

République algérienne démocratique et populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement supérieur et de la recherche scientifique  
المركز الجامعي لعين تموشنت  
Centre Universitaire Belhadj Bouchaib d'Ain-Temouchent  
Institut de Technologie  
Département de Génie Mécanique



Projet de fin d'études  
Pour l'obtention du diplôme de Master en :  
**Domaine** : TECHNOLOGIE  
**Filière** : GENIE électrique  
**Spécialité**: Réseau et Télécom

Thème

*Choix de Protocoles De Routage Hiérarchiques Dans Les Réseaux De Capteurs Sans Fil Pour Une Application Domotique*

Présenté Par :

- 1) Khalladi Fatima Zohra
- 2) Rezigui Hayet

Devant les jurys composés de :

<b>BENOSMANE M. Mourad</b>	MCB	C.U.B.B (Ain Temouchent)	Président
<b>BENDIMERAD M. Karim</b>	MAA	C.U.B.B (Ain Temouchent)	Encadrant
<b>BEMMOUSSAT Chemseddine</b>	MCB	C.U.B.B(Ain Temouchent)	Examineur

# Remerciements

*"La crainte de l'Éternel est le commencement de la sagesse"*

*Tout d'abord, merci pour le bon « DIEU » qui nous a donné la volonté et le*

*Courage pour le parachèvement de ce travail.*

*On remercie notre encadreur Mr BENDIMERAD.*

*Un spécial remerciement pour Monsieur BEMMOUSSAT et DEBBAL qui n'ont pas lésiné sur les moyens et sur leurs temps pour nous apporter leurs aides précieuses et qui nous a permis de mener à bien ce travail.*

*Et à tous nos Professeurs, nous tenons à leur exprimer toutes nos reconnaissances pour leurs dévouements, la confiance qu'ils nous ont accordée, leur rigueur et la qualité des commentaires et suggestions dont ils nous ont fait part.*

*Nous remercions tous les ami(e)s de près et de loin qui ne nous ont pas privés de leurs soutiens, leurs bonnes humeurs quotidiennes.*

# DÉDICACE

❖ *Je remercie le seigneur de m'avoir donné le courage et la volonté de mener à terme ce travail.*

➤ *Je dédie ce Modeste travail :*

❖ *A ma très chère 'MAMAN' en signe d'amour, de reconnaissance et de gratitude pour tous les soutiens et les sacrifices dont elle a fait preuve à mon égard.*

❖ *A la mémoire de 'Mon Père' aucune dédicace ne pouvait exprimer l'amour et le respect qui j'ai toujours en pour vous.*

❖ *A mes sœurs : Fatima et Ibtisem*

❖ *A mes frères : Kaddour, Mohamed et Zoheir*

❖ *A mes petites anges : Aya, Adel, Samado, Tasnimo, Yasmina.*

❖ *A tous les membres de ma famille Rezigui et Yaichi*

❖ *A mon binôme WARDA qui m'a supporté durant le travail.*

❖ *A ceux que j'aime beaucoup mes chères copines :*

*BibaHabouba, Fatima, Amina, Sara, Khaoula, Houria,*

*AminaS,kawtherS,Khadija,kawther,Fatna,Amel*

❖ *Et mes Amis : Saido, Miloud, Abdellatif, Med Fekharji,Karim,Jtn,Ismail.*

❖ *Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.*

**HAYET**

# Dédicace

*Du profond de mon cœur, je dédie ce travail à tous ceux qui me sont chers,*

## *A MA CHÈRE MÈRE*

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

## *A LA MÉMOIRE DE MON PÈRE*

*Ce travail est dédié à mon père, décédé trop tôt, j'aurais aimé qu'il soit présent aujourd'hui. J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'une fille qui a toujours priée pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde !*

## *A Mon Cher Frère*

*DIDEN*

*MON Beau Frère*

*ABDELKADER*

*Aux Prunelles De Mes Yeux : WAHIBA, WAFAA*

*Sans Oublié Mon Bébé D'amour MAHDI OMAR*

*A mes meilleurs CHANEZ, SOUMIA, AMINA, AMEL, NARJIMENE, MOUAD, SEIFEDDINE et IMAD et mon binôme REZIGUI HAYET.*

*A Toute Ma Famille Et Toutes Les Personnes Que J'aime.*

*WARDA*

# Résumé

*Les Réseaux de Capteurs Sans Fil (RCSF) constituent une technologie émergente, dotée de potentialités puissantes et créatives. Dans le domaine du bâtiment en général et de la domotique en particulier, il existe un large éventail de travaux qui traite du déploiement des RCSF. Le RCSF est une collection d'un grand nombre de petits nœuds qui agissent en tant que routeurs également. Ces nœuds sont porteurs d'une source d'énergie très limitée, non rechargeable et non-remplaçable, ce qui rend la consommation d'énergie un facteur très contraignant. De ce fait la minimisation de ce facteur devient incontournable dans la prolongation de la vie du réseau. Pendant que les nœuds de capteurs agissent comme des routeurs, la détermination de(s) technique(s) ou protocole(s) de routage joue une fonction clé dans la gestion et le contrôle de la consommation d'énergie. Notre travail s'attèle à apporter une contribution dans le domaine de la modélisation par algorithme et l'analyse par simulation de protocoles de routage hiérarchiques réputés pour leur efficacité énergétique. Plus précisément, notre étude met en opposition le comportement par rapport à des métriques de performance préétablis de deux grandes approches de routage que sont l'approche basée grappe ou « Cluster-based approach » et l'approche basée chaîne ou « Chain-based approach ». Pour se faire, nous nous bornons à exploiter un code algorithmique déjà existant, implémenté sous MATLAB, qui rend compte de la dualité comportementale des protocoles de groupage (LEACH) et de chaînage (PEGASIS) utilisés dans les RCSF, avec des métriques adaptés à l'application domotique : gestion de confort thermique. La finalité étant de dégager une approche évolutive, plus performante, fruit de combinaison entre les deux protocoles.*

**Mots clés:** *RCSF, efficacité d'énergie, protocole de routage hiérarchique, approche basée cluster, approche basée chaîne, LEACH, PEGASIS.*

# Abstract

*Wireless Networks Sensors (WSN) constitute an emergent technology with powerful and creative potential. In the field of building in general and house automation in particular, there has been plenty of works those treat WSN's deployment. WSN is a collection of a large number of small nodes which act as routers also. These nodes carry very limited power source which is non-rechargeable and not-replaceable, which makes energy consumption a significant issue. Energy conservation is a very important issue for prolonging the lifetime of the network. As the sensor nodes act like routers as well, the determination of routing technique plays a key role in controlling the consumption of energy. Our work gets on with making a contribution in both the algorithm-based modeling and the simulation analysis fields of hierarchical routing protocols, rated for their energy efficiency. More specifically, our study puts in opposition the behavior through preset performance metrics of two great routing approaches defined as "Cluster-based approach" and "Chain-based approach". To be done, we limit ourselves to run an already existing algorithmic code, implemented under MATLAB, which give us an account of the behavior al duality of the clustering protocols (LEACH) versus chaining ones (PEGASIS) used in WSN, along with metrics suitable to the house automation application: thermal comfort management. The objective being to give off a scalable, more powerful approach, fruit of two protocols-based combinations.*

**Keywords:** *WSN, energy efficient, hierarchical routing protocol, Cluster-based approach, Chain-based approach. LEACH, PEGASIS.*

A detailed pencil sketch of a lightbulb, oriented horizontally. The bulb is shaded to show its rounded form and the textured surface of the glass. The base of the bulb, with its characteristic screw threads, is rendered with fine lines. The word "SOMMAIRE" is written across the middle of the bulb in a bold, black, serif font. The sketch is set against a light, circular glow that fades into the white background.

***SOMMAIRE***

# SOMMAIRE

Résumé

Abstract

Remerciements

Dédicaces

Liste des Figures

Liste des Tableaux

Liste des Abréviations

Introduction générale.....1

## Chapitre I

### Généralités sur les réseaux de capteurs sans fil & Applications domotiques

I.1 - Introduction..... 4

I.2 - RESEAUX DE CAPTEURS SANS

FIL(RCSF)..... 4

I.2.1 - *Définition* ..... 4

I.2.2 - *Composants d'un nœud de capteur* ..... 4

I.2.3 - *Architecture d'un RSCF*.....5

I.2.4- *Caractéristiques des RCSF*.....6

I.3 - DOMAINES D'APPLICATION DES RCSF.....7

I.3.1 - *Application militaires*.....8

I.3.2 - *Applications à la sécurité*.....8

I.3.3 - *Applications environnementales*.....9

I.3.4 - *Applications médicales*.....9

I.3.5 - *Applications domotiques*.....10

I.3.5.1 - *Description* .....10

I.3.5.2 - *Fonctions de la domotique* .....11

I.3.5.3 - *Application confort thermique*.....12

I.4 - Conclusion.....12



## Chapitre II

### Protocoles de routage : Approches de routage hiérarchique

II.1 -Introduction.....	13
II.2 - LES PROTOCOLES DE ROUTAGE DANS LES RSCF - CLASSIFICATION.....	13
II.3 - ROUTAGE HIERARCHIQUE DANS LES RCSF – EXEMPLES.....	15
II.3.1 -SAR (Sequential Assignment Routing).....	15
II.3.2 -TEEN ( <i>Threshold Sensitive Energy Efficient Sensor Network protocol</i> ).....	16
II.3.3 - Approche à grappe « <i>Cluster-based approach</i> ».....	16
II.3.3.1- LEACH “ <i>Low Energy Adaptive Clustering Hierarchy</i> ”.....	18
II.3.3.1.1- <i>Avantages et inconvénients de LEACH</i> .....	18
II.3.4- Approche à chaîne « <i>Chain-based approach</i> ».....	22
II.3.4.1- PEGASIS.....	23
II.3.4.1.1 - <i>Approche de construction de chaîne</i> .....	23
II.3.4.1.2 - <i>Avantages et inconvénients de PEGASIS</i> .....	26
II.4 -Conclusion.....	27

## Chapitre III

### Caractérisation du modèle de protocole de routage et simulation sous MATLAB

III.1 - Introduction.....	28
III.2- Application Domotique Confort Thermique - Mode opératoire .....	28
III.3- Environnement de simulation .....	29
III.3.1- <i>Présentation de MATLAB</i> .....	30
III.3.2 - <i>Modèle radio</i> .....	31
III.4 - PARAMETRES DE SIMULATION.....	33
III.5- EVALUATION DES PERFORMANCES.....	34
III.5.1 - <i>Consommation d'énergie</i> .....	35
III.5.2– <i>Durée de vie du réseau</i> .....	35

III.6 - Conclusion.....	37
<b>Conclusion générale.....</b>	<b>40</b>
<b>Bibliographie.....</b>	<b>41</b>
<b>Annex</b>	

## Liste des Figures

<i>Figure I.1 Composants d'un nœud capteur .....</i>	<i>5</i>
<i>Figure I.2 Architecture de communication dans un RCSF.....</i>	<i>5</i>
<i>Figure I.3 Architecture pour les réseaux de capteurs.....</i>	<i>6</i>
<i>Figure I.4 Réseaux de capteurs militaires.....</i>	<i>7</i>
<i>Figure I.5 Applications à la sécurité.....</i>	<i>8</i>
<i>Figure I.6 Applications environnementales.....</i>	<i>8</i>
<i>Figure I.7 Ensemble de capteurs dans un corps humain.....</i>	<i>9</i>
<i>Figure I.8 Applications domotiques.....</i>	<i>9</i>
<i>Figure I.9 Facteurs environnementaux du confort thermique.....</i>	<i>11</i>
<i>Figure II.1 Les principaux protocoles de routage dans les RCSF.....</i>	<i>13</i>
<i>Figure II.2 Approche à grappe« Cluster-based approach ».....</i>	<i>17</i>
<i>Figure II.3 Algorithme de routage LEACH.....</i>	<i>19</i>
<i>Figure II.4 Opérations de l'étape d'initialisation.....</i>	<i>20</i>
<i>Figure II.5 Construction de chaînes en utilisant l'algorithme avide.....</i>	<i>22</i>
<i>Figure II.6 Approche de dépassement de jeton.....</i>	<i>23</i>
<i>Figure III.1 Système domotique.....</i>	<i>29</i>
<i>Figure III.2 Modèle de consommation d'énergie.....</i>	<i>32</i>
<i>Figure III.3 Topologie de 25 nœuds aléatoires pour un réseau de (100 x 100) m<sup>2</sup>.....</i>	<i>34</i>
<i>Figure III.4 Energie consommée dans le réseau.....</i>	<i>35</i>
<i>Figure III.5 Nombre des nœuds vivants/temps.....</i>	<i>36</i>
<i>Figure III.6 Nombre des nœuds vivants/rounds.....</i>	<i>37</i>

# Liste du Tableau

*Tableau III.1 Les paramètres des simulations.....33*

# **LISTE DES ABREVIATIONS**

<b>RSCF</b>	<b>RESEAU DE CAPTEUR SANS FIL</b>
<b>WSN</b>	<b>Wireless Sensor Networks</b>
<b>PEGASIS</b>	<b>Power-Efficient Gathering in Sensor Information Systems</b>
<b>LEACH</b>	<b>Low Energy Adaptive Clustering Hierarchy</b>
<b>TEEN</b>	<b>Threshold Sensitive Energy Efficient Sensor Network protocol</b>
<b>SAR</b>	<b>Sequential Assignment Routing</b>
<b>ADCs</b>	<b>Analog-to-Digital Converters</b>
<b>QOS</b>	<b>Quality de service</b>
<b>CH</b>	<b>Cluster head</b>
<b>TDMA</b>	<b>Time division Multiplexed Access</b>
<b>BS</b>	<b>Station de base</b>



*Introduction  
Générale*

---

*"Le travail éloigne de nous trois grands maux : l'ennui, le vice et le  
besoin"*

*Voltaire*

# Introduction générale

Depuis ces dernières années, la technologie des réseaux sans fil croît sans cesse, parallèlement aux développements et avancées dans divers domaines liés à la micro-électronique. De plus, l'émergence des Réseaux de Capteurs Sans fil (RCSF) ou WSN (en anglais : Wireless Sensor Networks) a fait découvrir de nombreux axes de recherche : une multitude d'applications potentiellement réalisables, liées néanmoins à de nombreuses contraintes intrinsèques.

Les nouvelles thématiques offertes sont autant nombreuses que les défis à relever, pour répondre aux besoins des personnes et aux exigences de plusieurs domaines d'application (militaire, scientifique, industriel, culturel, environnemental), observation de l'évolution des espèces végétales et animales rares, surveillance et gestion des infrastructures, optimisation des traitements médicaux, etc.

