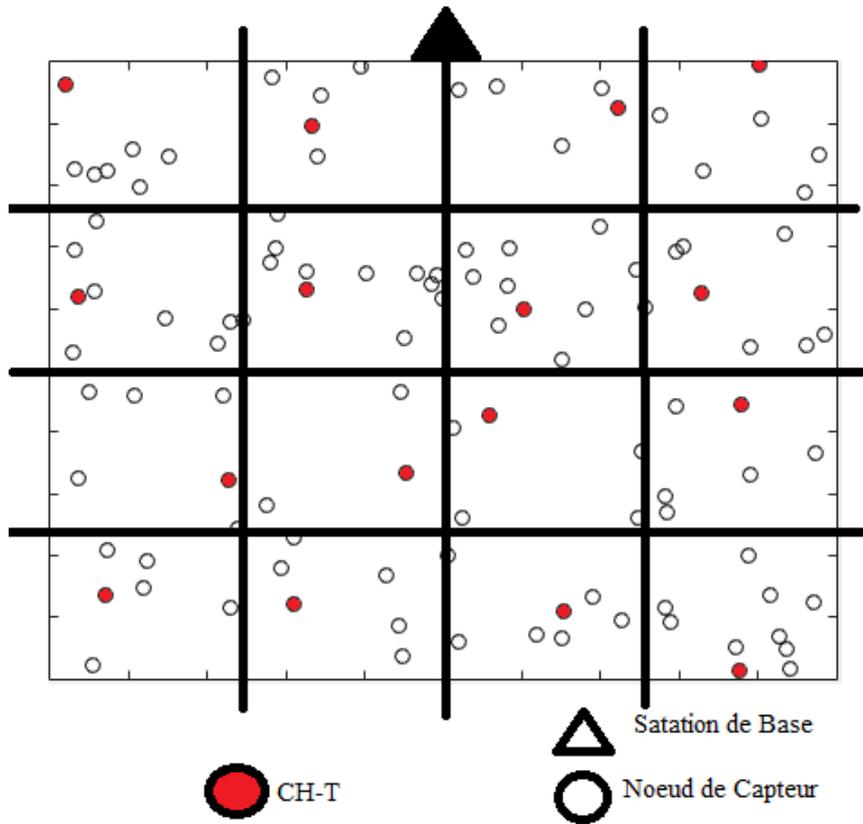


FIGURE 3.2 – Désignation des CH\_T.

Les messages utilisés dans cette phase, sont des messages de taille petite, avec un petit ID et un entête qui les distinguent comme message d'annonce. Comme le protocole LEACH, afin de minimiser la probabilité des collisions entre les messages au cours de la phase d'installation, CSMA (*Carrier Sense Multiple Access*) est utilisé comme couche MAC. La station de base envoie les règles de temps de communication, avec la technique d'ordonnement TDMA (*Time Division Multi Ac-*

*cess*) pour affecter des times slot (*Schedule*) pour chaque nœud de chaque cluster. A la fin des deux étapes, chaque cluster est fixé sur un niveau et une zone, un CH\_T est désigné aléatoirement par la station de base afin d'alléger le travail sur le CH principale qui sera élu dans la phase suivante.

L'algorithme qui résume la phase d'installation est donné comme suit :

**Algorithm 1** Algorithme de la phase d'installation.

---

```
1: Soit les nœuds  $i, j$  et SB la station de base ;
2: Soit  $V_j$  la table des voisins des nœuds  $i$  ;
3: Soit  $mon.num\_niveau$  : variable pour stocker le numéro de niveau assigné
   à chaque nœud ;
4: Soit  $mon.num\_zone$  : variable pour stocker le numéro de zone assigné à
   chaque nœud ;
5: Soit  $R_i$  : la portée du nœud  $i$  ;
6: Soit  $K$  : des compteurs ;
7: Soit  $s, d$  : distance par rapport à la station de base ;
8:  $K \leftarrow 1$  ; // Initialisation du compteur  $K$  à 1
9:  $s \leftarrow 0, d \leftarrow 25$  ; // Initialisation par rapport à l'axe d'abscisse ou l'axe
   coordonné ;
10: //Création des niveaux
11: while (la distance ( $i, SB$ )  $> K * R_i$ ) do
12:   if ( $Ni.y \geq s$  et  $Ni.y < d$ ) // Divisions en niveaux then
13:     //Incrémenter la valeur du compteur  $K$ 
14:      $K \leftarrow K + 1$  ;
15:      $s \leftarrow s + 25 ; d \leftarrow d + 25$  ; // augmenter le niveaux de 25 mètre
16:   end if
17: end while
18: //Mise à jour sur la valeur du  $num\_niveau$ 
19:  $Mon.num\_niveau \leftarrow mon.num\_niveau + K$  ;
20: //Création des zones
21: while (la distance ( $i, SB$ )  $> K * R_i$ ) do
22:   if ( $Ni.x \geq s$  et  $Ni.x < d$ ) // Divisions en zones then
23:     //Incrémenter la valeur du compteur  $K$ 
24:      $K \leftarrow K + 1$  ;
25:      $s \leftarrow s + 25 ; d \leftarrow d + 25$  ; // augmenter la zone de 25 mètre
26:   end if
27: end while
28: //Mise à jour sur la valeur du  $num\_zone$ 
29:  $Mon.num\_zone \leftarrow mon.num\_zone + K$  ;
30: //Le nœud  $i$  informe ses voisins en diffusant un paquet Hello contenant
   ses informations.
31: Diffuser le paquet Hello ; Lors de la réception d'un paquet Hello par  $j$ 
   depuis le nœud  $i$ 
32: if ( $i \notin V_j$ ) then
33:   Ajouter une entrée pour  $i$  dans  $V_j$  ;
34: else if then
35:   Mettre à jour les informations de  $i$  ;
36: end if
```

---

• **Explication de l'algorithme**

Après la déclaration des variable, la station de base diffuse différentes puissances de single avec la fonction (la distance (i, SB) > K \* Ri) en incrémentant le K, et avec la fonction suivent (Ni.y >= s et Ni.y < d) qui nous permet de diviser notre zone d'intérêt en niveaux de même tailles avec la l'incrimination de 25 mètre le s et le d. Le même principe est utiliser pour former les zones mais a la différence de la précédant la station de base fait un balayage vertical.

