

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



N° Réf :.....

Centre Universitaire  
Abd elhafid boussouf Mila

Institut des sciences et de la technologie

Département de Mathématiques et Informatique

**Mémoire préparé En vue de l'obtention du diplôme de  
Master  
En :Filière informatique**

**Spécialité sciences et technologies de l'information et de la communication stic**

# **Un protocole de routage pour les réseaux de capteurs sans fil**

**Préparé par** : ROUMEISSA Belkacemi

**Soutenu devant le jury**

**Président** Meriem TALAI M . A . A

**Examineur** Saliha MEZZOUDJ M . A . B

**Encadrer** Meriem BOUMASSATA M . A . B

**Année universitaire :2014/2015**

## Remerciement

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire " Ya Kayoum ".

Un grand remerciement à notre encadreur Madame « Boumassata Meriem » de nous avoir accordé sa confiance et permis de réaliser ce travail de recherche avec elle. A travers ses qualités professionnelles, en tant que directrice de ce travail, elle nous a transmis de précieuses connaissances. Merci également pour sa disponibilité, sa patience et sa bonne humeur constantes qui ont rendu ce travail très agréable et enrichissant, ainsi que pour sa rigueur scientifique.

Mes remerciements vont également aux membres de jury d'avoir accepté de juger mon travail.

Je remercie tous les personnes qui m'ont aidé durant la préparation de ce mémoire de près ou de loin.

Nous n'oublions pas nos enseignants qui tout au long du cycle d'étude au centre universitaire de Mila.

## Dédicace

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à mes chers parents qui m'ont éclairé le chemin de la vie par leur grand soutien et leurs encouragements, par leurs dévouements exemplaires et les énormes sacrifices qu'ils m'ont consentis durant mes études et qui ont toujours aimé me voir réussir. Je les remercie pour tout ce qu'ils m'ont fait.

Que dieu les garde et les protège.

A ma belle-sœur : Ghada.

A le plus bon garçon dans ma famille : Sari.

A Mon fiancé : Yakoub pour lequel je garde mes beaux souvenirs et ce qui m'a fait comprendre le vrai sens de la vie.

A tous mes oncles et mes tantes.

A toutes mes chères amies avec lesquelles j'ai partagé les meilleurs moments : Imen, Sara, Kenza, Marwa.

A tous qui sont dans ma mémoire et que mon stylo  
les oublie.

*Roumeissa*

## Résumé

Le réseau de capteurs sans-fil (RCSF) est reconnu comme un domaine de recherche très actif vu les spécificités de ce type de réseau, où la consommation d'énergie est considérée comme le défi majeur de la conception des différents protocoles pour ces réseaux.

Le routage de données est l'un de ces protocoles qui nécessitent, dans leur conception, la prise en compte des ressources énergétiques limitées des capteurs. Parmi les types de routage existants pour les RCSF, le routage hiérarchique est considéré comme un outil permettant plus de performance en ce qui concerne la consommation de l'énergie par rapport aux autres types de routage.

Dans ce mémoire nous avons essayé de proposer une nouvelle amélioration des performances de l'un des protocoles de routage hiérarchique les plus utilisées pour les RCSF, à savoir le protocole LEACH. Cette amélioration consiste en la combinaison des avantages de deux améliorations existantes de LEACH. Notre objectif principal est la minimisation de la consommation d'énergie d'un capteur et le prolongement de la durée de vie du réseau.

Les performances de notre protocole sont évaluées et comparées avec celles des autres protocoles existants à l'aide du simulateur NS2.

**Mots clés :** RCSF, Consommation énergétique, Routage hiérarchique, Cluster, LEACH, NS2.

# Table des matières

Résumé .....	II
Table de matière .....	II
Liste des figures .....	VII
Introduction général .....	01
 <b>Chapitre 01 : Généralités sur les réseaux de capteurs sans-fil</b>	
1. Introduction .....	04
2. réseaux sans-fil .....	04
2.1.Définition.....	04
2.2.Catégories des réseaux sans-fil .....	04
3. Nœud des de capteurs .....	05
3.1.Définition d'un capteur (Senseur) .....	05
3.2.Caractéristiques d'un capteur .....	06
3.3.Architecture de base d'un capteur .....	07
3.4.Modelés de nœuds capteurs commercialises .....	08
4. Réseaux de capteurs sans-fil(RCSF) .....	09
4.1.Définition .....	09
4.2.Architecture d'un RCSF .....	09
4.2.1. Les réseaux de capteurs sans-fil plats .....	10
4.2.2. Les réseaux de capteurs sans-fil hiérarchiques .....	11
4.3.Caractéristiques des RCSF .....	11
4.4.Domaines d'application d'un RCSF .....	12
4.5.La pile protocolaire dans un RCSF .....	13
4.6.Les réseaux de capteurs standards .....	14
4.7.Les systèmes d'exploitation pour les RCSF .....	16
5. Conclusion .....	17
 <b>Chapitre 02 : Le routage dans les réseaux de capteurs sans-fil</b>	
1. Introduction .....	19
2. Facteurs de conception des protocoles de routage .....	19
2.1. Tolérance aux pannes .....	19

# Table des matières

2.2.Consommation d'énergie .....	20
2.3.Limitation de capacités des nœuds .....	20
2.4.La mise à l'échelle (scalabilite) .....	20
2.5.Connectivite .....	20
2.6.Modelés de transmission de données .....	20
2.7.Hétérogénéité .....	21
3. Métriques de routage .....	21
3.1.Métriques pour la consommation énergétique .....	21
3.2.Nombre de sauts .....	22
3.3.Perte de paquets .....	22
3.4.Délai de bout-en-bout(EED) .....	22
4. Les causes de la surconsommation de l'énergie .....	23
4.1.La communication .....	23
4.2.Le traitement local de données .....	23
4.3.L'acquisition de données .....	24
4.4.La mobilité .....	24
5. Classification des protocoles de routage dans les RCSF .....	24
5.1.Classification selon la topologies du réseau .....	25
5.1.1. Topologie plat .....	26
5.1.2. Topologie hiérarchique .....	26
5.2. Classification selon la méthode d'établissement de routes .....	28
5.2.1. Protocoles proactifs .....	28
5.2.2. Protocoles réactifs .....	28
5.2.3. Protocoles hybrides .....	28
5.3. Classification selon les paradigmes de communication .....	28
5.3.1. Centre-nœuds .....	28
5.3.2. Centre-données .....	29
5.3.3. Base-localisation .....	29
5.4. Classification selon le mode de fonctionnement du protocole .....	29
5.4.1. Routage basé sur les multi-chemins .....	29
5.4.2. Routage basé sur les requêtes .....	30
5.4.3. Routage basé sur la négociation .....	30
5.4.4. Routage basé sur la qualité de service .....	30