Néanmoins, les potentialités et avancées concernant les différents domaines cités sont corrélés à énormément de difficultés à résoudre. En effet, les capteurs ou nœuds capteurs des réseaux en question, tendent vers de petits voire très petits dispositifs, ayant une faible capacité de stockage des informations, des possibilités de calcul limitées, et sont par-dessus tout alimentés par des batteries de capacité très limitée, et généralement non rechargeables et non-replaçables. Ainsi, les protocoles établis pour traiter le contrôle de l'accès au médium, le routage, etc. dans les réseaux de capteurs, forment un noyau interdépendant avec des contraintes à la fois significatives et inévitables, indispensables à éluder de maîtrise de la consommation d'énergie et de maximisation de la durée de vie dans ces réseaux. L'exemple en est pour certaines applications où les capteurs sont déployés dans des zones hostiles : le remplacement voire l'accès tout court aux batteries reste difficile sinon impossible. Une solution technique de choix pour économiser l'énergie consiste dans ce cas de figure à pouvoir mettre les capteurs en état de veille. Le passage du mode actif (participation à la vie du réseau) à celui de veille est dévolu aux protocoles d'ordonnancement d'activité. Les décisions de changement d'état incombent à une entité centrale, ayant une connaissance globale du réseau, ou bien aux nœuds eux-mêmes, se basant uniquement sur des informations de voisinage. Ces considérations seront traitées dans notre travail de recherche.

Par conséquent, pour qu'un réseau de capteurs ait une autonomie de longue durée et donc ait une longévité maximale (quelques mois ou quelques années), il faut que le facteur de la consommation d'énergie soit pris en compte à tous les niveaux de l'architecture réseaux (de la couche physique à la couche application), le rendant ainsi très contraignant.

Notre exemple d'application qui relève du domaine de la domotique et plus précisément la gestion du confort thermique constitue un cas d'étude d'actualité très pertinent, empruntant aux exigences des bâtiments quant aux considérations d'efficacité énergétique et de faibles consommations. D'autre part il s'accommode volontiers d'une modélisation orientée vers les paradigmes des protocoles de routage. Ceux-ci constituent avec le paramètre efficacité énergétique en point de mire, relatée précédemment, l'objet essentiel de notre étude.

A cette fin, notre ouvrage s'articulera sur trois grandes parties : un premier chapitre aborde les RCSF sans fil à travers leurs caractéristiques et leurs applications avec une mise en évidence de l'application domotique orientée confort thermique à l'étude ici. Un deuxième chapitre traite des protocoles de routage dans les RCSF ; leur classification dans un premier temps, nous amène à cerner les types de protocoles (hiérarchiques : LEACH, PEGASIS, etc.) qui peuvent satisfaire aux critères de la modélisation désirée pour ensuite être analysés par simulation. Le troisième et dernier chapitre est consacré à l'exploitation d'un code algorithmique déjà existant, implémenté sous environnement MATLAB, destiné à une étude par opposition des protocoles LEACH et PEGASIS. Notre travail d'analyse s'articule sur les choix des métriques de performance et une spécification des paramètres de simulation propres au cas domotique à l'étude. Il s'ensuit relevés de résultats, graphes, interprétations.

# Chapitre I

*Généralités sur les réseaux de  
capteurs sans fil & Applications  
domotiques*

---

*"Tout travail procure de l'abondance, mais les paroles en l'air ne  
mènent qu'à la disette"*



## I.1 Introduction

Grâce aux avancées technologiques, il devient aujourd'hui envisageable de produire en masse des systèmes d'une taille extrêmement réduite et embarquant des unités de calcul et de communication sans fil pour un coût réduit. Ayant ces caractéristiques, les nœuds capteurs sont capables de générer et d'échanger des données d'une manière autonome et complètement transparente pour les utilisateurs. Les réseaux de capteurs représentent actuellement un nouveau domaine en plein développement, émergeant des innovations des technologies de communication.

Dans ce chapitre nous donnons une description synthétique des réseaux de capteurs, leurs architectures, leurs caractéristiques et contraintes ainsi que leurs domaines d'applications. Ensuite nous nous intéressons aux applications domotiques avec une indication particulière pour l'exemple à l'étude.

## I.2 Réseaux de Capteurs Sans Fil (RCSF) ou « Wireless Sensor Network (WSN) »

### I.2.1 Définition

Les RCSF sont souvent composés d'un nombre très important de nœuds appelés capteurs. Ces nœuds sont des entités capables d'opérer en toute autonomie afin de collecter, traiter et envoyer les données relatives à leur environnement. Les capteurs communiquent par radio afin de concentrer l'information sur une station collectrice située au cœur ou en bordure du territoire, appelée nœud « Sink » [1][3].

Ces entités doivent être équipées d'une batterie qui est une source d'énergie non durable, ce qui présente la contrainte la plus gênante pour la survie d'un tel réseau.

Les réseaux de capteur ne définissent pas une architecture préexistante puisque les capteurs peuvent se trouver fixés à un endroit fixe ou mobile, formant ainsi un réseau sans infrastructure prédéfinie. L'exploitation d'un tel réseau n'est pas réservée à un domaine particulier, elle peut concerner une diversité de secteurs comme la biologie, la chimie, l'environnement, ainsi que la surveillance sismique et même la télésurveillance personnelle [1][3][4][5].

#### ➤ *Composants d'un nœud capteur*

Un nœud capteur est composé de quatre unités principales, présentées en **(Figure I.1)**:

- a) **Unité d'acquisition « SensingUnit »** : elle est généralement composée de deux sous-unités : les capteurs et les convertisseurs analogique-numérique (Analog-to-Digital Converters : ADCs). Les capteurs relèvent des mesures analogiques sur les

paramètres environnementaux. Les ADCs convertissent ces signaux analogiques en signaux numériques.

- b) **Unité de traitement « Processing Unit »** : elle est composée de deux interfaces : une avec l'unité d'acquisition et une autre avec le module de transmission. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition, et stocke les données collectées.
- c) **Module de communication « Transceiver Unit »** : Il est responsable de toutes les communications via un support de communication radio qui relie le nœud au réseau.
- d) **Batterie « Power Unit »** : Source d'alimentation des unités citées précédemment.

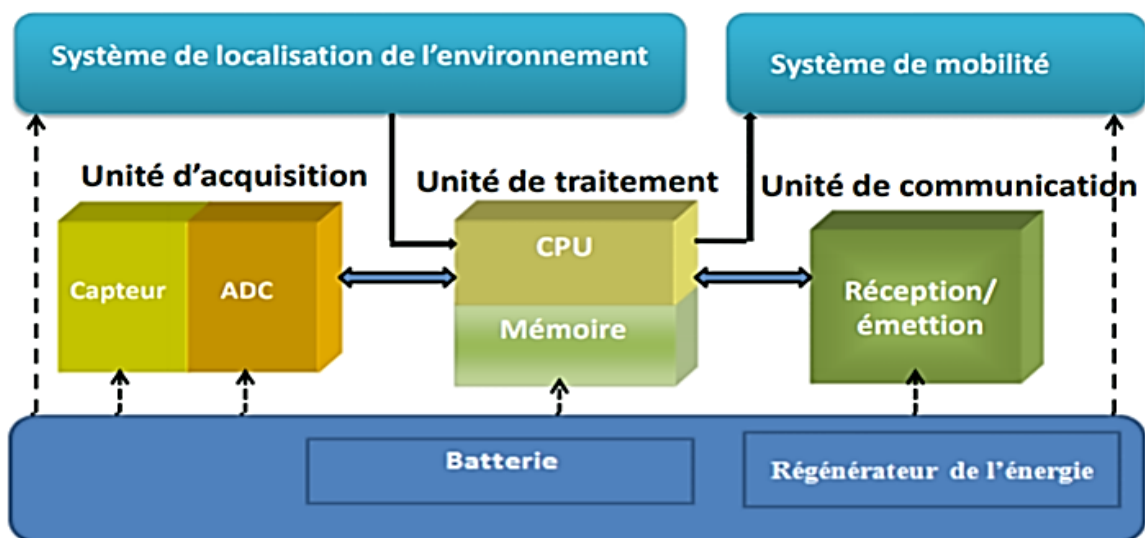


Figure I.1 : Composants d'un nœud capteur [2]

### ➤ Architecture d'un RCSF

Les nœuds capteurs sont habituellement dispersés dans un champ de capteurs « SensorField ». Ils forment ainsi un réseau sans fil multi-sauts. Chaque nœud a pour mission la collecte des données et leurs routages vers le nœud de contrôle « Sink ». À son tour, le nœud de contrôle transmet les informations reçues à travers l'Internet ou par satellite à l'utilisateur final.

(Figure I.2)

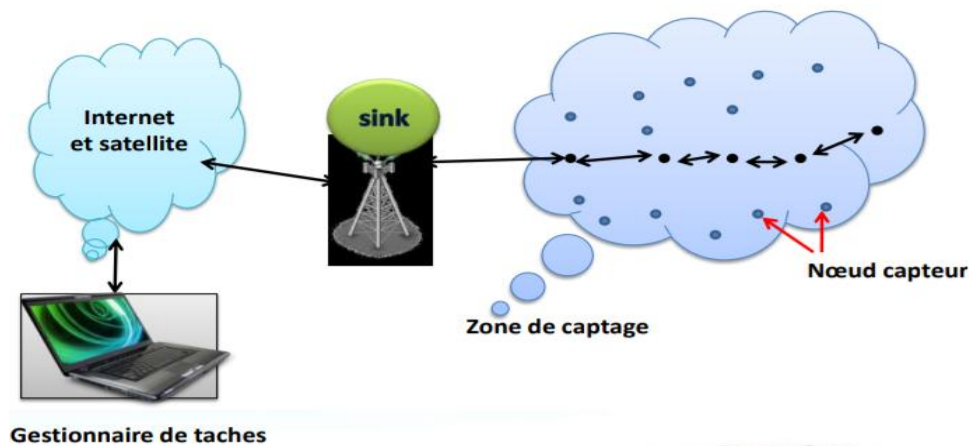


Figure I.2 : Architecture de communication dans un RCSF [2]

Il existe deux types d'architectures pour les RCSF (Figure I.3) [3]

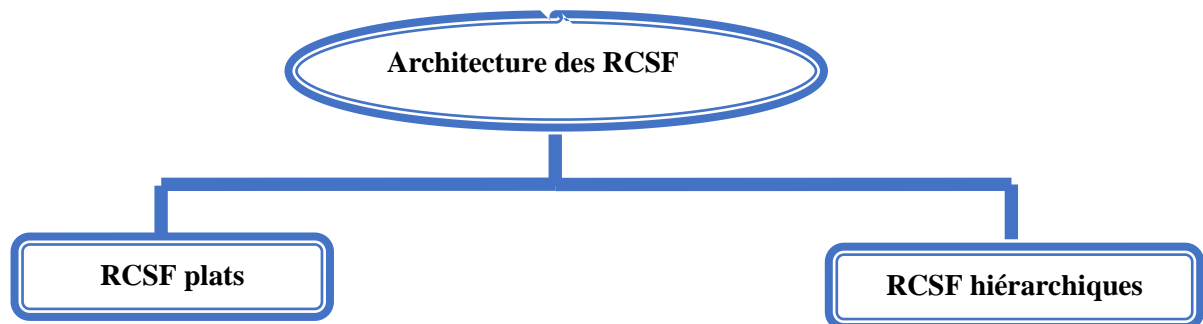


Figure I.3 : Architecture pour les réseaux de capteurs

➤ *Caractéristiques des RCSF*

Les principales caractéristiques des RCSF se résument dans ce qui suit :

- a) **Densité « importante » des nœuds** : Les RCSF se composent généralement d'un nombre très important de nœuds pour garantir une couverture totale de la zone surveillée. Ceci engendre un niveau de surveillance élevé et assure une transmission plus fiable des données sur l'état du champ de capteurs.
- b) **Topologie dynamique** : La topologie des réseaux de capteurs instable est le résultat de trois facteurs essentiels suivants :
  - ✚ **Mobilité des nœuds** : les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.

- ✚ **Défaillance des nœuds** : du fait de l'autonomie énergétique limitée des nœuds, la topologie du réseau n'est pas fixée (Les nœuds « morts » sont d'un point de vue logique, simplement supprimés).
  - ✚ **L'ajout de nouveaux nœuds** : de nouveaux nœuds peuvent facilement être rajoutés. Il suffit de placer un nouveau capteur qui soit à portée de communication d'au moins un autre nœud capteur du réseau déjà existant.
- c) **Auto organisation** : L'auto organisation s'avère très nécessaire pour ce type de réseau afin de garantir sa maintenance. Ces différentes raisons nous permettent d'avoir une topologie instable du réseau de capteur sans fil. Il devra être capable de s'auto-organiser pour continuer ses applications.
- d) **Tolérance aux fautes** : Le réseau doit être capable de maintenir ses fonctionnalités sans interruptions en cas de défaillance d'un ou plusieurs de ses capteurs. Cette défaillance peut être causée par une perte d'énergie, ou par dommage physique ou interférence de l'environnement. Le degré de tolérance dépend du degré de criticité de l'application et des données échangées [4].
- e) **Evolutivité/Extensibilité « Scalability »**: Les réseaux de capteurs peuvent contenir des centaines voire des milliers des nœuds capteurs. Un nombre aussi important engendre beaucoup de transmissions inter-nodales et nécessite que le nœud « Sink » soit équipé d'une mémoire importante pour stocker les formations reçues [4].

### I.3 Domaines d'applications des RCSF

La miniaturisation, l'évolutivité, le faible coût et la communication sans fil permettent aux réseaux de capteurs d'occuper plusieurs domaines d'applications. Ils permettent aussi d'étendre le domaine des applications existantes. Parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions : le militaire, la santé, l'environnemental, et les maisons intelligentes.

#### *I.3.1 Applications militaires*

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial Pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine [5].



Figure I.4 : Réseaux de capteurs militaires [2]

### *I.3.2 Applications à la sécurité*

*Les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires.*



Figure I.5 : Applications à la sécurité [2]

### *I.3.3 Applications environnementales :*

Des thermo-capteurs dispersés à partir d'un avion sur une forêt peuvent signaler un éventuel début d'incendie dans le champ de captage ; ce qui permettra une meilleure efficacité pour la lutte contre les feux de forêt. Dans les champs agricoles, les capteurs peuvent être semés avec les graines.



Figure I.6: Applications environnementales [2]

### **I.3.4 Applications médicales :**

On pourrait imaginer que dans le futur, la surveillance des fonctions vitales de l'être humaniserait possible grâce à des micro-capteurs qui pourraient être avalés ou implantés sous la peau. Actuellement, des micro-caméras qui peuvent être avalées existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures [6].

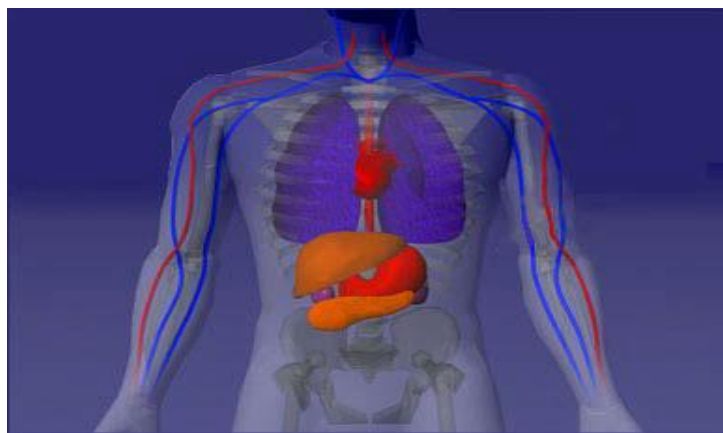


Figure I.7 Ensemble de capteurs dans un corps humain [2]

### **I.3.5 Applications domotiques :**

#### **I.3.5.1 Description**

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau

externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance.

Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique se met en état d'arrêt quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, le déclenchement d'une alarme par le capteur anti-intrusion quand un intrus veut accéder à la maison. [7]. [Voir figure I.8]



Figure I.8 : Application domotique par RSCF

### I.3.5.2 Fonctions de la domotique :

Nous pouvons dire que la domotique trouve sa place dans trois principales fonctions essentielles :

#### ❖ *La Domotique apporte du confort :*

Le fait d'automatiser sa maison a un véritable apport sur le confort qu'on y trouve. Plus besoin de se prendre une averse pour ouvrir le portail en rentrant à la maison, plus besoin de prendre froid en ouvrant les volets le matin, et fini les retours de weekend dans une maison toute froide.

#### ❖ *La Domotique permet des économies d'énergie :*

En gérant les volets selon la saison, ainsi que le chauffage, le système domotique nous permet d'économiser de l'énergie, et donc de l'argent, même si au départ on ne recherchait que le

confort en plus. La consommation d'énergie peut être suivie très finement, qu'il s'agisse de votre consommation d'électricité, d'eau, ou même de gaz.

❖ *La domotique apporte de la sécurité :*

Les automatismes que nous avons vus plus haut peuvent tout à fait contribuer à la sécurité de vos biens, en réalisant ce qu'on appelle une simulation de présence.

**I.3.5.3 Application Confort Thermique :**

Dans un bâtiment, le confort thermique c'est la température ressentie qui dépend des facteurs environnementaux tels que la température et la vitesse de l'air dans le local, le taux d'humidité et les températures des parois et de vitres. **[Voir figure I.9]**

Dans notre étude nous nous intéressons moins à l'aspect technique du déploiement d'un RCSF pour la gestion d'un environnement de confort thermique à savoir le système organisationnel en termes d'équipements et de logiciels qu'à l'aspect routage d'un RCSF – nœuds capteurs routeurs – ayant une fonction de gestion d'un confort thermique domotique en privilégiant l'efficacité énergétique dans la consommation.



**Figure I.9 : Facteurs environnementaux du confort thermique**



## **I.4 Conclusion**

Nous avons essayé à travers ce chapitre de mettre en lumière quelques aspects des RCSF à travers leurs architectures, leurs caractéristiques et leurs domaines d'applications. Parmi les applications citées il y en a une qui nous intéresse par-dessus tout c'est l'application domotique orientée confort thermique. L'aspect efficacité énergétique y est significatif et contraignant, d'où l'importance de la modélisation du routage de contrôle et donc la nécessité de définir les protocoles de routages correspondants. Vela va être étudié en chapitre II.

# Chapitre II

*Protocoles de routage : Approches de  
routage hiérarchique*

---

*"... La vie d'un homme ne dépend pas de ses biens, fût-il dans  
l'abondance"*

## II.1 Introduction

Le routage de données sur les RCSF est considéré comme le domaine de recherche de prédilection sur ce type de dispositifs. Il constitue un type particulier du routage sur les réseaux sans fil, vu les contraintes imposées par la nature des RCSF, où l'énergie est considérée comme ressource limitée et le trafic de données converge vers une destination unique qui est le puits.

Cependant, suivant le nombre de nœuds du réseau et l'étendue de champ de captage, certains nœuds ne pourront pas transmettre directement leurs messages au nœud collecteur. Ainsi, la collaboration entre les nœuds pour garantir cette transmission est exigée. De cette manière, les messages sont propagés par les nœuds intermédiaires en établissant les chemins multi-sauts entre la source lointaine et le puits. Ce processus d'acheminement des messages d'un nœud source du réseau vers un nœud destinataire s'appelle le routage. En tenant compte des capacités réduites des nœuds capteurs (calcul, énergie, mémoire), la communication avec le puits devrait se faire sans protocole de routage. Dans ce cas, la solution la plus simple serait pour chaque nœud capteur d'envoyer ses messages par diffusion jusqu'à ce qu'il arrive au collecteur.

Cette simplicité provoque néanmoins des inconvénients tels que l'implosion et le chevauchement des messages. Une implosion est détectée parce que les nœuds reçoivent des copies multiples du même message (problème de redondance des données). De plus, les nœuds ne tiennent pas compte de leurs ressources pour limiter leurs opérations (calcul, émission). Ainsi, pour qu'un réseau de capteur soit efficace, la mise en place d'un algorithme de routage devient inévitable. Néanmoins, vu les contraintes imposées par ces réseaux, la mise en place d'un protocole de routage n'est pas une tâche facile. Plusieurs travaux de recherches dans le domaine des RCSF ont été effectués récemment et ont abouti à une multitude de protocoles de routage destinés à ces réseaux. [8]

Ce chapitre est consacré aux protocoles de routage et axé plus particulièrement vers les protocoles hiérarchiques afin de caractériser notre protocole d'étude par la suite.

## II.2 Les protocoles de routage dans les RCSF - Classification

Le fait qu'un grand nombre d'applications fasse appel aux RCSF, n'empêche pas que ces applications soient confrontées à des restrictions liées à ces mêmes réseaux de capteurs qu'elles doivent contourner : faible puissance de calcul, réserve d'énergie limitée et bande passante réduite aux connections sans fil entre capteurs. [8]

Un des principaux objectifs dans la conception des RCSF est la transmission fiable de données via une heuristique de préservation d'énergie et de prévention de perte de connectivité (ex. aucun nœud isolé). Ceci est accompli par l'utilisation d'une politique stricte de gestion d'énergie. En effet, la principale source de consommation d'énergie d'un capteur est l'utilisation du réseau sans fil via son module de radiocommunication.

Les protocoles de routage au sein des RCSF sont influencés par un facteur déterminant à savoir la consommation d'énergie sans perte d'efficacité. Les capteurs utilisent leur réserve d'énergie à des fins de calcul et de transmission de données. La durée de vie d'un capteur dépend essentiellement de celle de sa batterie. Dans un RCSF, chaque nœud joue le rôle d'émetteur et de routeur. La défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier.

La plupart des protocoles de communication dans les réseaux Ad-Hoc ne s'adaptent pas aux particularités des réseaux de capteurs, d'où la nécessité de les améliorer ou de développer de nouveaux protocoles. De nombreuses stratégies de routages ont été créées pour les réseaux de capteurs sans fil. Certaines sont des adaptations de stratégies qui existaient pour d'autres types de réseaux (principalement pour les réseaux sans fil au sens le plus large) tandis que d'autres ont été conçues spécialement pour les réseaux de capteurs sans fil. [8]

La (Figure II.1) résume les principaux protocoles de routage

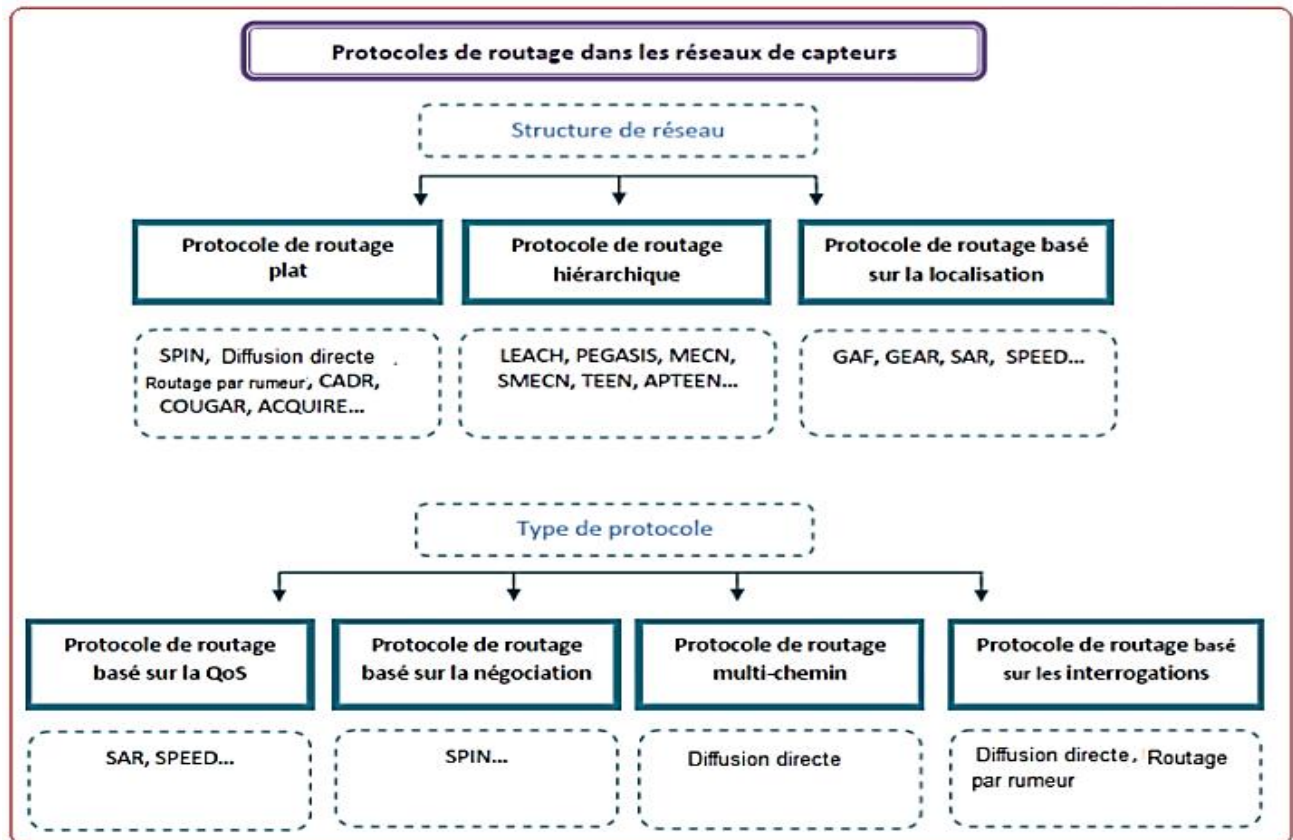


Figure II.1 : Les principaux protocoles de routage dans les RCSF [19]

### II.3 Routage hiérarchique dans les RCSF –Exemples

Le routage hiérarchique est considéré comme étant l'approche la plus favorable en termes d'efficacité énergétique. Il se base sur le concept (nœud standard – nœud maître) où les nœuds standard acheminent leurs messages à leur maître, lequel les achemine ensuite dans le réseau tout entier via d'autres nœuds maîtres jusqu'à la station de base (sink). [8][9][10][11]

Le point fort de ce type de protocoles est l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis au sink, ce qui implique une meilleure économie d'énergie. En fait, deux grandes approches sont dérivées de ce type de protocoles à savoir : approche basée chaîne « chain-based approach » [12][13][14]

et approche basée grappe « cluster-based approach » [15][16] [17][18].

Nous citons dans cette section quelques protocoles de routage hiérarchique proposés pour les réseaux de capteurs sans fil :

### ***II.3.1 SAR (Sequential Assignment Routing)***

SAR est une approche multi-chemins qui s'efforce à réaliser l'efficacité énergétique et la tolérance aux fautes. SAR crée des arbres en prenant en compte les métriques : QoS, la ressource énergétique sur chaque chemin et le niveau de priorité de chaque paquet. En utilisant ces arbres, des routes multiples du sink aux capteurs sont formés. Une ou plusieurs routes peuvent alors être empruntées [19].

### ***II.3.2 TEEN (Threshold Sensitive Energy Efficient Sensor Network protocol)***

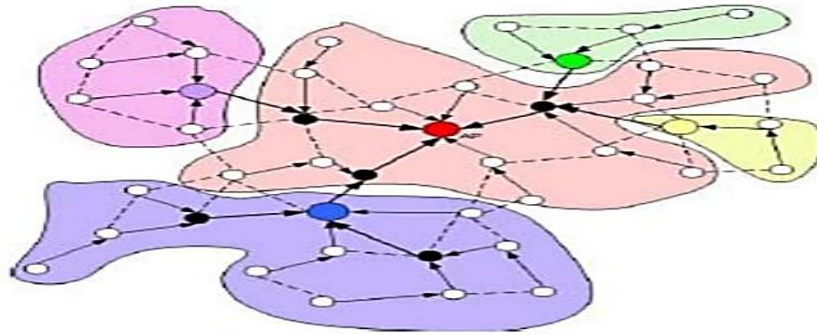
TEEN est un protocole hiérarchique conçu pour être sensible aux changements improvisés des attributs détectés tels que la température. L'architecture du réseau est basée sur un groupement hiérarchique où les nœuds les plus proches forment des grappes « clusters ». Après la construction de ceux-ci, le chef de grappe « ClusterHead » (CH) diffuse deux seuils aux nœuds [19].

### ***II.3.3 Approche à grappe « Cluster-based approach »***

Elle consiste, de façon similaire aux réseaux téléphoniques cellulaires, à partitionner le réseau en « clusters » où dans chacun d'entre eux, un seul capteur est sélectionné comme leader (CH) pour jouer le rôle spécial de point de transfert. D'ailleurs, chaque CH crée un plan de transmission pour les capteurs dans la grappe, ce qui permet aux antennes radio de chaque nœud non-CH d'être éteintes tout le temps, excepté pendant son temps de transmission. Les nœuds dont l'énergie est la plus élevée peuvent être employés dans le traitement et l'envoi d'informations tandis que ceux de basse énergie peuvent être employés dans la collection de données. L'agrégation des capteurs en grappes permet de réduire la complexité des algorithmes de routage, d'optimiser la ressource médium en la faisant gérer localement par un chef de grappe, de faciliter l'agrégation des données, de simplifier la gestion du réseau, en particulier l'affectation d'adresses, d'optimiser les dépenses d'énergie, et enfin de rendre le réseau plus évolutif « scalable ». Le groupage « clustering » permet aux nœuds d'effectuer des communications sur de petites distances avec leurs CH. [8]

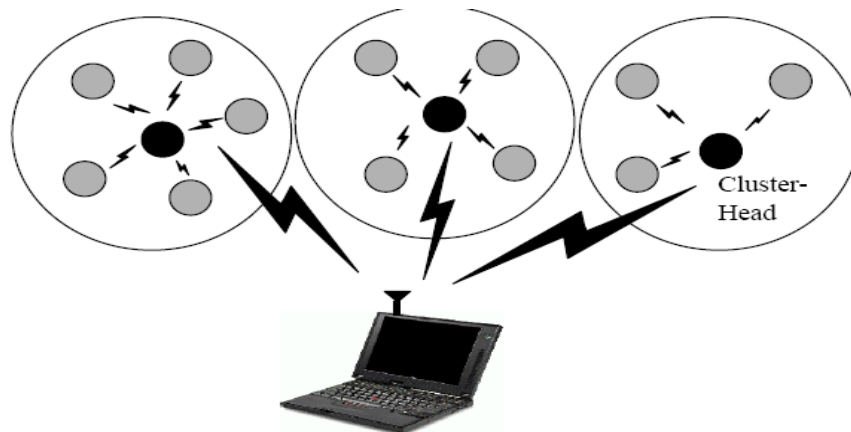
La rotation des CH s'avère également un facteur important pour l'organisation des réseaux de capteurs. Puisque la station de base (BS ou « sink ») est généralement loin du champ des

capteurs, les CH diffusent une quantité plus importante d'énergie pour la transmission de données à la BS. Par conséquent, les CH mourront rapidement si le même nœud fonctionne continuellement comme un CH. Ainsi, pour ne pas puiser la puissance de batterie d'un capteur simple, les algorithmes de groupage « clustering » étudiés jusqu'ici présentent la rotation de CH parmi les capteurs. [8](voir **Figure II.3**)



**Figure II.2: Approche à grappe "Cluster-based approach" [8]**

### **II.3.3.1 LEACH "Low Energy Adaptive Clustering Hierarchy" (Figure II.4)**



**Figure II.3 : Algorithme de routage LEACH [8]**

L'idée derrière LEACH est de former des clusters de nœuds capteurs selon la force reçue du signal et d'utiliser les têtes locales de grappe « cluster Head » comme des routeurs pour conduire des données à la station de base. Les dispositifs principaux de LEACH sont :

- **Coordination et contrôle localisés** : pour l'initialisation et le traitement de grappe
- **Rotation randomisée de la grappe** : effectuée par les stations de base ou les têtes de grappe.
- **Compression locale (agrégation)** : les nœuds CH compressent les données arrivant des nœuds appartenant à leurs grappes respectives, et envoient un paquet d'agrégation à la station de base afin de réduire la quantité d'information qui doit être transmise à la station de base.