2. **Phase d'élection des CHs**

Les nœuds capteurs qui se situent dans la même zone et même niveau et possèdent un même identifiant ID\_niveau et même identifiant ID\_zone envoient un message pour commencer l'élection du cluster. Le message contient l'identifiant du nœud et sa note Ni calculée suivant les deux métriques, énergie résiduelle et distance par rapport à la station de base, calculer suivant la formule (3.2).

$$Ni = \frac{Er}{\sum_{\kappa \in \text{Membre } C} Er_{\kappa}} + \frac{Dist\_SB}{\sum_{\kappa \in \text{Membre } C} Dist\_SB} \dots (3.2)$$

Où :

Er : Energie résiduelle du nœud capteur ;

Er<sub>k</sub> : Energie résiduelle des voisins du nœud i ;

Dist\_SBi : Représente la distance sépare le nœud i de la SB ;

Dist\_SBk : La distance des voisins du nœud i ;

Membre C : Les nœuds membre du groupe.

A la réception des messages par le CH\_T, celui-ci désigne comme CH le nœud possédant la note Ni maximale, en cas d'égalité, un autre paramètre est pris en compte, ce dernier est l'identité minimale des nœuds.(figure 3.3).

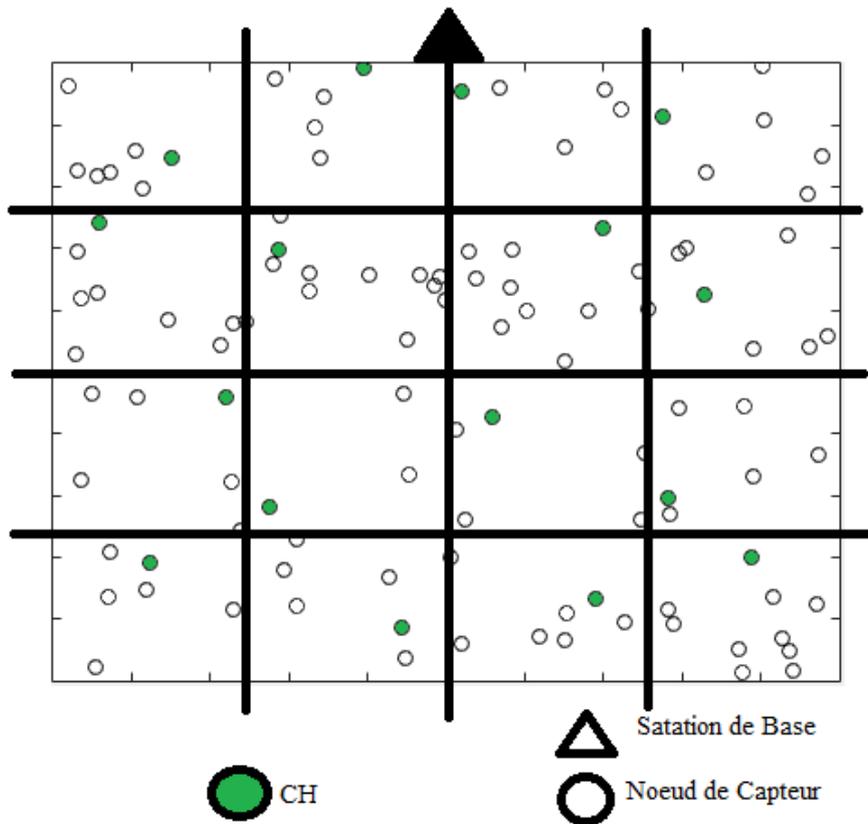


FIGURE 3.3 – Election des CHs.

L'élection des CHs et CH\_T est détaillée par l'algorithme d'élection des CHs et CH\_T (algorithme 3.2).

---

**Algorithm 2** Algorithme d'élection des CHs et CH\_T.

---

Soit  $i$  un nœud dont le numéro de couche égale à  $C_i$  et numéro de zone  $Z_i$ ;

2: Soit ECH, ECH\_T : variables booléennes utilisées pour indiquer le statut du nœud;

Soit  $\max 1\_N_i$  : fonction qui retourne vrai si la note  $N_i$  est maximal ou minimal;

4: Soit  $\max 2\_N_i$  : fonction qui retourne vrai si la note  $N_i$  est maximal ou minimal;

Fonction d'élection CH () // pour un nœud  $i$

6: **if** ( $N_i \leftarrow \max 2\_N_i$ ) **then**  
    ECH\_T  $\leftarrow$  vrai;

8: **else if** **then**  
    // Ce n'est pas un CH\_T

10: Mettre à jour les informations de  $i$ ;  
    **if** ( $N_i > \max 1\_N_i$ ) **then**

12:     ECH  $\leftarrow$  vrai;  
    **else if** **then**

14:     // Ce n'est pas un Cluster Head  
       ECH  $\leftarrow$  faux;

16: **end if**  
**end if**

---

### 3. Phase de transmission

Cette phase assure la collecte et l'acheminement des données vers la SB, en utilise le multi sauts pour minimiser l'énergie des nœuds capteurs et augmenter la durée de vie du réseau. Afin d'éviter les collisions des données lors des transmissions des données par les nœuds capteurs au moment des captures, nous utilisons l'ordonnancement avec TDMA (*Time Division Multiple Access*), qui alloue aux nœuds membres d'un cluster des portions de temps (*time slot*) pour émettre leurs données sans risque de collision (*voir figure 3.4*). Le calendrier des temps est envoyé par le nœud capteur CH\_T à tous les membres du groupe pour le tour actuel.

Au moment de capture de données par un capteur, il envoie au CH du cluster puis ce dernier fait une agrégation des paquets reçus afin d'éviter une surconsommation d'énergie.