# Table des matières

5.5. Classification selon le modelé de livraison de données .....	30
5.5.1. Time-driven .....	30
5.5.2. Query-driven .....	31
5.5.3. Event-driven .....	31
6. Routage hiérarchique dans les RCSF .....	31
6.1. Les taches d'un protocole de routage hiérarchique .....	31
6.1.1. L'algorithme de clustering utilisé .....	31
6.1.2. La réélection des ClusterHeads .....	32
6.1.3. La nature des clusters général .....	32
6.1.4. La communication intra-cluster .....	32
6.1.5. La communication inter-cluster .....	33
6.1.6. Le niveau d'agrégation de données .....	33
6.2. Protocoles hiérarchique existants.....	33
6.2.1. Le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy).....	33
6.2.2. Le protocole PEGASIS (Power-Efficient Gathering in Sensor Information Systems) .....	34
6.2.3. Le protocole HEED (Hybrid Energy-Efficient Distributed clustering).....	34
6.2.4. Le protocole LCH (Layered Clustering Hierarchy protocol).....	34
6.2.5. Le protocole KOCA (K-hop Overlapping Clustering Algorithm)...	35
6.2.6. Le protocole MCR (Multi-hop Clustering Routing protocol).....	36
6.3. Synthèse des protocoles hiérarchiques.....	36
7. Conclusion .....	39
 <b>Chapitre 03 : Amélioration des performances du protocole LEACH</b> 	
1. Introduction .....	41
2. Le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy) .....	41
2.1. Protocole MAC utilisé par LEACH .....	41
2.1.1. Accès aléatoire .....	41
2.1.2. Allocation fixe .....	42

# Table des matières

2.2. Architecture de communication de LEACH .....	42
2.3. Algorithme détaillé de LEACH .....	44
2.3.1. Phase d'initialisation .....	44
2.3.1.1. Phase d'annonce .....	44
2.3.1.2. Phase d'organisation de groupes .....	46
2.3.1.3. Phase d'ordonnancement .....	47
2.3.2. Phase de transmission .....	47
2.4. Avantages et inconvénients de LEACH .....	48
2.4.1. Avantages .....	48
2.4.2. Inconvénients .....	48
3. Amélioration existantes de LEACH .....	49
3.1. LEACH-C (LEACH CENTRALISED) .....	49
3.2. EEE-LEACH (Energy Efficient Extended LEACH) .....	50
3.3. DISTANCE BASED CLUSTER FORMATION TECHNIQUE FOR LEACH .....	50
3.4. MODLEACH (MODIFIED LEACH) .....	53
4. Amélioration proposée « DistModLeach» .....	54
4.1. Les grandes étapes du protocole « DistModLeach» .....	54
4.1.1. Phase d'initialisation.....	54
4.1.1.1. Phase d'annonce .....	54
4.1.1.2. Phase d'organisation .....	55
4.1.1.3. Phase d'ordonnancement .....	56
4.1.2. Phase de transmission .....	56
4.2. Pseudo code .....	56
5. Conclusion .....	58
<b>Chapitre 04 : Réalisation</b>	
1. Introduction .....	60
2. Environnement de simulation.....	60
2.1. Les simulateurs des RCSF .....	60
2.2. Le simulateur N2 (Network simulator 2) .....	60
2.3. L'extension Manassim .....	63

# Table des matières

2.4. Caractéristique de l'environnement de simulation utilise .....	63
3. Implémentation .....	64
3.1. Modalisation du système .....	64
3.2. Implémentation .....	64
4. Simulation et analyse des performances .....	69
4.1. Métrique d'évaluation .....	69
4.2. Résulta de simulation .....	69
4.3. Discussion .....	74
5. Conclusion .....	74
Conclusion général .....	75
Références .....	76

# Liste des figures

## Liste des figures

Figure 1. 1. Réseaux sans infrastructure ou avec infrastructure.....	4
Figure 1. 2. Exemples de capteurs .....	6
Figure 1. 3. Modèle d'un nœud capteur .....	7
Figure 1. 4. Les différentes technologies des nœuds capteurs .....	9
Figure 1. 5. Architecture d'un réseau de capteurs .....	10
Figure 1. 6. La pile protocolaire dans les réseaux de capteurs .....	13
Figure 1. 7. Principales normes des réseaux sans-fil .....	15
Figure 2. 1. Classification des protocoles de routage dans les RCSF.....	25
Figure 2. 2. Topologie plate.....	26
Figure 2. 3. Configurations pour les RCSF découpés en ensembles.....	27
Figure 3. 1. Architecture de communication de LEACH .....	43
Figure 3. 2. Opérations de l'étape d'initialisation de LEACH .....	44
Figure 3. 3. Répartition du temps et différentes phases pour chaque round .....	47
Figure 3. 4. Transmissions supplémentaires.....	51
Figure 3. 5. Exemple montrant la minimisation de la transmission supplémentaire.....	51
Figure 4. 1. Le réseau à l'instant 0.0 s .....	69
Figure 4. 2. Le réseau à l'instant 6.76 s.....	70
Figure 4. 3. Le niveau d'énergie restante dans tout le réseau en fonction du temps.....	72
Figure 4. 4. Le niveau d'énergie restante dans le nœud BS en fonction du temps.....	72
Figure 4. 5. Le niveau d'énergie restante dans le nœud N <sup>o</sup> 55 en fonction du temps...	73

## Liste des tableaux

Tableau 2.1. Tableau comparatif des solutions de routage hiérarchique multi-sauts.....	38
Tableau 4.1. les paramètres de simulation .....	64
Tableau 4.2. tableau comparatif des résultats de simulation .....	73

# Introduction générale

## Introduction Générale

Vu les avancées illustres que connaît le monde informatique, le thème des réseaux de capteurs sans-fil (RCSFs) provoque un intérêt croissant au sein des communautés scientifiques et industrielles. Ces réseaux ont su attirer un nombre croissant d'industriels, vu leur réalisme et leur apport concret. En effet, le besoin d'un suivi continu d'un environnement donné est assez courant dans diverses activités de la société [1].