Dans LEACH, le traitement est séparé dans des cycles de longueur constante, où chaque cycle commence par une phase d'initialisation suivie d'une phase de transmission. La durée d'un cycle est déterminée. Dans la première phase, les grappes sont organisées et les CH sont sélectionnés. Cette élection est basée sur le pourcentage désiré de CH et le nombre d'itérations au cours duquel un nœud a pris le rôle de CH. Ainsi, un nœud prend une valeur aléatoire entre 0 et 1. Si cette valeur est inférieure au seuil  $T(n)$ , le nœud se déclare CH.

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

Avec :

***P*** : pourcentage désiré de CH.

***r*** : itération actuelle.

***G*** : ensemble des nœuds qui ont été sélectionnés comme CH durant les dernières  $(1/P)$  itérations.

Chaque CH élu émet un message de signalisation au reste des nœuds dans le réseau, et qui présente les nouveaux leaders. Tous les nœuds non leaders, qui reçoivent ce message, décident de la grappe à laquelle ils veulent appartenir. Cette décision est basée sur la force du signal du message. Les nœuds non leaders informent les leaders appropriés qu'ils seront un membre de la grappe. [8]

Après la réception de tous les messages des nœuds qui voudraient être inclus dans la grappe et être basés sur le nombre de nœuds dans la grappe, le nœud leader crée un programme « TDMA



(Time division Multiplexed Access) » et assigne à chaque nœud un slot de temps quand il peut transmettre. Ce programme est émis à tous les nœuds dans la grappe.

L'algorithme LEACH utilise la technique de multiplexage temporel TDMA (**Figure II.5**) comme méthode d'accès au médium. Chaque nœud utilise la totalité de la bande passante allouée par le système de transmission durant son slot de temps. En fait, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données dans sa grappe. Il crée un modèle de communication en se basant sur TDMA, ensuite il le transmet aux nœuds de sa grappe « cluster ». Etant donné que chaque nœud connaît d'avance le slot de temps qu'il va occuper, cela permet alors au nœud de passer à l'état "endormi" durant les slots inactifs. Ainsi, la perte d'énergie due aux états de sur-écoute « overheard » et d'écoute passive « idle » est évitée. [8]

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

Dans la deuxième phase, le transfert des données actuelles à la station de base a lieu. La durée de la deuxième phase est plus longue que celle de la première phase afin de réduire au minimum les problèmes « d'overhead ». Cependant, la collection de données est centralisée et est exécutée périodiquement. Par conséquent, ce protocole s'avère le plus approprié quand on constate un besoin de surveillance constante par le réseau de capteurs.

Après un intervalle de temps donné, une rotation randomisée du rôle du CH est conduite de sorte que la dissipation uniforme d'énergie dans le réseau de capteurs soit obtenue. Les auteurs ont trouvé, en se basant sur leur modèle de simulation, que seulement 5% des nœuds ont besoin d'agir comme leaders.

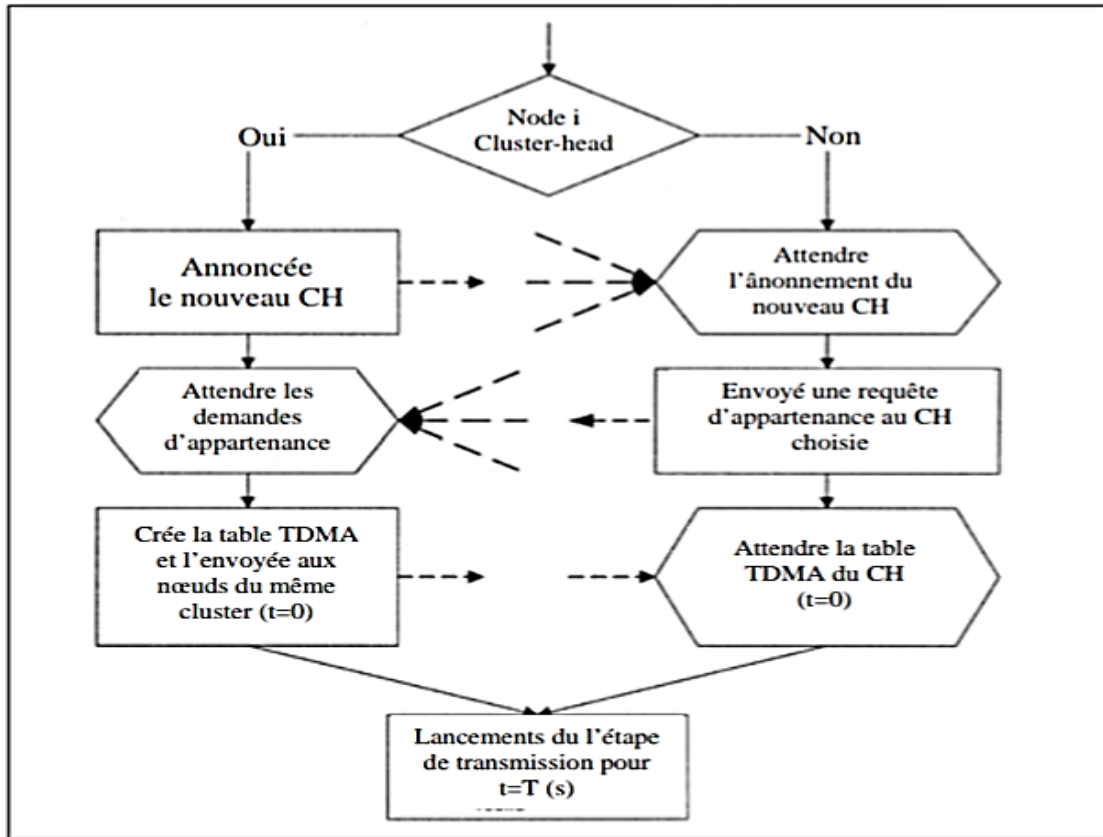


Figure II.4 : Opérations de l'étape d'initialisation [8]

### II.3.3.1.1 Avantages et inconvénients de LEACH

Bien que LEACH puisse augmenter la durée de vie du réseau en manipulant ses ressources tout en respectant plusieurs contraintes telle que la consommation d'énergie, il présente certaines limitations. Dans ce qui suit, nous citons quelques avantages et inconvénients du protocole LEACH. [27]

#### ✚ Avantages

Le protocole LEACH présente les avantages suivants :

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

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

#### Inconvénients

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

#### **II.3.4 Approche à chaîne « Chain-based approach »**

Dans cette approche, le principe de clustering est abandonné, les nœuds du réseau sont organisés de façon à former une grande chaîne de proches voisins, où un seul nœud est sélectionné pour transmettre au sink. En fait, l'idée de formation de chaîne a été proposée pour la première fois dans l'algorithme PEGASIS (Power-Efficient Gathering in Sensor Information Systems). [23, 24][25]

### *II.3.4.1 PEGASIS (Power-Efficient Gathering in Sensor Information Systems)*

C'est une amélioration du protocole LEACH qui est proposée. Le protocole « PEGASIS (Power-Efficient Gathering in Sensor Information Systems) » est un protocole basé chaîne, proche de l'optimal. L'idée de base du protocole est que, dans le but de prolonger la durée de vie du réseau, les nœuds vont être organisés de telle sorte à ce qu'ils forment une chaîne, n'auront ainsi besoin de communiquer qu'avec leurs voisins les plus proches et se relaient dans la communication avec la station de base. Quand le cycle de tous les nœuds communiquant avec la station de base aura fini, un nouveau cycle commence et ainsi de suite. Ceci réduit l'énergie exigée pour transmettre des données par cycle du moment que la dissipation d'énergie est diffusée uniformément au-dessus de tous les nœuds. Par conséquent, PEGASIS a deux principaux objectifs. D'abord, augmenter la durée de vie de chaque nœud en employant des techniques de collaboration et augmenter par conséquent la durée de vie du réseau. En second lieu, permettre seulement la coordination locale entre les nœuds voisins de sorte que la largeur de bande consommée dans la communication soit réduite. A la différence de LEACH, PEGASIS évite la formation de grappes et n'utilise qu'un seul nœud dans une chaîne afin de transmettre à la BS, au lieu d'en utiliser plusieurs.

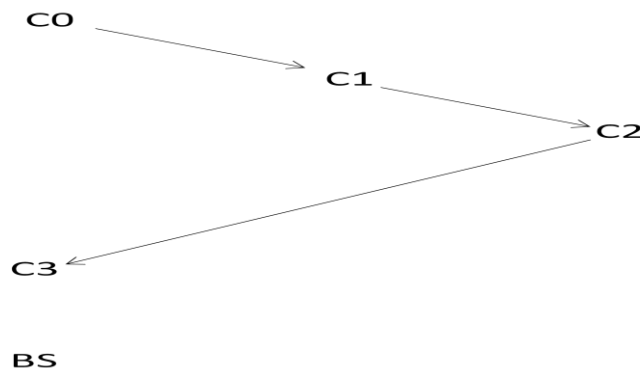
Dans PEGASIS, pour localiser le voisin le plus proche, chaque nœud utilise la force du signal pour mesurer la distance vers tous les nœuds voisins, et ajuster par la suite la force du signal de telle sorte que seul un nœud peut être entendu. La forme agrée des données sera envoyée à la station de base par n'importe quel nœud dans la chaîne et les nœuds dans cette dernière vont se prendre en relais dans la transmission à la station de base. [8]

#### *III.3.4.1.1 Approche de construction de chaîne*

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

Pour construire la chaîne, PEGASIS commence avec le nœud le plus éloigné de la BS (choisir un nœud aléatoirement s'il y a une cravate).

Le voisin le plus proche de ce nœud sera le nœud suivant dans la chaîne. Les voisins successifs sont sélectionnés de cette manière parmi les nœuds non visités (avec une cravate brisée arbitrairement) afin de former la chaîne de nœuds. L'algorithme commence par le nœud le plus lointain pour s'assurer que les nœuds les plus loin de la BS ont des voisins proches à mesure que, dans l'algorithme avide, les distances voisines augmenteront graduellement puisque des nœuds déjà présents sur la chaîne ne peuvent pas être revisités. La figure II.6 montre le nœud  $c_0$  se reliant au nœud  $c_1$ , le nœud  $c_1$  se reliant au nœud  $c_2$ , et le nœud  $c_2$  se reliant au nœud  $c_3$ , dans cet ordre. Quand un nœud meure, la chaîne est reconstruite de la même manière pour dévier le nœud Mort.



**Figure II.5 Construction de chaînes en utilisant l'algorithme gloutton**

Pour collecter des données des nœuds capteurs dans chaque cycle, chaque nœud reçoit les données d'un voisin, les fusionne avec les siennes, et transmet à un autre voisin dans la chaîne. Il faut noter que ce nœud  $i$  serait à une certaine position aléatoire  $j$  sur la chaîne.

Les Nœuds se relient dans la transmission à la BS et PEGASIS va utiliser le mod  $N$  du numéro de nœud  $i$  ( $N$  représente le nombre de nœuds) afin de transmettre à la BS dans le cycle  $i$ . Ainsi, le leader dans chaque cycle de communication sera à une position aléatoire sur la chaîne.

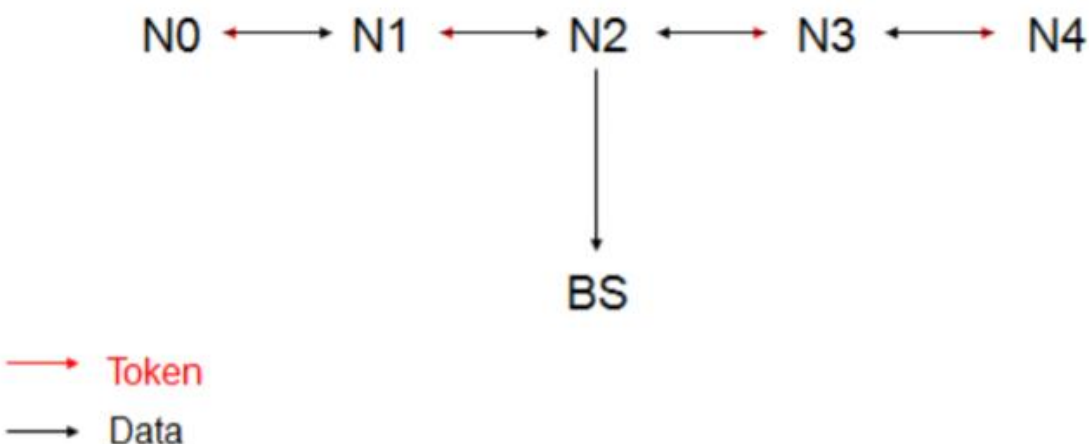
Chaque cycle de collecte de données peut être lancé par la BS avec un signal de balise qui synchronisera tous les nœuds capteurs. Puisque tous les nœuds connaissent leurs positions sur la chaîne, PEGASIS peut employer une approche de slot de temps (TDMA) pour la transmission des données. Dans le  $i$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

Le nœud  $c_0$  de fin transmettra ses données au nœud  $c_1$  dans le premier slot,  $c_1$  fusionne et transmet les données dans le deuxième slot, et ainsi de suite jusqu'à ce que le nœud leader soit atteint.

Dans les slots suivants, les transmissions de données ont lieu depuis le nœud  $c_{(n-1)}$  et se déplacent vers le nœud leader de la bonne extrémité de la chaîne. Finalement, dans le nième slot, le leader transmet les données à la BS.