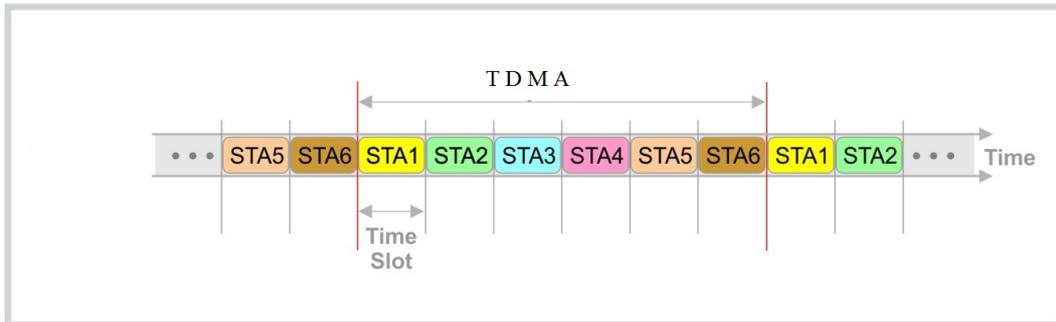


FIGURE 3.4 – Ordonnancement TDMA [13].

#### • Algorithme de recherche de relais

La station de base (*SB*) déclenche cette phase par un envoi d'un message de découverte (*en utilisant une portée étendue*), seuls les CHs (*clusterheads*) traitent ce type de message. Afin d'assurer un chemin optimal, nous nous basons sur deux métriques qui sont :

- **Cout** : sert à déterminer par la suite les meilleurs nœuds à utiliser pour acheminer les données selon l'énergie résiduelle des nœuds capteurs.
- **Dist\_BS** : sert à éviter les boucles lors de déroulement de la phase de recherche de relais. Il sert à estimer la distance séparant les CHs de la

station de base. En effet, selon la force du signal reçu et connaissant la portée utilisée, le chef Chi estime la distance d1 le séparant de l'émetteur du message de découverte (CHj).

Si  $(d1 + dist_{BS}(CHj))$  est inférieur à  $dist_{BS}(CHI)$  (*initialement de valeur infinie*), Chi traite le message reçu et met à jour sa valeur  $dist_{BS}$  qu'il envoie avec son message de découverte.

● **Principe fonctionnement de l'algorithme**

La station de base diffuse un message de découverte avec comme paramètres son identifiant (*ID*), les champs *cout* et  $dist_{BS}$  sont initialisés à 0.

$$Cout(BS) = 0, dist_{BS} = 0 \dots \dots (3.3)$$

Les CHs ne font suivre ces messages que si leurs émetteurs sont plus proches que celui de la station de base selon le paramètre  $dist_{BS}$ . A la réception d'un message de découverte, le Clusterhead CHj met à jour le *cout* total du chemin. Ainsi si le message est reçu du Chi, alors

$$C_{CH_i,CH_j} = cout(CH_i) + \frac{(R_r(CH_j))}{E_{init}} \dots \dots (3.4)$$

Nous utilisant les probabilités de sélection des nœuds relais, et après le calcul du CCHj, CHi, tous les CHs voisins sont ajoutés à la table de relais RTj du chef CHj.

$$RT_j = \{i, CH_i \text{ voisins plus proche de la station de base.} \dots \dots (3.5)$$

Cette approche se base sur l'utilisation des probabilités pour le choix du chemin. Où CHi va assigner une probabilité à chacun de ces chefs voisins dans sa table de relais RTj. Cette probabilité est inversement proportionnelle au *cout*.

$$P_{CH_i,CH_j} = \frac{1}{\sum_{\in \kappa RT_j} (C_{CH_j,CH_i})} \dots \dots (3.6)$$

Cette CH dispose par la suite d'un nombre de CHs relais à travers lesquels il peut acheminer les données vers la station de base. Une fois l'échange de message de découverte s'achève, le chef CH<sub>j</sub> calcule le coût moyen pour atteindre la station de base.

$$Cout(CH_j) = \sum_{i \in RT_j} P_{CH_j, CH_i} * C_{CH_j, CH_i} \dots \dots (3.7)$$

Ce coût moyen est mis dans le champ cout et est transmis avec le message de découverte. La fin de cette phase est marquée par l'expiration d'un temporisateur t dont la valeur est choisie de manière à assurer que les messages de découverte circulent tout au long du réseau et dont la valeur dépend du diamètre du réseau.

### 3.3 Conclusion

Dans ce chapitre nous avons détaillé un nouveau protocole (*EEPCZ*) en exposant son principe de fonctionnement, les mécanismes utilisés pour la structuration du réseau ainsi que les étapes de communication des données vers la station de base sont expliquées. Notre protocole résout le problème du point chaud en utilisant le mécanisme de recherche des relais, et le problème de la transmission directe par le multi sauts.

Dans le chapitre suivant, on va simuler notre protocole sur MATLAB.

# Simulation sur MATLAB

---

## 4.1 Introduction

Dans ce chapitre, nous réaliserons une simulation de notre protocole EEPSCZ sur MATLAB, le but de cette partie est de valider de manière générale la présentation théorique de protocole décrite précédemment.

## 4.2 Simulation et analyse de performances

Une simulation consiste à gérer le temps ainsi que les actions qui sont liées aux différents instants d'un système réel et à faire fonctionner abstraitement le modèle qui représente ce système.

La simulation connaît de nos jours un essor considérable. Ceci est dû aussi bien à l'intérêt théorique que présente la modélisation des systèmes simulés, qu'au besoin croissant de simuler, par ordinateur, des réalisations de plus en plus complexes. Nous constatons, en effet, que les conditions d'expérimentation sont aujourd'hui toujours plus difficiles et plus coûteuses ; nous avons recours donc de plus en plus à l'expérimentation indirecte, notamment aux techniques de simulation. Celles-ci consistent à représenter la réalité simulée sur ordinateur, en mettant en évidence les progrès spectaculaires de l'infographie, à donner à celle-ci des moyens de perception réalisés artificiellement, puis à simuler son fonctionnement [29].

Nous distinguons plusieurs modèles de simulation selon qu'ils soient déterministes ou aléatoires, continus ou discrets. Pour expérimenter la solution proposée, nous avons choisi d'utiliser un outil de simulation qui est MATLAB. Cette décision a été prise après l'étude des simulateurs de RCSFs existants

(*NS-2*, *TOSSIM*, *OPNET*), qui étaient pour la plupart beaucoup trop lourds, difficile à maîtriser dans des délais aussi courts et qui sont peu adaptés à nos besoins.