Les RCSFs sont des réseaux composés d'un grand nombre de dispositifs électroniques appelés capteurs, de taille extrêmement réduite, capables de mesurer une valeur physique environnementale (température, lumière, pression, etc.) et de la communiquer via des liens sans-fil à un centre de contrôle. Ces capteurs collaborent entre eux et remontent l'information de nœud en nœud selon un mode multi-sauts jusqu'à l'aboutissement à la station de base. Où chaque nœud capteur joue à la fois le rôle d'émetteur et de routeur.

Cependant, un capteur est un équipement de taille très réduite englobant des ressources très limitées en matière de mémoire et de calcul et alimenté avec des batteries de faibles puissance. Donc, la conception d'un protocole de routage pour les RCSF est influencée par un facteur déterminant à savoir la réduction de la consommation d'énergie sans perte d'efficacité [2].

Dans ce mémoire, nous allons nous intéresser au protocole de routage LEACH proposé par Heinzelman et al [3]. Ce protocole est l'un des protocoles de routage hiérarchique les plus populaires et utilisés dans les RCSFs. Il combine l'efficacité en consommation d'énergie et la qualité de l'accès au média. Cependant, ce protocole présente quelques limitations en termes de consommation énergétique.

Notre projet consiste à étudier le fonctionnement du protocole LEACH, ainsi que les améliorations existantes portées sur ce protocole, plus précisément les deux améliorations « Distance Based Cluster Formation Technique For Leach » et « MODLEACH ». Ensuite, à proposer notre propre amélioration qui consiste en la combinaison des deux solutions précédentes dans le but de minimiser la consommation énergétique et d'augmenter la durée de vie du RCSF. Pour cela, nous avons conçu un nouveau protocole de routage hiérarchique pour les RCSF appelé « DistModLeach », basé essentiellement sur le protocole LEACH avec certaines modifications inspirées des deux améliorations précédentes de LEACH.

## Introduction générale

Pour consolider nos affirmations, nous allons utiliser le simulateur NS2 avec l'extension Mannasim. Ce simulateur permet de simuler le fonctionnement du RCSF et des différents protocoles de routage afin d'évaluer et comparer leurs performances.

Le reste du mémoire sera organisé en quatre chapitres comme suit :

Dans le premier chapitre, nous présenterons quelques généralités sur les réseaux de capteurs sans-fil. Nous présenterons les caractéristiques d'un nœud capteur, la classification des RCSF par rapport aux réseaux sans-fil, leur architecture, leurs domaines d'application, ainsi que les systèmes d'exploitation qu'ils utilisent.

Dans le deuxième chapitre, nous aborderons l'aspect routage dans les réseaux de capteurs sans-fil. Nous présenterons les facteurs de conception des protocoles de routage, les métriques de routage utilisées, et les causes de la surconsommation d'énergie dans ces réseaux. Nous présentons, aussi, une classification des protocoles de routage utilisés dans les RCSFs tout en concentrons sur le routage hiérarchique.

Dans le troisième chapitre, nous présenterons notre proposition. Nous commencerons par une explication de l'architecture de communication et de l'algorithme détaillé de LEACH. Ensuite, nous présenterons un aperçu sur les différentes améliorations existantes de LEACH. Et enfin, nous présenterons la conception de notre protocole proposé « DistModLeach » et le pseudo-code qui décrit son fonctionnement.

Dans le quatrième chapitre, nous présenterons la réalisation de notre projet. Nous commencerons par la présentation de l'environnement de simulation qui est composée essentiellement du simulateur NS2 et l'outil Mannasim. Nous expliquerons, ensuite, les étapes d'implémentation des différents protocoles de routage. Et enfin, nous présenterons les résultats de simulation avec une discussion de ces résultats.

**CHAPITRE 01 :**  
**GÉNÉRALITÉS SUR LES**  
**RÉSEAUX DE CAPTEURS**  
**SANS-FIL**

## 1. INTRODUCTION

Les réseaux de capteurs sans fil (RCSF) ou Wireless Sensor Networks (WSN), en anglais, sont devenus de plus en plus omniprésents. Les milieux scientifiques et industriels leurs prêtent de plus en plus d'attention du fait de leurs riches applications dans les domaines : médical, commercial, militaire, etc.

Les RCSF sont des réseaux de nœuds sans fil dédiés à des applications spécifiques. Ils sont considérés comme un type particulier des réseaux Ad-hoc, dans lesquels les nœuds sont des capteurs intelligents (smart sensors). Les RCSF sont composés d'un nombre potentiellement très grand (plusieurs milliers) de capteurs.

Les capteurs sont des petits appareils dotés d'une batterie, capables de communiquer entre eux et de détecter des événements s'ils se trouvent à l'intérieur de leur rayon de perception.

Dans ce premier chapitre, nous présenterons les réseaux sans fils et leurs catégories. Ensuite, nous présenterons la notion de nœud capteur, ses caractéristiques et les modèles commercialisés. Et en fin, nous présenterons un ensemble de généralités sur les réseaux de capteurs sans-fil, leur architecture, leurs domaines d'application ainsi que les systèmes d'exploitation qu'ils utilisent.

## 2. RÉSEAUX SANS-FIL

### 2.1. DÉFINITION

Un réseau sans fil (Wireless network, en anglais) est un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire via des ondes radioélectriques ou électromagnétiques. Ces équipements sont distants d'une dizaine de mètres à quelques kilomètres.

### 2.2. CATÉGORIES DES RÉSEAUX SANS-FIL

Selon l'infrastructure du réseau, les réseaux sans fil peuvent être classifiés en réseaux sans infrastructure ou avec infrastructure comme montre la figure 1.1 [04].

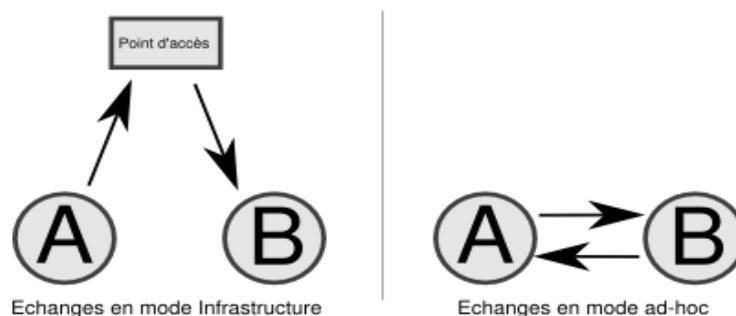


Figure 1.1 : réseaux sans infrastructure ou avec infrastructure [4].