Alternativement, dans un cycle donné, PEGASIS peut utiliser une approche de déplacement de jeton à contrôle simple lancée par le leader pour commencer la transmission des données des extrémités de la chaîne. Le coût est très petit du fait que la taille du jeton est très petite. Dans la figure II.7 suivante, le nœud  $c_2$  est le leader et va passer le jeton le long de la chaîne commençant au nœud  $c_0$ . Le nœud  $c_0$  passera ses données au nœud  $c_2$ .

Après que le nœud  $c_2$  ait reçu les données du nœud  $c_1$ , il passera le jeton au nœud  $c_4$ , et le nœud  $c_4$  passera ses données au nœud  $c_2$  avec fusion des données le long de la chaîne.



**Figure II.6 Approche de dépassement de jeton**

PEGASIS exécute la fusion de données à chaque nœud excepté les nœuds de fin de chaîne.

Chaque nœud va fusionner les données de ses voisins avec les siennes afin de générer un paquet simple et les transmet par la suite à son autre voisin (S'il en a deux). Dans l'exemple précédent, le nœud  $c_0$  va transmettre ses données au nœud  $c_1$ .  $c_1$  fusionne les données de  $c_2$  avec les siennes et transmet ensuite au leader. Après que  $c_2$  ait passé le jeton à  $c_4$ , le nœud  $c_4$  transmet ses données à  $c_3$ . Le nœud  $c_3$  fusionne ses données avec celles de  $c_4$  et les transmet ensuite au leader. Le nœud  $c_2$  attend de recevoir les données de ses deux voisins et fusionne alors ses données avec les leurs. Finalement, le nœud  $c_2$  transmet un message à la BS. Ainsi, dans

PEGASIS, chaque nœud va recevoir et transmettre un paquet de données dans chaque cycle et sera le leader une fois chaque  $N$  cycles. En addition, les nœuds reçoivent et transmettent des paquets de contrôle de jeton très petits. Dans la formation de la chaîne, il est possible que certains nœuds puissent relativement avoir des voisins distants le long de cette dernière. De tels nœuds vont dissiper plus d'énergie dans chaque cycle, comparé à d'autres capteurs.

L'auteur a amélioré la performance de PEGASIS en ne permettant pas à de tels nœuds de devenir leaders. L'auteur a accompli ceci en plaçant un seuil sur la distance voisine pour être leaders. Il peut être capable d'améliorer légèrement la performance de PEGASIS en appliquant un seuil adaptatif au niveau d'énergie restante dans les nœuds. Chaque fois qu'un nœud meurt, la chaîne sera reconstruite et le seuil peut être changé afin de déterminer quel nœud peut être leader.

Le protocole PEGASIS s'améliore par rapport à LEACH par l'économie énergie dans plusieurs étapes. En premier lieu, dans la collecte locale, les distances que la plupart des nœuds transmettent sont beaucoup moins par rapport à la transmission au leader dans LEACH. En second lieu, la quantité de données à recevoir par le leader est deux messages au plus au lieu de 20 (20 nœuds par grappe dans LEACH pour un réseau de 100 nœuds). Finalement, seul un nœud transmet à la BS dans chaque cycle de communication. [8]

#### *III.3.4.1.2 Avantages et inconvénients du protocole PEGASIS*

Bien que le protocole PEGASIS engendre beaucoup d'avantages en ce qu'il offre comme bonne gestion de ressources du réseau en respectant plusieurs contraintes telle que la consommation d'énergie, un nombre d'inconvénients restent plus ou moins apparents.

##### **Avantages**

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

### Inconvénients

- Bien que l' « overhead » du « clustering » soit évité, PEGASIS exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données.
- Ce protocole atteint rapidement ses limites de fonctionnement dans le cadre des réseaux fortement denses.
- Le délai de livraison des données est très important lorsque la chaîne formée est très longue. Et, le nœud qui transmet les données vers le puits peut devenir un point de congestion du réseau [26]

## III.4 Conclusion

Dans ce chapitre, nous avons axé notre étude sur le routage hiérarchique qui vise à rendre les protocoles plus favorables au passage à l'échelle tout en restant plus économiques en consommation d'énergie. Et cela, afin de couvrir une zone de captage plus large.

Ainsi, nous avons identifié le protocole LEACH qui suit une approche basée sur les groupes et le protocole PEGASIS qui est un protocole basé sur les chaînes.

Cela, nous permet de passer au dernier chapitre, la simulation des deux protocoles, l'un basé sur les grappes (LEACH) et l'autre basé sur les chaînes (PEGASIS) en essayant de comparer les résultats en termes de consommation d'énergie et de durée de vie du réseau.



# Chapitre III

*Protocole LEACH/PEGASIS : Analyses  
par Simulation*

---

*" Il y a plus de bonheur à donner qu'à recevoir "*

### **III.1. Introduction**

Après avoir cerner les contours des protocoles de routage de type hiérarchiques en chapitre II qui peuvent satisfaire à la gestion de notre application présentée en chapitre I, il convient de procéder à une caractérisation sous forme d'algorithmes des protocoles à l'étude: LEACH, basé sur une approche de grappes et PEGASIS, basé sur une approche de chaîne. S'ensuit alors une analyse par simulation, mettant en évidence l'aspect d'opposition ou dualité des deux protocoles cités, pour tirer les avantages/inconvénients par rapport aux métriques de performance choisis.

Mais avant, Nous commençons par redéfinir l'application domotique à l'étude, présentée très sommairement en chapitre I, sous le prisme du mode opératoire. Ensuite, il est fait état d'une brève description du simulateur utilisé: MATLAB

### **III.2. Application Domotique Confort Thermique - Mode opératoire**

En domotique, un petit boîtier tactile fixé au mur ou une tablette ou même un Smartphone peuvent nous mettre en communication avec les équipements ou appareillages de confort thermique : l'éclairage (ajouté au système confort thermique), chauffage/conditionnement d'air, etc.

Les informations peuvent emprunter 3 voies filaires possibles : voie filaire spécifique, câble téléphonique ou câblage électrique existant de la maison. Dans Notre cas, la communication est transmise par des ondes radio (wifi, Bluetooth, Zig Bee, etc.) et le message s'adresse au gestionnaire de la température qui est le thermostat: boîtier présent dans une pièce servant à mesurer la température et la réguler. Il est équipé d'un capteur de température mais il peut contenir aussi un capteur de présence: le thermostat enregistre notre va-et-vient et intègre ces informations dans son calendrier horaire ensuite il lui suffit d'envoyer ces données aux radiateurs pour stopper le chauffage ou le redémarrer. Ces mécanismes permettent des économies d'énergie. Concernant l'éclairage, quand nous rentrons dans la pièce, un détecteur de présence donne consignes aux lampes de s'allumer et s'il fait trop sombre, un détecteur de luminosité augmente leurs intensités et quand on sort les lumières s'éteignent toutes seules.



**Figure III.1 : Système domotique (intégrant le confort thermique)**

### **III.3. Environnement de Simulation**

Lors du développement d'un RCSF pour une application du domaine du réel, la simulation numérique du comportement et des performances du réseau tient une place de choix.

Selon Shannon, la simulation est « le processus de modélisation d'un système réel sur lequel on opère des expérimentations afin de comprendre son comportement et/ou d'évaluer plusieurs stratégies opérationnelles sur ce système ». Cela veut dire qu'une simulation est une représentation fidèle d'un système réel avec des hypothèses initiales qui permettent de réduire la complexité de la modélisation et la rendre faisable techniquement. La simulation est souvent utilisée lorsqu'aucune modélisation mathématique n'est possible vu la complexité et la grandeur du système étudié.

Plusieurs simulateurs de RCSF ou de simulateurs de réseaux adaptés aux RCSF ont été développés. Nous citons par exemple :

- **Castalia Simulator** : simulateur spécialisé uniquement pour les RCSF ; développé par leLaboratoire NICTA et basé sur la plateforme du simulateur OMNeT++. Ce dernier est un simulateur plus général de réseaux.

- **MiXiM** : simulateur basé également sur OMNeT++ et spécialisé dans la simulation desRCSF.
- **TOSSIM** : simulateur de RCSF développé pour simuler les plateformes utilisant le système d'exploitation TinyOS. Ce simulateur est performant pour modéliser le comportement des couches d'application, mais présente de faibles performances pour simuler le comportement des protocoles MAC.
- **WSNet** : simulateur de RCSF qui peut être utilisé avec un autre outil de simulation WSim pour générer des codes pour les microcontrôleurs de capteurs.
- **NS2**: simulateur de réseaux qui peut simuler les RCSF. Il supporte plusieurs protocoles de communication au niveau de toutes les couches OSI.
- **Matlab** : c'est un environnement puissant, complet et facile à utiliser destiné au calcul scientifique. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant calcul numérique et visualisation. C'est un environnement performant, ouvert et programmable qui permet de remarquables gains de productivité et de créativité.

### ***III.3.1. Présentation MATLAB***

MATLAB est un logiciel de calcul numérique produit par Math Works (logo : voir **Figure III.2**). Il est disponible sur plusieurs plateformes. C'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 prédéfinies. 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, protégées 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 rendent MATLAB, un langage de programmation et de simulation très utilisé.

Pour notre part le choix du simulateur à été grandement influencé par l'existence d'un code programme développé par **XXXX (voir annexe)**, que nous cherchons à exploiter et adapter dans le cas d'étude de notre application domotique.

### **III.3.2. Modèle radio**

Dans notre travail, nous prenons comme hypothèse le modèle radio de consommation d'énergie proposé par (*Heinzelman et al*) [20], schématisé par la **Figure III.3**, qui couvre à la fois les communications en émission et en réception, où l'onde radio se dissipe à raison de :  $E_{elec} = 50 \text{ nJ} / \text{bit}$  (en mode émission ou réception).

Lors de notre simulation des protocoles LEACH et PEGASIS qui va suivre, le modèle précédemment cité et dont les caractéristiques sont à venir, est pris en considération.

Ainsi, les énergies nécessaires pour émettre  $ETx(s, d)$  et recevoir  $ERx(s)$  des messages sont données par :

- Pour émettre un message de  $s$  bits vers un récepteur loin de  $d$  mètres, l'émetteur consomme :

$$ETx(s, d) = ETx_{elec}(s) + ETx_{amp}(s, d) \quad (\text{III.1})$$

$$ETx(k, d) = (E_{elec} * k) + (E_{amp} * s * d^2) \quad (\text{III.2})$$

- Pour recevoir un message de  $s$  bits, le récepteur consomme :

$$ERx(s) = ERx_{elec}(s) \quad (\text{III.3})$$

$$ERx(s) = E_{elec} * s \quad (\text{III.4})$$

$E_{elec}$  et  $E_{amp}$  font référence respectivement à l'énergie de transmission électronique et d'amplification

Avec  $E_{elec} = 50 * 10^{-9} J/bit$  (comme indiqué précédemment)

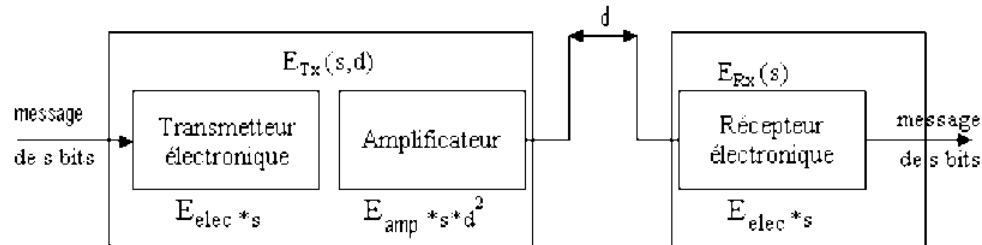


Figure III.2 : modèle de consommation d'énergie

$$ERx(k) = ERx-elec(k)$$

$$ERx(k) = E_{elec} * k$$

### III.3. Paramètres de simulation

Notre réseau de nœuds capteurs comporte les paramètres indiqués sur le **Tableau III.1**.

La particularité de notre simulation consistera à définir : les dimensions du réseau (insérer les valeurs dans les lignes de codes déjà existantes) ; la position de la BS (idem) ; le type de nœuds (là il faut trouver les mécanismes pour modifier les caractéristiques qui sont par défaut aléatoires et dynamiques) ; le nombre des nœuds (insérer les valeurs dans les lignes de codes déjà existantes) et l'énergie initiale des nœuds (idem)

#### Hypothèses et choix des valeurs

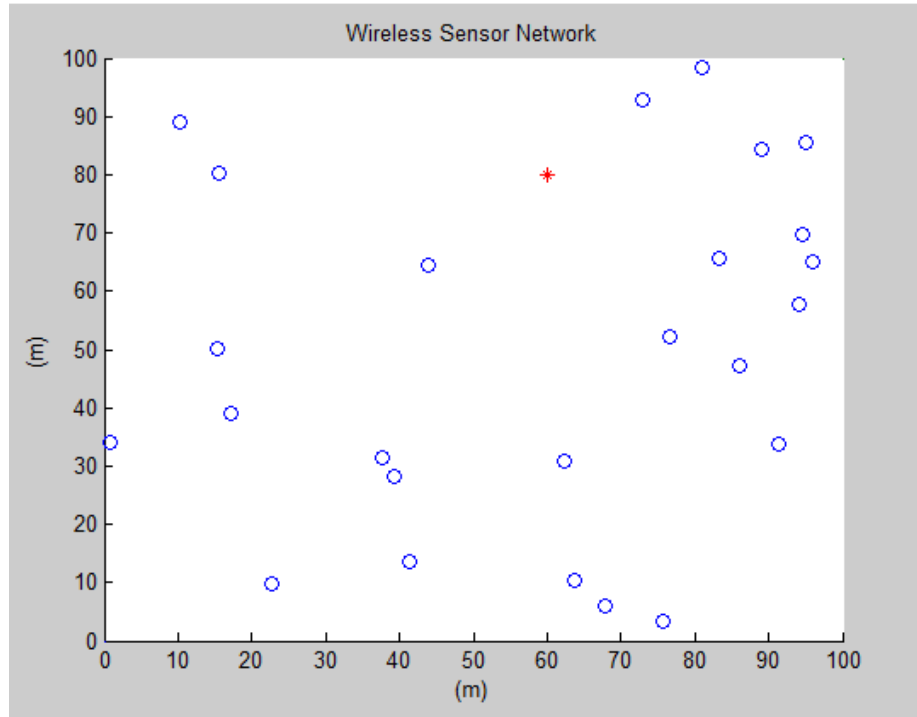
- ❖ **Surface du réseau** : valeurs clairement surdimensionnées par rapport à un réseau réel. Le choix des valeurs (100m x 100m) obéit plus à des considérations de seuils suffisants d'énergie consommée (voir **Figure III.5**)
- ❖ **Position de la BS** : choix arbitraire sans implication particulière

- ❖ **Type de nœuds** : renvoie à une topologie qui dans notre cas néglige des défaillances prématurées des nœuds et par ailleurs les nœuds et BS ne sont pas susceptibles d'être en mouvement ; considérations acceptables dans les RCSF pour applications domotiques (le déploiement est affiché sur **Figure III.4**)
- ❖ **Nombre de nœuds** : valeur compromis (un nombre plus important est préférable pour les mêmes raisons évoquées dans le choix des valeurs de la surface)
- ❖ **Energie initiale** : valeur indispensable pour démarrer notre programme mais surtout valeur compromis (idem)