### 4.3 Aperçu sur les simulateurs des réseaux de capteurs

Ils existent plusieurs simulateurs dédiés pour les réseaux de capteurs :

- **OPNET (*Optimum Network Performance*)** : comme NS, OPNET est un simulateur à événements discrets. C'est un outil très puissant pour la simulation et l'évaluation de performances des réseaux. Il permet aussi à l'utilisateur de construire ses propres modèles des plus simples aux plus complexes [28] ;

- **NS-2/NAM (*Network Simulator 2*)** : c'est un simulateur développé à Lawrence Berkeley National Laboratory (*LBNL*). Le langage de base de NS est le C++. L'outil NAM (*Network Animator*), associé au simulateur NS, permet de visualiser des animations de la simulation (*transfert des paquets d'un nœud à un autre, taille des paquets, etc.*) ;

- **TOSSIM** : est le simulateur de TinyOs créé par l'université de Berkeley. Il permet de simuler le comportement d'un capteur (*envoi/réception de messages via les ondes radios, traitement de l'information, etc.*) au sein d'un réseau de capteurs. Pour une compréhension moins complexe de l'activité d'un réseau, TOSSIM peut être utilisé avec une interface graphique, TinyViz, permettant de visualiser de manière intuitive le comportement de chaque capteur au sein du réseau.

### 4.4 Choix de MATLAB

MATLAB est un logiciel de calcul numérique produit par MathWorks. Il est disponible sur plusieurs plateformes. MATLAB est un langage simple et très efficace, optimisé pour le traitement des matrices, d'où son nom. Pour le calcul numérique, MATLAB est beaucoup plus concis que les vieux langages (*C, Pascal, Fortran, Basic*) et pour la programmation, il optimise le code des programmes en utilisant des fonctions pré-définies. Nous pouvons traiter la matrice comme une simple variable.

MATLAB contient une interface graphique puissante, et on peut l'enrichir en ajoutant des « boîtes à outils » (*toolbox*) qui sont des ensembles de fonctions supplémentaires, profilées pour des applications particulières (*traitement de signaux, analyses statistiques, optimisation, etc.*).

MATLAB contient également un langage de programmation de haut niveau dans lequel nous retrouvons la majorité des concepts des langages de programmation modernes (*types Pascal et C.*). L'ordre d'exécution des instructions est déterminé par des structures de contrôle. Il permet aussi la création de fonctions et distingue les données locales des données globales. Ces avantages ont rendus de MATLAB, un langage de programmation et de simulation très sollicité [21].

## 4.5 Métriques de performances

Afin de mesurer l'efficacité énergétique de notre protocole EEPSCZ, également on va se focaliser essentiellement sur les points suivants :

- **Consommation d'énergie** : dans les réseaux de capteurs sans fil, la plus part des approches visent à minimiser la consommation d'énergie et prolonger la durée de vie du réseau, afin de remédier au problème de la charge de batterie de ses nœuds capteurs à cause de leurs tailles et de leurs zone de déploiement. De ce fait, la contrainte d'énergie est le paramètre le plus étudié lors de la validation.

- **Durée de vie du réseau** : la durée de vie du réseau comme étant la durée d'épuisement des nœuds capteurs par unité de temps jusqu'à la mort du premier nœud, de même pour un pourcentage de 25% et 50%. Enfin pour le dernier nœud mort dans le réseau.

## 4.6 Etapes de réalisation

Les étapes de réalisation sont les suivantes :

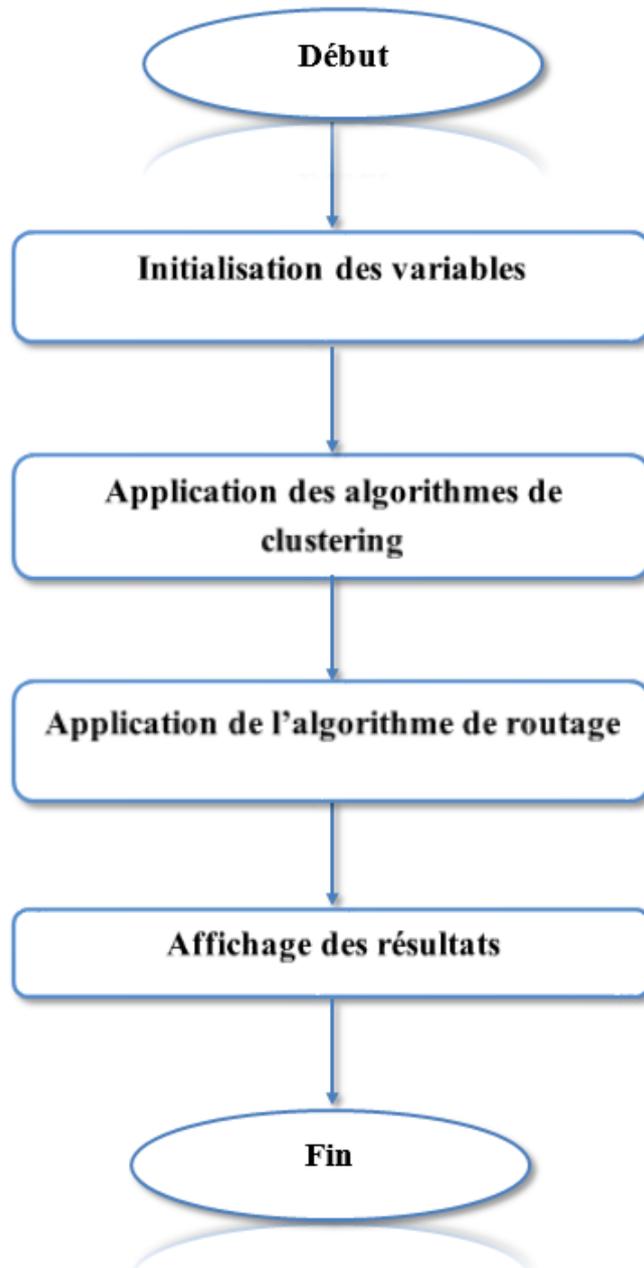


FIGURE 4.1 – *Étapes de simulation.*

• **Initialisation des variables** : cette étape correspond à la déclaration des variables globales (*nombre de capteurs, surface du terrain simulé, nombre de tests et d'exécution à réaliser...*), leur initialisation, la création des capteurs (*portée de transmission, capacité énergétique, capacité mémorielle, capacité calculatoire des capteurs*) et leur déploiement aléatoire sur le terrain simulé. Pour simuler sous MATLAB notre protocole, le déploiement des nœuds est fait d'une manière aléatoire sur une surface de capture de  $(100 \times 100)$  m<sup>2</sup>. Chaque capteur dans le réseau est représenté par ses coordonnées  $(x, y)$ .