- **Réseaux avec infrastructure**

Dans ce mode de fonctionnement le réseau est obligatoirement composé d'un point d'accès appelé station de base, munis d'une interface de communication sans fil pour la communication directe avec les sites ou unités mobiles (UM). Une station de base couvre une zone géographique limitée. Une unité mobile est rattachée à un moment donné qu'à une station de base lui offrant tous les services tant que l'UM est à l'intérieure de la zone de couverture de la SB.

- **Réseaux sans infrastructure (Ad hoc)**

Il s'agit d'un mode Point à Point, ne nécessitant pas de points d'accès. Il permet de connecter les stations quand aucun point d'accès n'est disponible. L'absence d'infrastructure oblige les UM à jouer le rôle de routeurs [6].

Un réseau Ad hoc est une collection d'hôtes équipés d'antennes pour communiquer entre eux sans aucune administration centralisée. En effet, contrairement aux réseaux filaires où des nœuds spécifiques dits « routeurs » sont responsables de l'acheminement des données, dans un réseau Ad hoc tous les nœuds sont à la fois routeurs et terminaux, Les réseaux de capteurs sans-fil sont considérés comme un type particulier des réseaux Ad hoc [7].

### **3. NŒUDS CAPTEURS**

#### **3.1. DÉFINITION D'UN CAPTEUR (SENSEUR)**

C'est un système qui sert à détecter, sous forme de signal souvent électrique, un phénomène physique.

Un capteur est un petit appareil doté de mécanismes lui permettant de relever des informations sur son environnement. La nature de ces informations varie très largement selon l'utilisation qui est faite du capteur : ce dernier peut tout aussi bien faire des relevés de température, d'humidité ou d'intensité lumineuse. Un capteur possède également le matériel nécessaire pour effectuer des communications sans-fil par ondes radio [4].

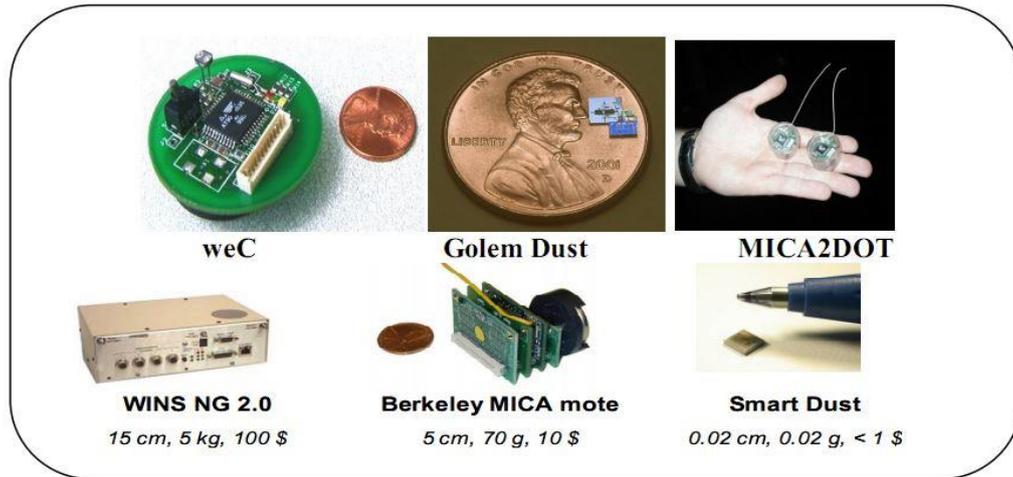


Figure 1.2 : Exemples de capteurs [8].

### 3.2. CARACTÉRISTIQUES D'UN CAPTEUR

Un capteur est doté des caractéristiques suivantes :

- ❖ Capable de calculer.
- ❖ Capable de communiquer.
- ❖ Capte toujours.
- ❖ Préposition / déploiement aléatoire.
- ❖ Limitation de la durée de vie des batteries.
- ❖ Densité (petit / grand nombre).
- ❖ Rapidité : le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information est prise en compte par la partie commande.
- ❖ Etendue de la mesure : la différence entre le plus petit signal détecté et le plus grand perceptible sans risque de destruction pour le capteur.
- ❖ Sensibilité : la plus petite variation d'une grandeur physique que peut détecter un capteur [9].

### 3.3. ARCHITECTURE DE BASE D'UN CAPTEUR

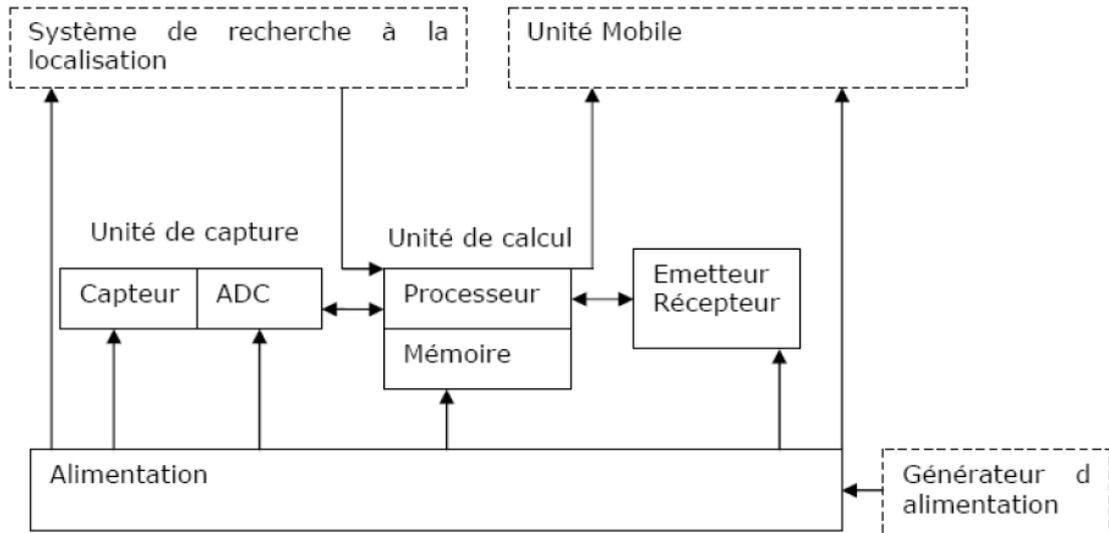


Figure 1.3 : Modèle d'un nœud capteur [10].