Paramètres	Valeurs
La surface du réseau	100m x 100m
La position de la station de base	60 x 80
Modèle de mobilité	fixe
Le nombre de nœuds	25
L'énergie initiale de nœuds	0.5 J
Eelec (énergie de calcul)	$50 * 10^{-9}$ J/bit
ETx (énergie de transmission)	$50 * 10^{-9}$
ERx (énergie de réception)	$50 * 10^{-9}$
Eamp (énergie d'amplification)	$100 * 10^{-12}$
K (Longueur de paquet)	2000 bits

**Tableau III.1 : Les paramètres de simulation**

La figure III.4 affiche le déploiement des nœuds (environnement indiqué en **Tableau III.1**)



**Figure III.3 : Topologie de 25 nœuds fixes pour un réseau de (100 x 100) m<sup>2</sup>**

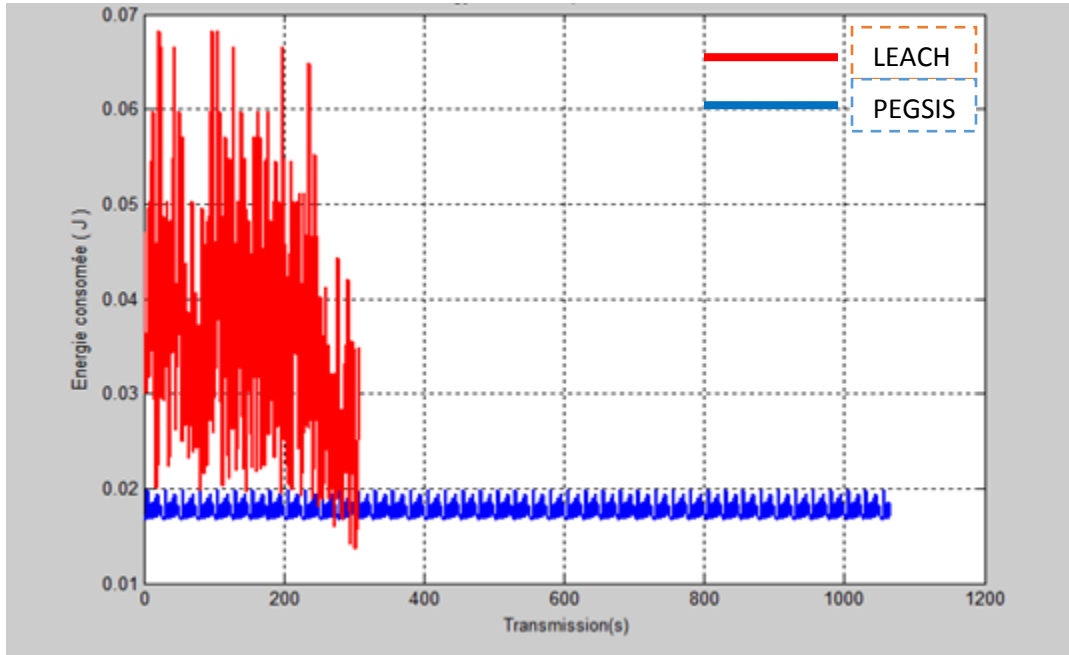
### III.4. Evaluation des performances

Les simulations qui vont suivre sont exécutées dans le but d'estimer la consommation d'énergie et la durée de vie des nœuds capteurs de notre RCSF : les deux métriques de performance choisies dans cet ouvrage. Les deux facteurs sont évidemment interdépendants. Par ailleurs les graphes sont tirés de manière à mettre en évidence l'aspect de dualité des protocoles LEACH/PEGASIS pour chaque métrique traitée .

#### III.4.1. La consommation d'énergie :

La figure III.4 montre la comparaison dans la consommation d'énergie entre les deux algorithmes pour cela nous prenons comme critère l'énergie consommée par le réseau.





**Figure III.4 : Energie consommée dans le réseau (dualité LEACH/PEGASIS)**

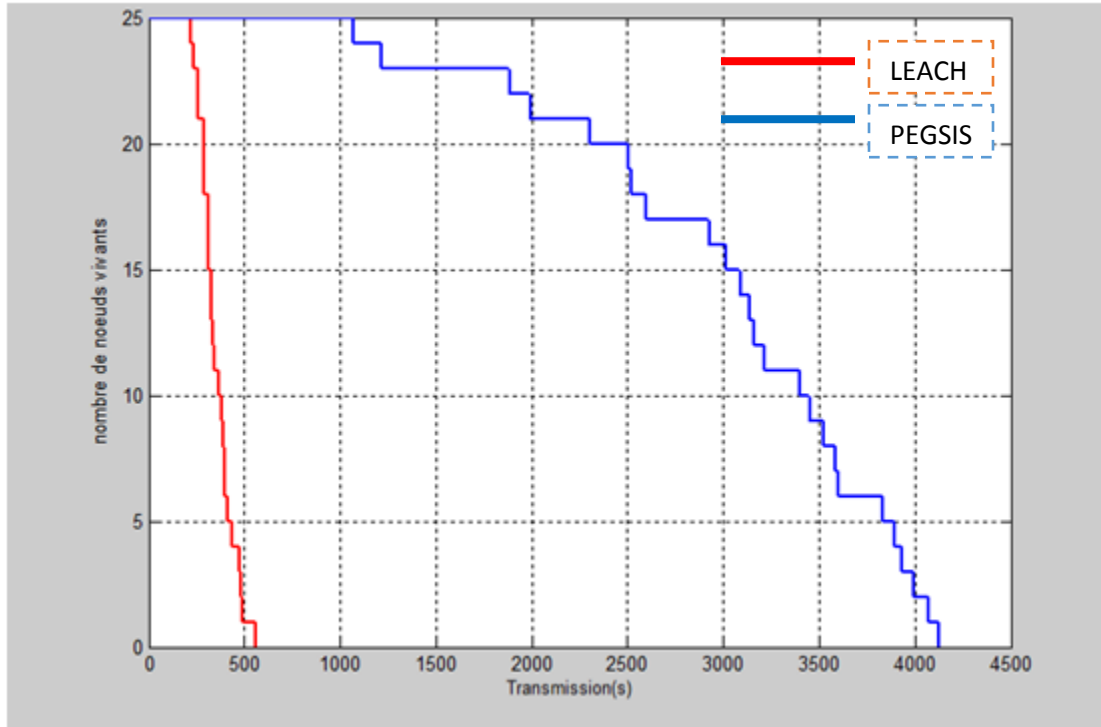
Les résultats indiquent d'une prédominance du protocole PEGASIS par rapport au protocole LEACH à consommer moins d'énergie et avec une certaine constance en termes de quantité et dans le temps (en face LEACH présente un comportement beaucoup moins stable dans ce cas de figure). Il va sans dire que la durée de simulation n'a pas atteint la durée de vie du réseau.

Comme conséquence, dans cette configuration le protocole PEGASIS présente clairement des qualités d'efficacité énergétique plus intéressantes que le protocole LEACH.

Par ailleurs, nous constatons que bien que le clustering soit évité (les nœuds sont organisés suivant une seule chaîne), PEGASIS présente un retard excessif pour les nœuds les plus éloignés sur la chaîne lors de la transmission des données.

#### **III.4.2. La durée de vie du réseau**

Concernant la durée de vie du réseau pour les deux protocoles LEACH /PEGASIS présentés en opposition sur les graphes des **Figure III.5** et **Figure III.6** où la durée de vie du réseau est comptabilisée respectivement en secondes de temps puis en nombres de rounds



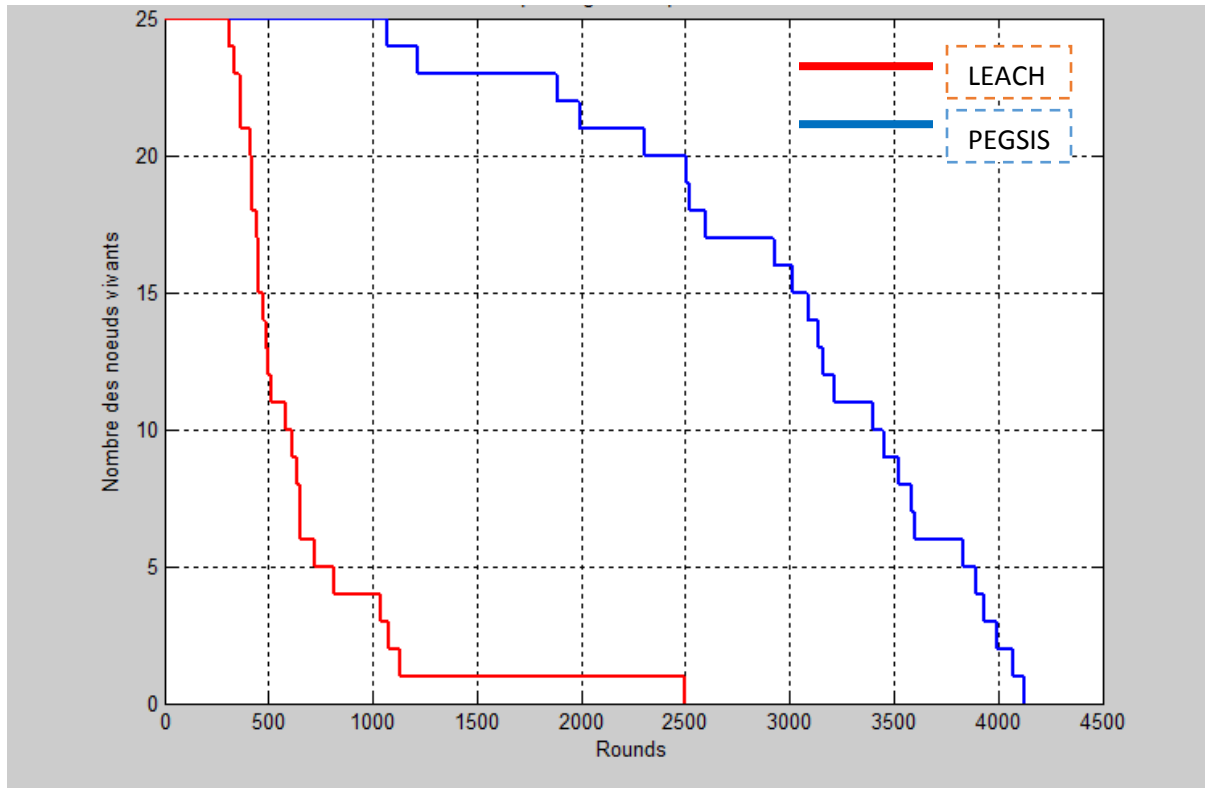
**Figure III.5 : évolution du nombre des nœuds vivants par rapport au temps (en secondes)**

Nous observons dans les graphes des **Figure III.5** et **Figure III.6** que le réseau basé sur la formation des chaînes (PEGASIS) a une durée de vie plus longue que celle de son homologue basé sur le clustering (LEACH): respectivement des ordres supérieur à sept (7) (**Figure III.5**) et supérieur à un et demi (1,5) (**Figure III.6**)

Plus précisément, nous remarquons qu'en **Figure III.5** le réseau du protocole LEACH s'affaiblit après 530 secondes de temps de simulation par contre celui du PEGASIS maintient une durée de vie de son réseau jusqu'à 4167 secondes.

Au niveau du protocole LEACH, la durée de vie du réseau est déficiente, car dans ce dernier, les nœuds s'affaiblissent plus rapidement à cause de la distance entre les CHs et leurs membres dans la grappe d'un côté et la distance entre les CHs et la station de base. Tout ça se passe dans la phase d'initialisation, une consommation énergétique supplémentaire est impliquée à cause des messages de contrôle répétitifs à chaque Round suite à la formation des clusters.

En **Figure III.6** le protocole PEGASIS est aussi « performant » en durée de vie puisqu'il arrive à atteindre les quelques 4100 rounds avant extinction des nœuds alors que son homologue du LEACH s'épuise à 2500 rounds.



**Figure III.6 : évolution du nombre des nœuds vivants par rapport nombre de rounds**

Les résultats obtenus sont conformes dans leurs interprétations avec ce qui a été avancé en chapitre II, à savoir :

- ❖ La cause de l'épuisement du protocole LEACH est bien la distance entre les nœuds des grappes et le CH, ainsi celle entre les CH et la SB.
- ❖ Le choix du CH qui se fait au niveau de la phase d'initialisation implique une consommation énergétique supplémentaire lors de la communication qui se produit à chaque cycle est aussi un facteur très important dans ce phénomène.
- ❖ Dans PEGASIS le gain dans la performance est réalisée à travers l'élimination D'over Head causée par la formation des grappes dynamique dans LEACH à travers aussi la diminution du nombre de transmission et de réception en employant l'agrégation des données.

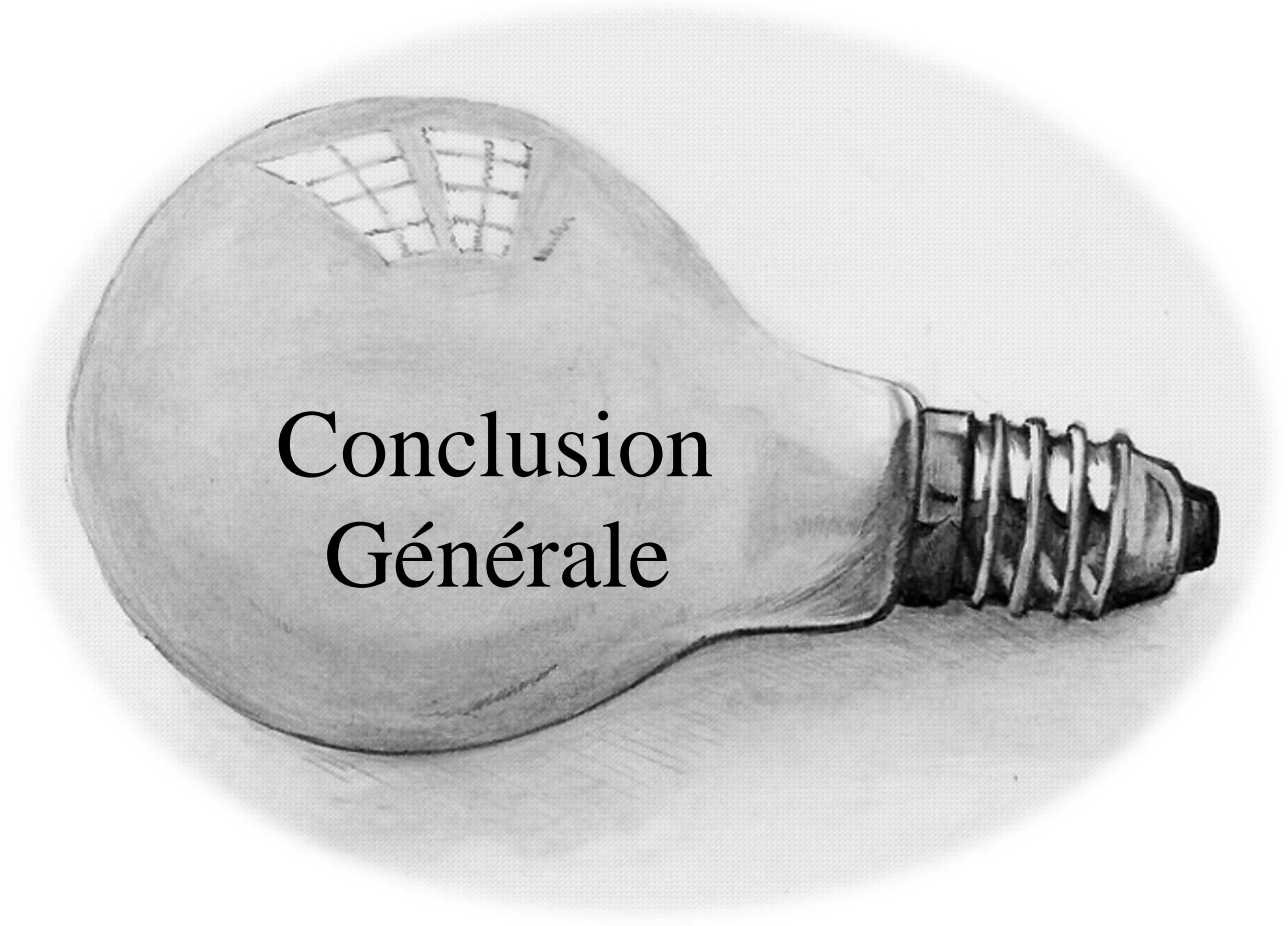
### III.5. Conclusion

Dans ce chapitre une comparaison entre deux approches de protocole de routage hiérarchiques a été effectuée.