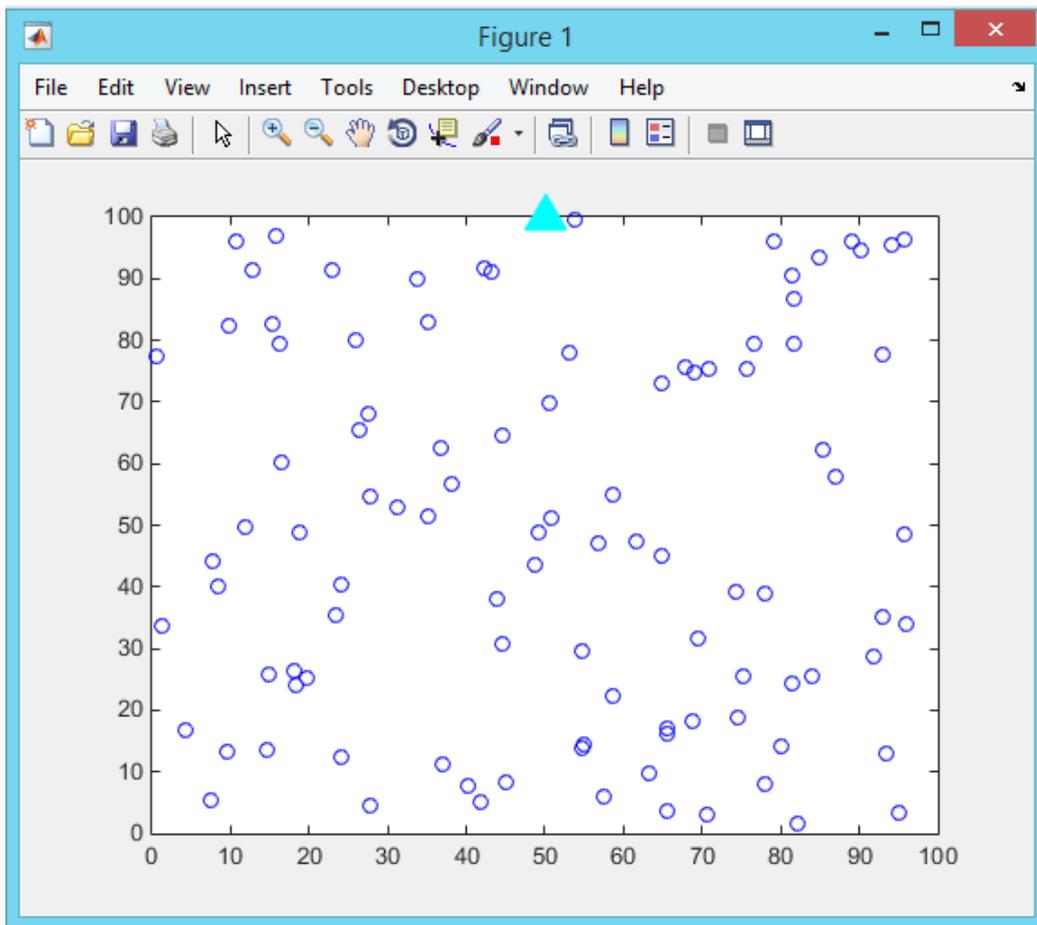


FIGURE 4.2 – *Déploiement des nœuds.*

• **Application de algorithme de clustering** : cette étape concerne la détermination des groupes (*clusters*) grâce à la station de base et des différents messages circulant dans le réseau (*vecteur de données, note...*).

- **Application de l'algorithme de routage :** cette étape correspond à la station sélection du chemin de routage de l'information vers la station de base, à la détection d'un événement à partir d'un nœud capteur quelconque. Une exécution des différentes phases de l'algorithme (*EEPSCZ*) implémentés.
- **Affichage des résultats :** les courbes obtenues par rapport à notre protocole implémentés, selon les critères d'évaluation de performances choisis (*Moyenne d'énergie consommée, Durée de vie du réseau*).

## 4.7 Modèle de simulation

Le système considéré est basé sur l'approche événementielle. La simulation par événements discrets [16] désigne la modélisation d'un système réel tel qu'il évolue dans le temps, par une représentation dans laquelle les grandeurs caractérisant le système (*variables*) ne changent qu'en un nombre fini ou dénombrable de points isolés dans le temps. Ces points sont les instants où se passent les événements.

Dans ce qui suit, nous détaillerons le modèle de simulation du système considéré.

### 4.7.1 Description du système

Notre système représente une zone de capture d'une surface de  $(100*100)$  m<sup>2</sup> comportant 150 capteurs et une station de base. De ce fait, les entités de notre système, ainsi recensées, sont :

- **Station de base :** elle est responsable de l'envoi des requêtes et de la collecte des données du réseau.
- **Nœuds capteurs :** ils servent à détecter les événements et participent au routage des informations dans le réseau.

Le modèle réseau utilisé consiste en un ensemble de capteurs sans fil, immobiles, aléatoirement dispersés sur le terrain simulé et présentant des caractéristiques différentes en mémoire et en puissance de traitement et dont l'énergie initiale de chacun d'entre eux est fixée à 1 joule.