Un capteur est composé de quatre éléments principaux :

- Un élément qui se charge de mesurer l'environnement extérieur (unité de capture où élément capteur) : composé de deux sous éléments :
  - Le capteur récupérant des données analogiques.
  - Un convertisseur (appelée ADC pour Analog to Digital Convertor) faisant passer les données analogiques du capteur à des données numériques envoyées à une unité de calcul.
- Une unité de calcul : permet de stocker les données et exécuter les tâches de perception qui lui sont assignées. Ce composant regroupe :
  - Un processeur.
  - Une unité de mémoire réduite.
- Un élément émetteur / récepteur : élément permettant de connecter le capteur au réseau.
- Une alimentation : la source d'énergie pour le capteur, comme tout dispositif embarqué, les capteurs disposent d'une alimentation autonome telle qu'une batterie.

Trois composants additionnels peuvent être implantés dans un capteur :

- ✓ Un système de recherche d'emplacement,
- ✓ Un générateur d'alimentation,
- ✓ Une unité mobile (permettant de faire bouger le capteur).

### 3.4. MODELES DE NOEUDS CAPTEURS COMMERCIALISES

- **MICA2**

Le capteur MICA2 est un capteur de 3<sup>ème</sup> génération utilisé pour les réseaux de capteurs sans-fil et à faible consommation. Ce type de capteurs a été développé par l'université de Berkeley et est utilisé dans les applications suivantes :

- ✓ Contrôles environnementaux.
- ✓ Surveillance et sécurité.
- ✓ Réseaux de capteurs de grande capacité (+1000 nœuds).

- **TelosB**

La plate-forme TelosB a été élaborée et publiée à la communauté scientifique par l'université de Berkeley. Cette plate-forme offre une faible consommation d'énergie permettant une longue autonomie de la batterie ainsi qu'un éveil rapide de l'état de veille. Ce type de nœud peut être utilisé dans les applications suivantes :

- ✓ Plate-forme à faible puissance pour le développement de la recherche.
- ✓ Expérimentation des réseaux de capteurs sans-fil.

- **Imote2**

Le Imote2 est une plate-forme avancée de nœuds capteurs sans-fil .Il est construit autour de la faible puissance et intègre également une radio compatible 802.15.4, utilisé dans les applications suivantes :

- ✓ Traitement des images numériques.
- ✓ Contrôle et analyse industriels.
- ✓ Surveillance des séismes et des vibrations.

- **TinyNode**

Le TinyNode est un nœud capteur développé par Shocksh SA. Il est optimisé de telle sorte à supporter TinyOS. Ce type de nœud, on le retrouve dans les applications suivantes :

- ✓ Surveillance de l'environnement.
- ✓ Agriculture de précision.
- ✓ Gestion de stationnement.

	Modèle	Micro-Contrôleur	Type Radio	Radio (kbps)	RAM	Flash	EEPROM	Batterie
Crossbow Technology	MICA2	Atmega128L MPR400 (8-bit)	ChipCon CC1000	38	4 KB	128 KB	4 KB	2xAA
	MICAz	Atmega128L MPR2400 (8-bit)	ChipCon CC2420	250	4 KB	128 KB	4 KB	2xAA
	TelosB	TI MSP 430 (16-bit)	ChipCon CC2420	250	10 KB	48 KB	16 KB	2xAA
	Imote2	Intel PXA271 (32-bit)	ChipCon CC2420	250	32 MB	32 MB		3xAAA
Shockfish S A	TinyNode	TI MSP 430 (16-bit)	Semtech XE1205	153	10 KB	48 KB	16 KB	2/3xAA

Figure 1.4 : Les différentes technologies des nœuds capteurs [11].

## 4. RÉSEAUX DE CAPTEURS SANS-FIL(RCSF)

### 4.1. DÉFINITION

Les réseaux de capteurs sans-fil WSN (Wireless Sensor Networks) sont considérés comme un type spécial des réseaux ad hoc où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs. Les nœuds capteurs sont des capteurs intelligents "smart sensors", capables d'accomplir trois tâches complémentaires : le relevé d'une grandeur physique, le traitement éventuel de cette information et la communication avec d'autres capteurs. L'ensemble de ces capteurs, déployés pour une application, forme un réseau de capteurs. Le but de celui-ci est de surveiller une zone géographique, et parfois d'agir sur celle-ci (il s'agit alors de réseaux de capteurs-actionneurs).

### 4.2. ARCHITECTURE D'UN RCSF

L'architecture d'un réseau de capteurs est montrée dans la figure 1.5. L'utilisateur accède à distance aux données capturées à travers un nœud appelé le nœud directeur de tâche "Task Manager Node". Le nœud directeur de tâche est relié à l'Internet ou au satellite à travers un nœud destinataire "puits" (sink en anglais). Ce dernier agit en tant que passerelle pour le réseau de capteurs, c'est à dire qu'il relie des réseaux de capteurs à d'autres réseaux. Ce nœud est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requises aux capteurs via des messages de requêtes. Il a également d'autres capacités de traitement de l'information pour une transformation ultérieure s'il y a lieu. Les nœuds capteurs sont habituellement dispersés dans une zone de capture appelée champ de captage. Les nœuds capteurs rassemblent les données et les conduisent au

destinataire. De cette manière, les utilisateurs peuvent rechercher l'information dans les nœuds destinataires pour surveiller et commander l'environnement à distance. Notons qu'un réseau de capteurs peut contenir plusieurs nœuds puits diffusant des intérêts (ce sont la description des données requises par le nœud destinataire en utilisant une appellation combinée attribut-valeur) différents. Par exemple, un nœud puits peut demander à tous les capteurs se trouvant dans la région nord du champ de captage d'envoyer un rapport de température chaque 1 minute, pendant qu'un autre peut être intéressé seulement par les hautes températures ( $> 40^{\circ}\text{C}$ ) dans la région sud. Par conséquent, un capteur doit pouvoir stocker toutes les requêtes reçues, et les traiter séparément [12].