Notre analyse est une contribution a un travail (code de simulation) établi par ALEXANDROS ZATTAS dans le cadre d'une comparaison entre deux protocoles de routage hiérarchique LEACH et PEGASIS.

Notre apport est localisé dans le choix des paramètres de déploiement des nœuds capteurs, la nature des nœuds sont homogène et fixe.

Les résultats obtenus montre que la performance de l' approche basée chaîne et meilleure que l'approche basé cluster illustrée dans les chapitres précédentes en terme de durée de vie et l'énergie consommée ; à savoir la technique de la formation des chaînes qui illumine le principe des grappes , les nœuds dans cette approche n'ont besoin de communiquer qu'avec leurs voisins les plus proches qui se joignent dans la communication avec la station de base c'est ce qui mène à la prolongation de la durée de vie d'un réseau.



## Conclusion Générale

*"Mieux vaut la fin d'une chose que son commencement"*

# Conclusion générale

Les réseaux de capteurs sans fil (RCSF) forment sans conteste une technologie émergente qui va bouleverser le quotidien de tout un chacun. Caractérisés par une grande faculté de déploiement et d'adaptation aux environnements auxquels ils sont destinés, les RCSF permettent une panoplie d'applications qui touchent et continuent à toucher tous les domaines de notre vie : à la base militaires, mais et surtout civils (surveillance de l'environnement, des structures et du trafic de tout type, applications médicales qui incluent la surveillance des patients à domicile ou à l'hôpital et puis les applications domotiques dont fait partie notre cas d'étude). Les petits voire très petits dispositifs électroniques, que sont les nœuds capteurs et qui forment l'ossature de ces réseaux, comportent néanmoins entre autres qualités intrinsèques, des dispositions limitées en termes d'efficacité énergétique et de durée de vie des nœuds. D'où la nécessité d'emploi de protocoles de routage adéquats et adaptés.

Nous avons essayé dans cet ouvrage d'apporter une contribution dans le domaine de la modélisation par algorithme et l'analyse par simulation de protocoles de routage hiérarchiques réputés pour ce qu'ils offrent comme optimisation de prolongement de temps de vie du réseau ainsi que la gestion efficace de la consommation énergétique. D'abord d'un point de vue théorique et modes opératoires les protocoles LEACH et PEGASIS issus respectivement des approches basée grappe ou « Cluster-based approach » et basée chaîne ou « Chain-based approach », sont traités et présentés. Ces considérations sont confirmées lorsque dans le prolongement, nous nous bornons à exploiter un code algorithmique déjà existant, implémenté sous MATLAB, qui rend compte de la dualité comportementale des protocoles de groupage (LEACH) et de chaînage (PEGASIS) utilisés dans les RCSF, avec des métriques adaptés à l'application domotique : gestion de confort thermique. La finalité étant de dégager une approche évolutive, plus performante, fruit de combinaison entre les deux protocoles.

Enfin, comme perspectives nous envisageons de proposer un nouvel algorithme qui combine les avantages des deux protocoles dans le but de réduire leurs inconvénients et fournir un meilleur rapport consommation d'énergie/durée de vie.

## Bibliographies

[1] Bouguer Safia « etude et simulation comparative entre les reseaux de capteurs sans fils traditionnels et les reseaux de capteurs vehiculaires » universiteaboubekrbelkaidtlemcenfaculte de technologie departementgenieelectrique et electronique.

\*C.Y. Chong and S.P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challenges", In Proceedings of the IEEE, vol.91, no.8, pp. 1247-1256, 2003.

[2] Yousef BENABASSI « Application de la redondance pour la surveillance par RCSF : cas du reseau de capteurs image sans fil » université'oranfacultédesSciences Exactes et Appliquées.

[3] ArchanaBharathidasan, Vijay ponduru, "Sensor Networks: An Overview", Department of Computer Science, University of California, 2000.

[4] Laura Raileanu et FatemiNastaran, « Les Réseaux de Senseurs », Rapport de Recherche, Haute école d'ingénierie et de gestion du Canton de Vaud, 10/01/2006.

[5] M. J. Brown, "JointBiologicalRemoteEarlyWarningSystem", Users Guide Developed for the JBREWS Project, Technical Report LA-UR-99-4676, Los Alamos National Laboratory of California University, 1999.

[6] P. Johnson and D. Andrews, "Remote continuous monitoring in the home", Journal of Telemedicine and Telecare, vol.2, no.2, pp.107-113, June 1996.

[7] E.M. Petriu, V.Z. Groza, "Sensor-based Information Appliance", IEEE Instrumentation Measurement Magazine, vol.3, no.4, pp.31-35, December 2000.

[8] Mr BoubicheDjallel Eddine Université de l'Hadj Lakhdar-Batna Faculté des sciences de l'ingénieur Département d'informatique « protocole de routage pour les reseaux de capteurs sans fil »

[9] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks", *MILCOM Proceedings on Communications for Network-Centric Operations*, 2001

[10] S. Bandyopadhyay, E. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", *Proceedings of IEEE INFOCOM*, Vol. 3, pp. 1713-1723. 2003

- [11] Y.Wang, C.Hsiao Tsai and H.Mao, 'HMRP: Hierarchy-Based Multipath Routing Protocol for Wireless Sensor Networks', *Tamkang Journal of Science and Engineering*, Vol. 9, No 3, pp. 255-264 (2006)
- [12] S. Lindsey, S. Raghavendra, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics", *IEEE Transactions on parallel and distributed systems*, Vol.13, No.9, 2002.
- [13] S. Lindsey, C. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", *IEEE Aerospace Conference Proceedings*, 2002, Vol. 3, 9-16 pp. 1125-1130.
- [14] K. Du, J. Wu and D. Zhou, "Chain-based protocols for data broadcasting and gathering In sensor networks," *International Parallel and Distributed Processing Symposium*, April 2003.
- [15] M. Younis, M. Youssef and K. Arisha, "Energy-aware Routing in Cluster-Based Sensor Networks", *the Proceedings of the 10th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems(MASCOTS2002)*, October 2002.
- [16] G. Lee, J. Kong, and O. Byeon; "Cluster based Energy Aware Routing Protocol for Sensor Networks", From Proceeding (527) *Networks and Communication Systems – 2006*.
- [17] Handy, M., Haase, M., Timmermann, "Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection," *4th IEEE International Conference on Mobile and Wireless Communications Networks*, Stockholm, 2002
- [18] S.Lee, J.Yoo, T.Choong "Distance-Based Energy Efficient Clustering for Wireless Sensor Networks", *29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, pp. 567-568, 2004.
- [19] Mr SAHRAOUI belkheyr, Etude d'un protocole de routage basé sur les colonies de Fourmis dans les réseaux de capteurs sans fil, Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique, Université Abou BakrBelkaid– Tlemcen,2012-2013.
- [20] W.R. Heinzelman, A. Chandrakasan, and H.Balakrishnan, "An Application-Specific



Protocol Architecture for Wireless Micro sensor Networks”, *IEEE Transactions on the Wireless communications*, Vol. 1, No. 4, pp. 660-670, Oct. 2002.

[21] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan “Energy-efficient communication Protocol for wireless micro sensor networks”, *IEEE Hawaii International Conference on System Sciences*, 2000.

[22] P. Agarwal and C. Procopiuc. Exact and Approximation Algorithms for Clustering. *In Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 658-667, January 1999

[23] S. Lindsey, S. Raghavendra, “Data Gathering Algorithms in Sensor Networks Using Energy Metrics”, *IEEE Transactions on parallel and distributed systems*, Vol.13, No.9, 2002.

[24] S. Lindsey, C. Raghavendra, “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”, *IEEE Aerospace Conference Proceedings*, 2002, Vol. 3, 9-16 pp. 1125-1130.

[25] K. Du, J. Wu and D. Zhou, “Chain-based protocols for data broadcasting and gathering In sensor networks,” *International Parallel and Distributed Processing Symposium*, April 2003.

[26] L. Pierron “GUI programmation en Tcl”, mars 2002.

[27] Mlle. Mehiaoui Amina « Etude comparative entre les deux protocoles de routage LEACH et PEGASIS dans les réseaux de capteurs sans fil » universite aboubekr belkaid tlemcen .

[28] Alexandros - Nikolaos Zattas, "Energy - Efficient Protocols in Wireless Sensor Networks", University of Peloponnese, 2018 Greece.

## Annexe

close all;

clear;

clc;

%%%%%%%% Network Establishment Parameters %%%%%%%%%

%%% Area of Operation %%%

% Field Dimensions in meters %

xm=100;

ym=100;

x=0; % added for better display results of the plot

y=0; % added for better display results of the plot

n=25; % Number of Nodes in the field %

dead\_nodes=0; % Number of Dead Nodes in the beggining %

% Coordinates of the Sink (location is predetermined in this simulation) %

sinkx=60;

sinky=80;

%%% Energy Values %%%

Eo=0.50; % Initial Energy of a Node (in Joules) %

% Energy required to run circuitry (both for transmitter and receiver) %

Eelec=50\*10<sup>(-9)</sup>; % units in Joules/bit

ETx=50\*10<sup>(-9)</sup>; % units in Joules/bit

ERx=50\*10<sup>(-9)</sup>; % units in Joules/bit

% Transmit Amplifier Types %

Eamp=100\*10<sup>(-12)</sup>; % units in Joules/bit/m<sup>2</sup> (amount of energy spent by the amplifier to transmit the bits)

% Data Aggregation Energy %

EDA=5\*10<sup>(-9)</sup>; % units in Joules/bit

```

% Size of data package %
k=4000; % units in bits
% Round of Operation %
rnd=0;
% Current Number of operating Nodes %
operating_nodes=n;
transmissions=0;
d(n,n)=0;
temp_dead=0;
dead_nodes=0;
selected=0;
flag1stdead=0;
count=0;
turn=0;
temp_val=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% End of Parameters %%%%%%%%%%%%%%%
%%% Creation of the Wireless Sensor Network %%%
load('matlabpeg.mat')
% Plotting the WSN %
fori=1:n
    hold on;
figure(1)
    plot(x,y,xm,ym,SN(i).x,SN(i).y,'ob',sinkx,sinky,'*r');
    title 'Wireless Sensor Network';
xlabel'(m)';
ylabel'(m)';
end

```

```

% Calculates Distance Between Each Node and the Sink (Base Station) %
for i=1:n
    SN(i).dts=sqrt((sinkx-SN(i).x)^2 + (sinky-SN(i).y)^2);
    SN(i).Esink=Eelec*k + Eamp*k*(SN(i).dts)^2;
    T(i)=SN(i).dts;
end

A=sort(T,'descend'); % Creates array A containing the distance between each node and the
sink% sorted in an ascending order
A_id(1:n)=0;
    % Creates array A_id which is sorted in a way that it's elements are
    % aligned with those of A. Contains the node ID
for i=1:n
    for j=1:n
        if A(i)==SN(j).dts
            A_id(i)=SN(j).id;
        end
    end
end

    % Creation of d Array with shortest distances %
for i=1:n
    SN(i).closest=0;
    for j=1:n
        d(j,i)=sqrt((SN(i).x-SN(j).x)^2 + (SN(i).y-SN(j).y)^2);
        if d(j,i)==0
            d(j,i)=9999;
        end
    end
end
end

```

```

fori=1:n

    [M,I]=min(d(:,i)); % finds the minimum distance of node to CH

    [Row, Col] = ind2sub(size(d),I); % displays the Cluster Number in which this node
    belongs too

    SN(i).closest=Row; % assigns node to the cluster

    SN(i).dis= d(Row,i); % assigns the distance of node to CH

end

% Choosing furthest node from sink %

fori=1:n
if SN(A_id(i)).E>0 && SN(A_id(i)).sel==0 && SN(A_id(i)).cond==1
    set= A_id(i);
    SN(set).sel=1;
    SN(set).pos=1;
    break;
end
end

order(1)=set;
    temp=1;
while selected<n
    min_dis=9999;
    fori=1:n
    if SN(i).sel==0
        d=sqrt((SN(i).x-SN(set).x)^2 + (SN(i).y-SN(set).y)^2);
        if d<min_dis
            min_dis=d;
            next=i;
        end
    end
end
end
end

```

```

        selected=selected+1;
SN(set).closest=next;
SN(set).dis=min_dis;
SN(next).sel=1;
SN(next).prev=set;
SN(next).dis2=sqrt((SN(set).x-SN(next).x)^2 + (SN(set).y-SN(next).y)^2);
        plot([SN(set).x SN(next).x], [SN(set).y SN(next).y])
        hold on;
        set=next;
        temp=temp+1;
        order(temp)=set;
end
        order(n+1)=[];
SN(set).pos=2;
SN(set).dis=0;
SN(set).closest=0;
for i=1:n
if SN(i).closest==set && SN(i).pos==0;
SN(set).prev=i;
SN(set).dis2=sqrt((SN(i).x-SN(set).x)^2 + (SN(i).y-SN(set).y)^2);
end
end
end