### 4.7.2 Modèle d'énergie

Le modèle radio (*voir figure 4.3*) proposé par Heinzelman et All [11]. A été utilisé pour calculer l'énergie consommée en émission et en réception des messages. Selon ce modèle l'énergie consommée pendant la transmission

( $E_{Tx}$ ) d'un message de  $s$  bits est donnée par :

$$E_{Tx} = E_{elec} * s + E_{amp} * s * d^2 \dots \dots (4.1)$$

Alors que l'énergie consommée pendant la réception ( $E_{Rx}$ ) d'un message de  $s$  bits est donnée par :

$$E_{Rx} = E_{elec} * s \dots \dots (4.2)$$

Où :

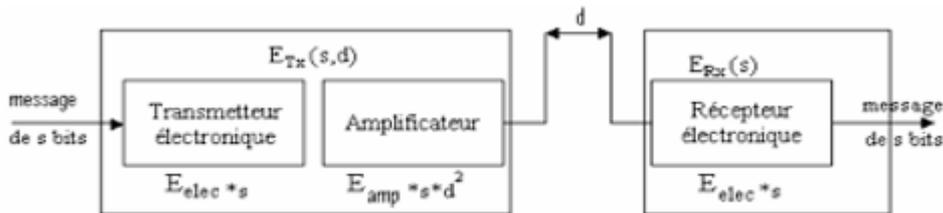


FIGURE 4.3 – Modèle d'énergie [11].

- $E_{Tx}$  est l'énergie consommée en émission.
- $E_{Rx}$  est l'énergie consommée en réception.
- $E_{elec}$  est l'énergie électronique (*énergie consommée par le transmetteur électronique*) dont la valeur est à 50 nJ/bit.
- $E_{amp}$  est l'énergie nécessaire pour l'amplification (*l'énergie consommée par l'amplificateur de l'émetteur*) dont la valeur est fixée à 50 nJ/bit.
- $s$  est la taille en bit d'un paquet de données.
- $d$  est la distance en mètres entre l'émetteur et le récepteur d'un paquet de données.

L'énergie nécessaire à l'agrégation des données est fixée à  $EDA = 5$  nJ/bit/signal.

### 4.7.3 Variable descriptive du système

Les différentes variables utilisées dans notre système sont illustrées dans le tableau suivant :

Définition de la variable	Nom de la variable	Type	Unité de mesure
Energie consommée lors de l'émission d'un message	$E_{Tx}$	Réel	Joule
Energie consommée à la réception d'un message	$E_{Rx}$	Réel	Joule
Energie consommée lors du traitement d'un message	Et	Réel	Joule
Position d'un capteur	(x, y)	(Réel, réel)	(Mètre, mètre)
Distance entre deux capteurs	d	Réel	Mètre

TABLE 4.1 – Variable descriptive du système.

- **Energie résiduelle** : elle est exprimée par la différence entre l'énergie courante et l'énergie consommée par un capteur.
- **Coordonnées d'un capteur sur un plan 2D** : c'est l'affectation de deux valeurs aléatoires ( $x$  et  $y$ ), comprises entre 0 et la taille du terrain, à chaque capteur sur un plan a deux dimensions.

## 4.8 Evaluation de performances

Dans ce qui suit, nous allons présenter et analyser les résultats de simulations obtenus suivant les métriques discutées précédemment. Pour les simulations réalisées, le paramètre variable utilisé est la portée de chaque capteur, augmenter la portée d'un capteur est alors équivalent à augmenter la densité du réseau ; il aurait donc été équivalent d'utiliser la surface du réseau pour variable.

### 4.8.1 Moyenne d'énergie consommée

On remarque clairement dans le graphe de la figure 4.4, une baisse considérable d'énergie consommée entre 1 (u) et 283 (u), qui est dû à la génération des clusters d'une part et à la création des clusters heads temporaires et des clusters heads d'une autre part.

Par contre de 283 (u) à 2400 (u), on remarque que la moyenne d'énergie diminue de façon progressive en fonction du temps jusqu'à son épuisement grâce à l'équilibrage dans la distribution de la charge qu'a présenté EEPSCZ entre les CHs des différentes couches et techniques utilisées dans la formation des clusters ainsi, que le choix des nœuds CHs, ces techniques prennent en compte les paramètres influant sur la bonne gestion de la ressource énergétique à savoir l'énergie résiduelle et la distance par rapport à la station de base.

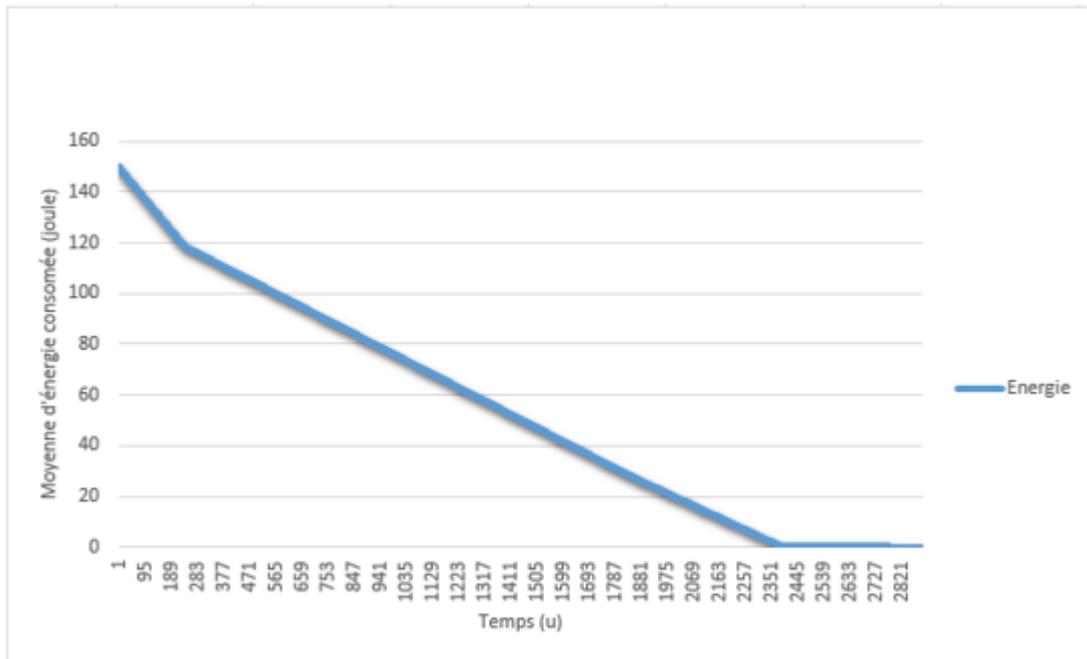


FIGURE 4.4 – Moyenne d'énergie consommée.