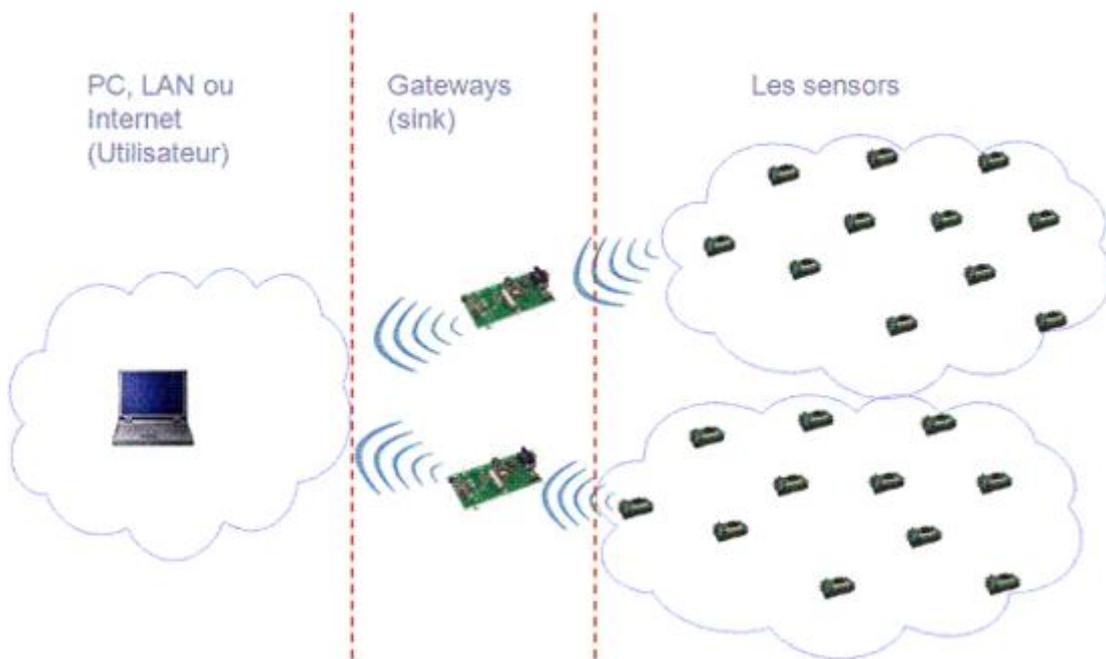


Figure 1.5 : Architecture d'un réseau de capteurs [13].

Il existe deux types d'architectures pour les réseaux de capteurs sans-fil :

#### 4.2.1. Les réseaux de capteurs sans-fil plats

Un réseau de capteurs sans-fil plat est un réseau homogène, où tous les nœuds sont identiques en termes de batterie et de complexité du matériel, excepté le nœud « Puits » qui joue le rôle d'une passerelle et qui est responsable de la transmission de l'information collectée à l'utilisateur final. Selon le service et le type de capteurs, une densité de capteurs élevée (plusieurs nœuds capteurs/m<sup>2</sup>) ainsi qu'une communication multi sauts peut être nécessaire pour l'architecture plate. En présence d'un très grand nombre de nœuds capteurs, le passage à

l'échelle devient critique. Le routage et le contrôle d'accès au médium (MAC) doivent gérer et organiser les nœuds d'une manière très efficace en termes d'énergie.

#### 4.2.2. Les réseaux de capteurs sans-fil hiérarchiques

Une architecture hiérarchique a été proposée pour réduire la complexité de la plupart des nœuds capteurs et leur déploiement, en introduisant un ensemble de nœuds capteurs plus puissants. Ceci permet de décharger la majorité des nœuds simples à faible coût de plusieurs fonctions du réseau. L'architecture hiérarchique est composée de plusieurs couches : une couche de capteurs, une couche de transmission et une couche de point d'accès. Cette architecture sans-fil est influencée par un certain nombre de facteurs et contraintes tels que la tolérance aux fautes, le redimensionnement, les coûts de production, l'environnement, la topologie du réseau, les contraintes matérielles, les médias de transmission et la consommation d'énergie [14].

#### 4.3. CARACTÉRISTIQUES DES RCSF

- ✓ **Energie limitée** : Dans un RCSF l'alimentation de chaque nœud est assurée par une source d'énergie limitée et généralement irremplaçable à cause de l'environnement hostile où il est déployé. De ce fait, la durée de vie d'un RCSF dépend fortement de la conservation d'énergie au niveau de chaque nœud.
- ✓ **Modèle de communication** : Les nœuds dans les RCSF communiquent selon un paradigme plusieurs-à-un (many to one). En effet, les nœuds capteurs collectent des informations à partir de leur environnement et les envoient toutes vers un seul nœud qui représente le centre de traitement.
- ✓ **Densité de déploiement** : Le nombre de nœuds capteurs peut atteindre des millions de nœuds pour permettre une meilleure granularité de surveillance. De plus, si plusieurs nœuds capteurs se retrouvent dans une région, un nœud défaillant pourra être remplacé par un autre. Cependant, la densité de déploiement donne naissance à des challenges pour la communication entre les nœuds. En effet, elle provoque des collisions ou des endommagements des paquets transmis.
- ✓ **Absence d'adressage fixe des nœuds** : Les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les RCSF. Ces derniers utilisent un adressage basé sur l'attribut du phénomène capté, on parle donc de l'adressage basé-attribut. En effet, les requêtes des utilisateurs ne sont pas généralement destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés par un attribut.

- ✓ **Limitations de ressources physiques** : A cause de la réduction des composants électroniques, les performances des nœuds capteurs sont limitées : la capacité de calcul, la capacité de stockage et la ressource d'énergie sont limitées. Par conséquent, les nœuds capteurs collaborent en traitant partiellement les mesures captées et envoient seulement les résultats à l'utilisateur. Une autre conséquence, ces limitations imposent des portées de transmission réduites contraignant les informations à être relayées de nœud en nœud avant d'atteindre le destinataire. C'est la raison pour laquelle les RCSF adoptent des communications multi-sauts.
- ✓ **Sécurité** : En plus des problèmes de sécurité rencontrés dans les réseaux Ad Hoc (récupérer du matériel cryptographique, supprimer un nœud du réseau, etc.), les RCSF rencontrent d'autres handicaps dus à leurs challenges, à savoir l'autonomie et la miniaturisation des capteurs. Cela engendre l'inapplicabilité des mécanismes de défense utilisés dans les réseaux Ad Hoc tout en créant d'autres mécanismes de sécurité pour les RCSF. De plus, l'absence d'une sécurité physique dans l'environnement hostile où ils sont déployés, expose les nœuds à un danger qui tend vers la falsification de l'information. En effet, les nœuds capteurs eux-mêmes sont des points de vulnérabilité du réseau car ils peuvent être modifiés, remplacés ou supprimés.

#### 4.4. DOMAINES D'APPLICATION D'UN RCSF

Parmi les domaines d'application où la technologie des réseaux de capteurs est la plus intéressante et efficace, nous citons :

- **Militaire**
  - ✓ Contrôle des forces ennemies.
  - ✓ Contrôle des équipements.
  - ✓ Détection d'attaques par des armes : nucléaires, chimiques, etc.
- **Santé**
  - ✓ Télésurveillance des signes vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées.
  - ✓ Contrôle à distance des données physiologiques.
  - ✓ Suivi et contrôle des médecins et des patients dans les hôpitaux.
- **Environnemental**
  - ✓ Microclimats.
  - ✓ Découverte de catastrophes naturelles telles que : détection d'incendie, détection d'inondation, etc.