```

% Data Transmission %

```
while operating_nodes > 0
    energy = 0;
    for i = 1:n
        SN(i).role = 0;
    end
    % Cluster Head Election %
    cluster_head = mod(turn, n) + 1;
    if SN(cluster_head).cond == 0
        while SN(cluster_head).cond == 0
            turn = turn + 1;
        end
        cluster_head = mod(turn, n) + 1;
    end
    if SN(cluster_head).cond == 1
        SN(cluster_head).role = 1;
        SN(cluster_head).tel = SN(cluster_head).tel + 1;
        figure(1)
        plot(SN(cluster_head).x, SN(cluster_head).y, 'r')
    end
    for i = 1:n
        if order(i) == cluster_head
            cl_pos = i;
            break;
        end
    end
end
```

```

fori=1:n
ifSN(order(i)).E>0 && SN(order(i)).cond==1
ifi<cl_pos
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==1 && SN(order(i)).role==0
ETx= Eelec*k + Eamp*k*SN(order(i)).dis^2;
SN(order(i)).E=SN(order(i)).E-ETx;
    energy=energy+ETx;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
If SN(order(i)).pos==0 && SN(order(i)).role==0
ERx=(EDA+Eelec)*k;
ETx= (EDA+Eelec)*k + Eamp*k*SN(order(i)).dis^2;
SN(order(i)).E=SN(order(i)).E-ETx-ERx;
    energy=energy+ETx+ERx;
end
end
ifi>cl_pos
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==2 && SN(order(i)).role==0
ETx= Eelec*k + Eamp*k*SN(order(i)).dis^2;
SN(order(i)).E=SN(order(i)).E-ETx;
    energy=energy+ETx;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==0 && SN(order(i)).role==0
ERx=(EDA+Eelec)*k;

```



ETx= (EDA+Eelec)\*k + Eamp\*k\*SN(order(i)).dis2^2;

SN(order(i)).E=SN(order(i)).E-ETx-ERx;

energy=energy+ETx+ERx;

end

end

ifi==cl\_pos

%%%%%%%%  
%

If SN(order(i)).pos==0 && SN(order(i)).role==1

ERx=(EDA+Eelec)\*k\*2;

ETx= (EDA+Eelec)\*k + Eamp\*k\*SN(order(i)).dts^2;

SN(order(i)).E=SN(order(i)).E-ETx-ERx;

energy=energy+ETx+ERx;

end

%%%%%%%%  
%

ifSN(order(i)).pos==1 && SN(order(i)).role==1

ERx=(EDA+Eelec)\*k;

ETx= (EDA+Eelec)\*k + Eamp\*k\*SN(order(i)).dts^2;

SN(order(i)).E=SN(order(i)).E-ETx-ERx;

energy=energy+ETx+ERx;

end

%%%%%%%%  
%

%%%%%%%%  
%

ifSN(order(i)).pos==2 && SN(order(i)).role==1

ERx=(EDA+Eelec)\*k;

ETx= (EDA+Eelec)\*k + Eamp\*k\*SN(order(i)).dts^2;

SN(order(i)).E=SN(order(i)).E-ETx-ERx;

```

        energy=energy+ETx+ERx;
end
end
end
ifSN(order(i)).E<=0 && SN(order(i)).cond==1
SN(order(i)).cond=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==1 && SN(order(i)).role==0
ifoperating_nodes==1
        done='OK'
else
t=i;
while SN(order(t)).cond==0 && t<n
t=t+1;
end
SN(order(t)).pos=1;
SN(order(t)).prev=0;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==2 && SN(order(i)).role==0
ifoperating_nodes==1
        done='OK';
else
        t=i;
whileSN(order(t)).cond==0 && t>1
t=t-1;
end

```

```

SN(order(t)).pos=2;
SN(order(t)).closest=0;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
ifSN(order(i)).pos==0 && SN(order(i)).role==1
SN(order(i)).role=0;
    after=i;
for l=after:n
ifSN(order(l)).cond==1
break;
end
end
bef=i;
for j=bef:-1:1
ifSN(order(j)).cond==1
break;
end
end
SN(order(j)).closest=order(l);
SN(order(l)).prev=order(j);
SN(order(j)).dis=sqrt((SN(order(l)).x-SN(order(j)).x)^2 + (SN(order(l)).y-SN(order(j)).y)^2);
SN(order(l)).dis2=SN(order(j)).closest;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==1 && SN(order(i)).role==1
SN(order(i)).role=0;

```

```

        t=i;
while SN(order(t)).cond==0 && t<n
t=t+1;
end
SN(order(t)).pos=1;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==2 && SN(order(i)).role==1
SN(order(i)).role=0;
        t=i;
while SN(order(t)).cond==0 && t>1
t=t-1;
end
SN(order(t)).pos=2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ifSN(order(i)).pos==0 && SN(order(i)).role==0
        after=i;
for l=after:n
ifSN(order(l)).cond==1
break;
end
end
bef=i;
for j=bef:-1:1
ifSN(order(j)).cond==1
break;

```

end

end

SN(order(j)).closest=order(l);

SN(order(l)).prev=order(j);

SN(order(j)).dis=sqrt((SN(order(l)).x-SN(order(j)).x)^2 + (SN(order(l)).y-SN(order(j)).y)^2);

SN(order(l)).dis2=SN(order(j)).dis;

end

operating\_nodes=operating\_nodes-1

dead\_nodes=dead\_nodes+1;

SN(order(i)).closest=0;

SN(order(i)).prev=0;

SN(order(i)).dis=0;

SN(order(i)).dis2=0;

SN(order(i)).pos=101;

SN(order(i)).rop=rnd;

end

end

if operating\_nodes < n && temp\_val == 0

temp\_val=1;

flag1stdead=rnd;

end

rnd=rnd+1

turn=turn+1;

op(rnd)=operating\_nodes;

if energy > 0

nrg(rnd)=energy;

end

```

end
sum=0;
fori=1:flag1stdead
    sum=nrg(i) + sum;
end
temp1=sum/flag1stdead;
temp2=temp1/n;
fori=1:flag1stdead
    avg_node(i)=temp2;
end
% Plotting Simulation Results "Operating Nodes per Transmission" %
figure(2)
    plot(1:rnd,op(1:rnd),'-b','Linewidth',2);
    title ({'PEGASIS'; 'Operational Nodes per Transmission';})
xlabel'Temps';
ylabel'Operational Nodes';
    hold on;
% Plotting Simulation Results "Energy consumed per Transmission" %
figure(3)
    plot(1:flag1stdead,nrg(1:flag1stdead),'-b','Linewidth',2);
    title ({'PEGASIS'; 'Energy consumed per Transmission';})
xlabel'Temps';
ylabel'Energie consommée ( J)';
    hold on;
% Plotting Simulation Results "Operating Nodes per Round" %
figure(4)
    plot(1:rnd,op(1:rnd),'-b','Linewidth',2);
    title ({'LEACH'; 'Operating Nodes per Round';})

```

```

xlabel'Rounds';
ylabel'Operational Nodes';
hold on;

    %%%%%%%%% Network Establishment Parameters %%%%%%%%%
                %%% Area of Operation %%%

% Field Dimensions in meters %
xm=100;
ym=100;
x=0; % added for better display results of the plot
y=0; % added for better display results of the plot
n=25; % Number of Nodes in the field %
dead_nodes=0;% Number of Dead Nodes in the beggining %

% Coordinates of the Sink (location is predetermined in this simulation) %
sinkx=60;
sinky=80;

%%% Energy Values %%%
Eo=0.50; % Initial Energy of a Node (in Joules) %

% Energy required to run circuitry (both for transmitter and receiver) %
Eelec=50*10^(-9); % units in Joules/bit
ETx=50*10^(-9); % units in Joules/bit
ERx=50*10^(-9); % units in Joules/bit

% Transmit Amplifier Types %
Eamp=100*10^(-12); % units in Joules/bit/m^2 (amount of energy spent by the amplifier to
transmit the bits)

% Data Aggregation Energy %
EDA=5*10^(-9); % units in Joules/bit

% Size of data package %
k=4000; % units in bits

```

```

% Suggested percentage of cluster head %

p=0.05; % a 5 percent of the total amount of nodes used in the network is proposed to give good
results

No=p*n; % Number of Clusters %
rnd=0; % Round of Operation %

% Current Number of operating Nodes %

operating_nodes=n;
transmissions=0;
temp_val=0;
flag1stdead=0;

                %%%%%%%%% End of Parameters %%%%%%%%%

%%% Creation of the Wireless Sensor Network %%%

load('matlab.mat')

% Plotting the WSN %

fori=1:n
    hold on;
figure(1)
    plot(x,y,xm,ym,SN(i).x,SN(i).y,'ob',sinkx,sinky,'*r');
    title 'Wireless Sensor Network';
xlabel'(m)';
ylabel'(m)';
end

                %%%%%%%%% Set-Up Phase %%%%%%%%%

whileoperating_nodes>0

% Displays Current Round %

rnd

% Threshold Value %

t=(p/(1-p*(mod(rnd,1/p))));

```



```

    % Re-election Value %
tleft=mod(rnd,1/p);
% Reseting Previous Amount Of Cluster Heads In the Network %
CLheads=0;
% Reseting Previous Amount Of Energy Consumed In the Network on the Previous Round %
energy=0;
% Cluster Heads Election %
fori=1:n
    SN(i).cluster=0; % reseting cluster in which the node belongs to
    SN(i).role=0; % reseting node role
    SN(i).chid=0; % reseting cluster head id
if SN(i).rleft>0
    SN(i).rleft=SN(i).rleft-1;
end
if (SN(i).E>0) && (SN(i).rleft==0)
    generate=rand;
if generate< t
    SN(i).role=1; % assigns the node role of acluster head
    SN(i).rn=rnd; % Assigns the round that the cluster head was elected to the data
table
    SN(i).tel=SN(i).tel + 1;
    SN(i).rleft=1/p-tleft; % rounds for which the node will be unable to become a CH
    SN(i).dts=sqrt((sinkx-SN(i).x)^2 + (sinky-SN(i).y)^2); % calculates the distance
between the sink and the cluster hea
    CLheads=CLheads+1; % sum of cluster heads that have been elected
    SN(i).cluster=CLheads; % cluster of which the node got elected to be cluster head
    CL(CLheads).x=SN(i).x; % X-axis coordinates of elected cluster head
    CL(CLheads).y=SN(i).y; % Y-axis coordinates of elected cluster head

```

```

        CL(CLheads).id=i; % Assigns the node ID of the newly elected cluster head to an
array
end
end
end

% Fixing the size of "CL" array %
CL=CL(1:CLheads);

% Grouping the Nodes into Clusters &calculating the distance between node and cluster head %
fori=1:n
if (SN(i).role==0) && (SN(i).E>0) && (CLheads>0) % if node is normal
for m=1:CLheads
        d(m)=sqrt((CL(m).x-SN(i).x)^2 + (CL(m).y-SN(i).y)^2);
% we calculate the distance 'd' between the sensor node that is
% transmitting and the cluster head that is receiving with the following equation+
%  $d=\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}$  where  $x_2$  and  $y_2$  the coordinates of
% the cluster head and  $x_1$  and  $y_1$  the coordinates of the transmitting node
end
d=d(1:CLheads); % fixing the size of "d" array
[M,I]=min(d(:)); % finds the minimum distance of node to CH
[Row, Col] = ind2sub(size(d),I); % displays the Cluster Number in which this node belongs
too
SN(i).cluster=Col; % assigns node to the cluster
SN(i).dtch= d(Col); % assigns the distance of node to CH
SN(i).chid=CL(Col).id;
end
end

```

%%%%%%%% Steady-State Phase %%%%%%%%%

% Energy Dissipation for normal nodes %

fori=1:n

if (SN(i).cond==1) && (SN(i).role==0) && (CLheads>0)

if SN(i).E>0

ETx=Eelec\*k + Eamp \* k \* SN(i).dtch^2;

SN(i).E=SN(i).E - ETx;

energy=energy+ETx;

% Dissipation for cluster head during reception

if SN(SN(i).chid).E>0 && SN(SN(i).chid).cond==1 && SN(SN(i).chid).role==1

ERx=(Eelec+EDA)\*k;

energy=energy+ERx;

SN(SN(i).chid).E=SN(SN(i).chid).E - ERx;

if SN(SN(i).chid).E<=0 % if cluster heads energy depletes with reception

SN(SN(i).chid).cond=0;

SN(SN(i).chid).rop=rnd;

dead\_nodes=dead\_nodes +1;

operating\_nodes= operating\_nodes - 1

end

end

end

if SN(i).E<=0 % if nodes energy depletes with transmission

dead\_nodes=dead\_nodes +1;

operating\_nodes= operating\_nodes - 1

SN(i).cond=0;

SN(i).chid=0;

SN(i).rop=rnd;

```

end
end
end
% Energy Dissipation for cluster head nodes %
fori=1:n
if (SN(i).cond==1) && (SN(i).role==1)
if SN(i).E>0
ETx=(Eelec+EDA)*k + Eamp * k * SN(i).dts^2;
    SN(i).E=SN(i).E - ETx;
    energy=energy+ETx;
end
if SN(i).E<=0 % if cluster heads energy depletes with transmission
dead_nodes=dead_nodes +1;
operating_nodes= operating_nodes - 1
    SN(i).cond=0;
    SN(i).rop=rnd;
end
end
end
ifoperating_nodes<n &&temp_val==0
temp_val=1;
    flag1stdead=rnd
end
% Display Number of Cluster Heads of this round %
%CLheads;
transmissions=transmissions+1;
ifCLheads==0
transmissions=transmissions-1;

```

```

end
% Next Round %
rnd= rnd +1;
tr(transmissions)=operating_nodes;
op(rnd)=operating_nodes;
if energy>0
nrg(transmissions)=energy;
end
end
sum=0;
fori=1:flag1stdead
sum=nrg(i) + sum;
end
temp1=sum/flag1stdead;
temp2=temp1/n;
fori=1:flag1stdead
avg_node(i)=temp2;
end
% Plotting Simulation Results %
figure(2)
plot(1:transmissions,tr(1:transmissions),'-r','Linewidth',2);
title({'LEACH'; 'Operational Nodes per Transmission';})
xlabel'Temps(s)';
ylabel'Operational Nodes';
hold on;
% Plotting Simulation Results %
figure(3)
plot(1:flag1stdead,nrg(1:flag1stdead),'-r','Linewidth',2);

```

```

    title ({'LEACH'; 'Energy consumed per Transmission';})
xlabel'Temps (s)';
ylabel'Energie consommée ( J)';
hold on;
% Plotting Simulation Results "Operating Nodes per Round" %
figure(4)
    plot(1:rnd,op(1:rnd),'-r','Linewidth',2);
    title ({'LEACH'; 'Operating Nodes per Round';})
xlabel'Rounds';
ylabel'Operational Nodes';
holdon;

```

Dans le code open source l'auteur a utilisé des valeurs qu'on a changé et qui s'adapte pas avec notre application parmi ses valeurs on peut citer :

- Le nombre des nœuds
- La position du SINK
- L'énergie initiale

```

% Field Dimensions in meters %
xm=100;
ym=100;
x=0; % added for better display results of the plot
y=0; % added for better display results of the plot
n=25; % Number of Nodes in the field %
dead_nodes=0; % Number of Dead Nodes in the beggining %

% Coordinates of the Sink (location is predetermined in this simulation) %
sinkx=60;

```

```
% Size of data package %
```

```
k=2000; % units in bits
```

On a aussi remplacé Le nombre de bites par un autre qui nous convient

Les nœuds sont aléatoirement disperser dans le programme initiale on a ajouté une commande pour les fixée

```
%%% Creation of the Wireless Sensor Network %%%
```

```
load('matlabpeg.mat')
```

```
% Plotting the WSN %
```

```
fori=1:n
```

```
    hold on;
```

```
    figure(1)
```

```
    plot(x,y,xm,ym,SN(i).x,SN(i).y,'ob',sinkx,sinky,'*r');
```