### 4.8.2 Durée de vie du réseau

Pour pouvoir étudier le fonctionnement des nœuds déployés, nous avons suivi l'évolution de 150 nœuds. La figure 4.5 présente la durée de vie du réseau en fonction du temps de notre protocole (*EEPSCZ*). Ce dernier a pour but de maximiser la durée de vie des nœuds capteurs et par conséquent, celle du réseau entier.

On constate que le nombre de nœud reste stable (*150 nœuds*) jusqu'atteindre 208 (u), qui se décline de vingt nœuds, à cause de la charge (*captage, émission et réception de données*) de ces dernier, en revanche le nombre de nœuds ce stabilise (*entre 120 et 130 nœuds*) étant donné l'équilibrage dans la consommation d'énergie des nœuds, qui permet ainsi d'allonger leur durée de fonctionnement jusqu'à l'effondrement du réseau.

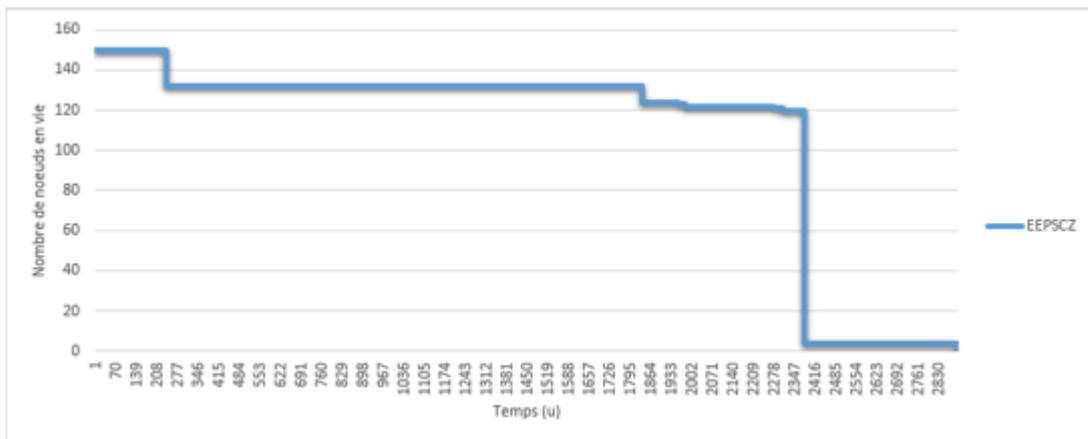


FIGURE 4.5 – *Durée de vie moyenne du réseau.*

## **4.9 Conclusion**

Dans ce chapitre, nous avons présenté la technique de simulation de notre protocole de routage EEPSCZ sur MATLAB ainsi que les résultats obtenus après implémentation, il a été évalué en fonction de la moyenne d'énergie consommée, et la durée de vie du réseau.

# Conclusion Générale

Les réseaux de capteurs sans fil ont connu un grand succès dès leur apparition dans tous les domaines (*militaire, médicale, environnementale, etc.*) grâce à leurs diverses fonctionnalités. Cependant, la conception de ces réseaux doit satisfaire quelques contraintes, parmi ces contraintes la limitation de la ressource énergétique.

Une problématique majeure, est la maîtrise de l'énergie consommée par chaque nœud capteur. En effet, chaque nœud capteur est muni d'une ressource énergétique (*généralement une batterie*), pour alimenter tous ses composants. Cependant, en raison de sa taille réduite, la ressource énergétique dont il dispose est limitée, et généralement irremplaçable. Des lors, l'énergie est la ressource la plus précieuse dans un réseau de capteurs, parce qu'elle influe directement sur la durée de vie des capteurs et du réseau en entier. Aussi, pour qu'un réseau de capteurs ait une longévité maximale, il faut que la consommation d'énergie soit prise en compte à tous les niveaux de l'architecture réseau.

Le routage est un service très important dans les réseaux de capteurs, il doit permettre l'arrivée des données à un nœud puits avec le minimum de pertes et de dissipation d'énergie. Dans ce travail, nous nous sommes intéressés à la problématique de l'économie d'énergie lors du routage dans les réseaux de capteurs sans fil, et l'amélioration de la durée de vie de ce type de réseaux. Nous avons proposé un protocole de routage à basse consommation d'énergie EEPSCZ, pour les réseaux de capteurs sans fil, nous l'avons intégré dans le simulateur, ensuite simuler et analyser les résultats qui sont la moyenne d'énergie consommée et la durée de vie du réseau.

En guise de perspectives :

Comme très souvent, le travail que nous avons effectué ouvre de nouvelles perspectives, permettant de le compléter de différentes manières.

- Comparée le protocole proposé avec le protocole EEPSC.
- Essayer d'obtenir les résultats de la simulation en implémentant le nouveau protocole sous d'autres simulateurs comme Network Simulator 2, OPNET, GloMoSim... etc.
- Enfin, mettre en pratique le protocole EEPSCZ (*Energy Efficient Protocol With Static Clustering Zones*) dans une application réelle des réseaux de capteurs.



# Bibliographie

- [1] Ameer Ahmed Abbasi and Mohamed Younis. A survey on clustering algorithms for wireless sensor networks. *Computer communications*, 30(14) :2826–2841, 2007.
- [2] M.Djoudi et T.Mboukem A.Fares. Développement d’une bibliothèque de capteurs. 2003.
- [3] M.Djoudi et T.Mboukem A.Fares. Optimisation par colonie de fourmis. 2006.
- [4] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks : a survey. *Wireless communications, IEEE*, 11(6) :6–28, 2004.
- [5] Kamal Beydoun. *Conception d’un protocole de routage hiérarchique pour les réseaux de capteurs*. PhD thesis, Université de Franche-Comté, 2009.
- [6] Florent Brissaud, Dominique Charpentier, Anne Barros, and Christophe Bérenguer. Capteurs intelligents : Nouvelles technologies et nouvelles problématiques pour la sûreté de fonctionnement. In *Maîtrise des Risques et de Sûreté de Fonctionnement, Lambda-Mu 16*, pages 3A–2. Institut pour la Maîtrise des Risques, 2008.
- [7] Mickael Cartron. *Vers une plate-forme efficace en énergie pour les réseaux de capteurs sans fil*. PhD thesis, 2006.
- [8] Boubiche Djallel Eddine. Memoire de magistere en informatique intitulé protocole de routage pour les réseau de capteur sans fil.
- [9] Antoine Gallais, François Ingelrest, Jean Carle, and David Simplot-Ryl. Maintien de la couverture de surface dans les réseaux de capteurs avec une couche physique réaliste. In *Colloque Francophone sur l’Ingénierie des Protocoles (CFIP 2006)*, 2006.
- [10] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4) :660–670, 2002.