- **Agriculture**
  - ✓ Contrôle d'irrigation.
- **Maison**
  - ✓ Automatisation.
- **Commercial**
  - ✓ Contrôle environnemental dans les usines et les bureaux.
  - ✓ Contrôle d'inventaire.
  - ✓ Suivi des véhicules et détection des encombrements.
- **Sécurité**
  - ✓ Détection d'intrusions [14].

#### 4.5. La pile protocolaire dans un RCSF

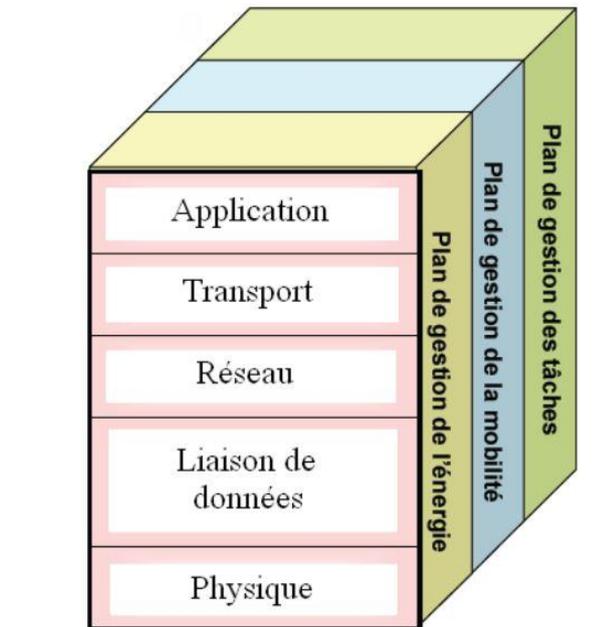


Figure 1.6 : La pile protocolaire dans les réseaux de capteurs [15].

Le rôle de cette pile consiste à standardiser la communication entre les participants afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles.

Ce modèle comprend 5 couches qui ont les mêmes fonctions que celles du modèle OSI ainsi que 3 couches pour la gestion de la puissance, la gestion de la mobilité et la gestion des tâches.

- **Couche application**

Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs.

- **Couche transport**

Elle vérifie le bon acheminement des données et la qualité de la transmission.

- **Couche réseau**

Elle s'occupe du routage de données fournies par la couche transport. Elle établit les routes entre les nœuds capteurs et le nœud puits et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc.

- **Couche liaison de données**

Elle est responsable de l'accès au media physique et la détection et la correction d'erreurs intervenues sur la couche physique. De plus, elle établit une communication saut-par-saut entre les nœuds.

- **Couche physique**

Elle permet de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences.

- **Le niveau de gestion d'énergie**

Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs. Dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un nœud voisin afin d'éviter la réception des messages dupliqués.

- **Le niveau de gestion de mobilité**

Ce niveau détecte et enregistre tous les mouvements des nœuds capteurs, de manière à leur permettre de garder continuellement une route vers l'utilisateur final, et maintenir une image récente sur les nœuds voisins.

- **Le niveau de gestion des tâches**

Le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau [4].

#### 4.6. LES RÉSEAUX DE CAPTEURS STANDARDS

Plusieurs gammes de produits sont actuellement commercialisées, mais la normalisation pourrait encore modifier les choses. Les groupes de travail qui se chargent de

cette normalisation proviennent de l'IEEE aux États Unis et de l'ETSI en Europe. La figure 1.7 décrit les principales normes des réseaux sans-fil.

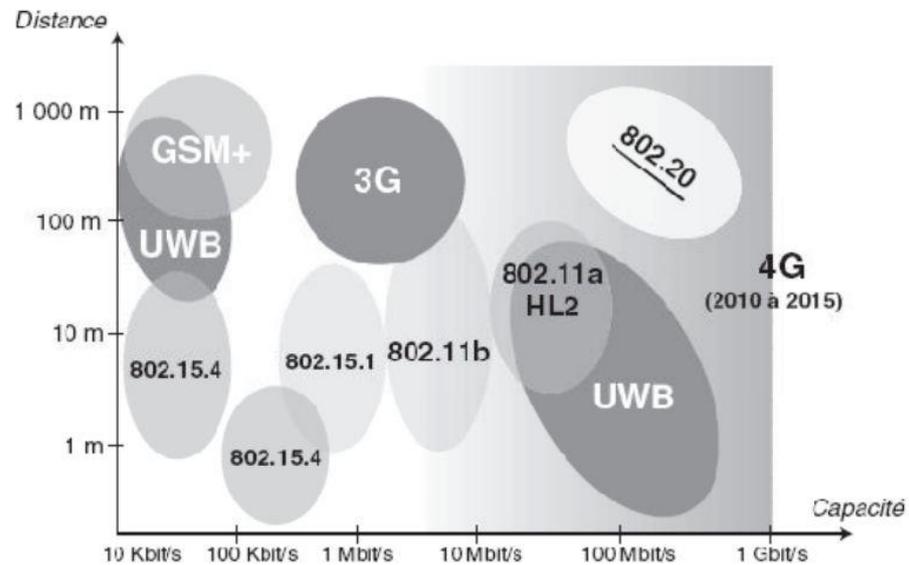


Figure 1.7 : Principales normes des réseaux sans-fil [16].

Dans ce qui suit, nous allons présenter les RCSF standards qui ont été étudiés dans la littérature.

- **IEEE 802.15.1**

Le plus connu, prend en charge la norme Bluetooth. Cette norme est rarement utilisée dans les RCSF à cause de sa consommation importante d'énergie.

- **IEEE 802.15.3**

Définit la norme UWB (UltraWide Band), qui met en œuvre une technologie très spéciale, caractérisée par l'émission à une puissance extrêmement faible, sous le bruit ambiant [17].

- **IEEE 802.15.4**

Les réseaux micro capteurs sans fil ont été l'objet de recherches intensives ces dernières années, ils émergent maintenant dans des applications industrielles. Une étape importante dans cette transition a été le dégagement de la norme d'IEEE 802.15.4 qui indique l'interopérabilité dans la couche physique et la couche MAC (Medium Access Control) visant la radio de transmission du nœud capteur [18,19]. Le standard IEEE 802.15.4 supporte différentes topologies de réseaux.