- [11] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on*, pages 10–pp. IEEE, 2000.
- [12] Wendi Rabiner Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174–185. ACM, 1999.
- [13] <http://www.accton.com/NewsPage.asp?sno=89/> .
- [14] <http://www.alertsystems.org/>.
- [15] Lyes Khelladi and Nadjib Badache. Les réseaux de capteurs : état de l’art. *Rapport de recherche Laboratoire des systèmes informatique, Faculté électronique et informatique Bab Ezzouar-Algérie*, 2004.
- [16] N. KHOULALENE. Regroupement avec équilibrage de charge dans les réseaux de capteur sans fil, thèse de magister 2009 université abderrahmane mira de bejaia.
- [17] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker. Modeling data-centric routing in wireless sensor networks. In *IEEE infocom*, volume 2, pages 39–44, 2002.
- [18] Stephanie Lindsey and Cauligi S Raghavendra. Pegasus : Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings, 2002. IEEE*.
- [19] Abdallah Makhoul. *Réseaux de capteurs : localisation, couverture et fusion de données, Université de Franche-Comté (LIFC)*. PhD thesis, 2008.
- [20] Khelifi Manel. Optimisation de la consommation de l’énergie et maximisation de la durée de vie des réseaux capteurs sans fil, mémoire pour l’obtention du grade de magistère en informatique, université abderrahmane mira de béjaia. 2007.
- [21] A. MANUEL. *Atelier d’outils informatiques pour la physique (Info-Phys), Eléments de MATLAB*. PhD thesis, Département de la Physique de la Matière Condensée, université de Genève.
- [22] Naourez Mejri and Farouk Kamoun. Algorithme de routage hiérarchique mheed à plusieurs sauts pour les grands réseaux de capteurs. In *4th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications*, pages 1–7, 2007.
- [23] CHARIF Meryem. Sécurisation du protocole de routage hiérarchique leach dans les réseaux de capteurs sans fil. Projet De Fin d’Etudes Master, Université Abou Bakr Belkaid– Tlemcen, 2013.

- [24] Zitouni Rafik. Mémoire de magistère : Routage à basse consommation d'énergie dans les réseaux de capteurs sans fil. 2005.
- [25] Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B Srivastava. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2) :40–50, 2002.
- [26] RITIME. Memoire de fin d'étude pour l'obtention du diplome d'ingénieur d'état en informatique réseau de capteurs sans fil (rcsf).
- [27] M Yasser ROMDHANE and M Nabil TABBENE. Evaluation des performances des protocoles s-mac et directed diffusion dans les réseaux de capteurs. *Rapport de fin d'étude d'ingénieur, Ecole supérieure de communication de Tunis*, 2007.
- [28] Yessad Samira. *La couche MAC avec contraintes d'énergie et d'équité dans les réseaux de capteurs*. PhD thesis, Université Abderrahmane Mira de Béjaia, 2006.
- [29] Lakshminarayanan Subramanian and Randy H Katz. An architecture for building self-configurable systems. In *Mobile and Ad Hoc Networking and Computing, 2000. MobiHOC. 2000 First Annual Workshop on*, pages 63–73. IEEE, 2000.
- [30] My T Thai, Feng Wang, David Hongwei Du, and Xiaohua Jia. Coverage problems in wireless sensor networks : designs and analysis. *International Journal of Sensor Networks*, 3(3) :191–200, 2008.
- [31] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2001.
- [32] Ossama Younis and Sonia Fahmy. Heed : a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3(4) :366–379, 2004.
- [33] Amir Sepasi Zahmati, Bahman Abolhassani, Ali Asghar Beheshti Shirazi, and Ali Shojaee Bakhtiari. An energy-efficient protocol with static clustering for wireless sensor networks. *International Journal of Electronics, Circuits and Systems*, 1(2) :135–138, 2007.



# Résumé

Les nœuds capteurs d'un réseau de capteur sans fil (*RCSF*) sont alimentés par des batteries épuisables à durée de vie limitée et non rechargeable. Cependant, l'acheminement des données vers un centre de traitement nécessite des mécanismes de conservation d'énergie ; afin de prolonger la durée de vie du réseau. L'objectif de ce travail est de proposer une nouvelle approche de routage à basse consommation énergétique tout en améliorant le protocole hiérarchique EEPSC. La première partie de notre travail consiste à étudier les généralités des réseaux de capteurs et la communication dans les réseaux de capteurs. La deuxième partie qui consiste à étudier le routage hiérarchique dans les réseaux de capteurs, ainsi qu'une étude de quelques protocoles de routage hiérarchique. Dans la troisième partie nous avons proposé un nouveau protocole nommée EEPSCZ (*Energy Efficient Protocol With Static Clustering Zones*), tout en détaillant son principe et son fonctionnement, à la fin nous avons simulé le protocole de routage proposé, sur MATLAB et faire une analyse des résultats obtenus.

**Mots clés** Réseaux de capteurs sans fil, Protocole de routage hiérarchique, consommation énergétique, EEPSC, EEPSCZ.

# Abstract

Sensor nodes of a wireless sensor network (*WSN*) are powered by batteries exhaustible limited lifespan and not rechargeable. However, the flow of data to a processing center requires energy conservation mechanisms ; to the extended life of the network. The objective of this work is to propose a new routing approach to low energy consumption while improving the hierarchical protocol EEPSC. The first part of our work is to study the general sensor networks and communication in sensor networks. The second part of studying the hierarchical routing in sensor networks, as well as a study of some hierarchical routing protocols, in the third part we proposed a new protocol called EEPSCZ (*Energy Efficient Protocol With Static Clustering Zones*), while detailing its principle and its operation at the end we simulated the proposed routing protocol, on MATLAB and we give an analysis of the results.

**Keywords** Networks of wireless sensors, hierarchical routing protocol, energy consumption, EEPSC, EEPSCZ.