- **ZigBee Alliance**

En 2002, ZigBee Alliance a été constituée par une association d'entreprises. Le but de ZigBee Alliance est de développer des produits de contrôle fiables avec un coût réduit, et qui ne demandent pas beaucoup d'énergie. Ces produits doivent pouvoir être gérés par un réseau sans fil en utilisant une norme standard globale. ZigBee Alliance permet de connecter jusqu'à 255 matériels par réseau sur une portée allant jusqu'à 100 mètres. ZigBee Alliance a été ratifiée en Août 2003 sous la norme IEEE 802.15.4 [20].

- **IEEE 1451.5**

L'IEEE 1451.5 a été lancé afin de développer une norme pour les méthodes de communication sans-fil ainsi que le format de données des transducteurs (un transducteur est un moyen qui permet la conversion de l'énergie d'un type à un autre, par exemple, l'énergie magnétique en énergie électronique et vice versa) [21].

#### **4.7. LES SYSTÈMES D'EXPLOITATION POUR LES RCSF**

Plusieurs systèmes d'exploitation ont été développés pour répondre aux contraintes particulières des réseaux de capteurs, ils sont les suivants :

- **TinyOS**

TinyOS est un système d'exploitation open source conçu pour les capteurs sans fils et développé par l'Université de Berkeley. Il est basé sur une architecture à base de modules : pilotes pour les capteurs, les protocoles réseau et les services distribués. Les composants sont programmés en NesC, un langage de programmation dérivé du C adapté aux faibles ressources physiques des capteurs. Un certain nombre de plateformes sont directement programmables comme par exemple les tmote ou les MicaZ.

- **MOS**

MOS est un système d'exploitation open-source pour les capteurs développés par le MANTIS group à l'université du Colorado. MOS est développé en C. Il supporte les plateformes de la famille MICA et de la famille Telos.

- **SOS**

SOS est un système d'exploitation open-source pour les réseaux de capteurs développés par le laboratoire réseaux et systèmes embarqués de l'université de Los Angeles, il est basé sur une architecture modulaire. SOS est écrit en langage C, il supporte les plateformes de la famille MICA [22].

## **5. CONCLUSION**

Nous avons essayé à travers ce chapitre de mettre le point sur l'architecture des RCSF, ainsi que leurs principaux domaines d'application et leurs caractéristiques.

Parmi les caractéristiques des RCSF est la limitation des ressources physiques : la capacité de calcul, la capacité de stockage et la ressource d'énergie. Ces contraintes exigent que de nouveaux protocoles de réseaux soient mis au point pour les RCSF.

Dans le prochain chapitre, nous abordons l'aspect des protocoles de routages dans les RCSF.

**Chapitre 02 :**  
**Le routage dans les réseaux**  
**de capteurs sans-fil**

## 1. INTRODUCTION

En général, le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau, suivant certains critères de performance.

Les protocoles de routage au sein des RCSF sont influencés par un facteur déterminant à savoir la réduction de 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.

L'utilisation des protocoles de routage conçus pour les réseaux Ad Hoc traditionnels, pour les RCSF est inappropriée. Ceci est en raison des caractéristiques par lesquelles se distinguent les deux types de réseaux, d'où la nécessité de les améliorer ou de développer de nouveaux protocoles de routage spécifiques aux RCSF [23].

Dans ce chapitre, nous allons en premier lieu, présenter les facteurs qui influencent sur la conception des protocoles de routage au sein des RCSF. En second lieu, nous présenterons les métriques qui permettent de tester l'efficacité du protocole une fois conçu. Par la suite, nous allons détailler une classification des protocoles de routage existants pour les RCSF. Et, en fin nous allons présenter une étude détaillée sur les protocoles hiérarchiques.

## 2. FACTEURS DE CONCEPTION DES PROTOCOLES DE ROUTAGE

Comme les RCSF imposent des fonctions spécifiques et des contraintes sur les ressources, cela rend le routage plus difficile à réaliser. Quelques facteurs sont mentionnés ci-dessous :

### 2.1. TOLÉRANCE AUX PANNES

La propriété de tolérance aux pannes est définie par l'aptitude du protocole de routage à maintenir ses fonctionnalités, en cas de panne de quelques nœuds. Le but de la tolérance aux pannes est d'éviter la faille totale du système malgré la présence de fautes dans un sous ensemble de ses composants élémentaires. Les pannes sont tolérées puisqu'elles sont plus fréquentes à cause de l'épuisement rapide d'énergie et de la nature contraignante de l'environnement qui expose les nœuds à des endommagements physiques. Pour cela, les protocoles de routage conçus doivent atteindre le niveau de tolérance aux pannes requis selon

l'application. A cet effet, les protocoles de routage doivent, en cas de défaillance de liens de communication, procéder à la formation de nouvelles routes entre les nœuds.

## **2.2. CONSOMMATION D'ÉNERGIE**

Il est à noter que l'efficacité en consommation d'énergie représente un facteur de performance significatif. C'est pourquoi les concepteurs peuvent au moment du développement des protocoles, négliger les autres facteurs au détriment de cette contrainte. S'étendre sur des protocoles basés sur les techniques de conservation d'énergie lors de la communication et le calcul est essentiel.

## **2.3. LIMITATION DE CAPACITÉS DES NŒUDS**

Les nœuds ont des capacités de calcul, de stockage et de communication limitées. Les concepteurs de protocoles de routage doivent englober des opérations simples et peu exigeantes en capacité de calcul et de stockage.

## **2.4. LA MISE A L'ÉCHELLE (SCALABILITÉ)**

Les applications des RCSF nécessitent en général un déploiement dense des nœuds. Les protocoles de routage doivent donc être très scalables. Autrement dit, les protocoles de routage ne devraient pas souffrir d'une dégradation de performances dans le cas d'endommagement de nœuds aussi bien qu'avec un nombre plus élevé de nœuds.

## **2.5. CONNECTIVITÉ**

Le nombre important de nœuds dans un RCSF, fait qu'ils sont généralement dispersés de façon aléatoire, et ne sont pas uniformément répartis sur le champ de captage. Ce qui implique que certaines régions du champ de déploiement puissent bénéficier d'une meilleure connectivité. Par conséquent, les protocoles de routage conçus pour les RCSF doivent avoir une capacité d'auto-organisation qui les adapte à la distribution aléatoire des nœuds et à la topologie dynamique du réseau.

## **2.6. MODELES DE TRANSMISSION DE DONNÉES**

Les paquets de données peuvent être transmis selon quatre modèles : continu, orienté-événement, orienté-requête et hybride.

Selon le modèle continu, les nœuds capteurs envoient continuellement les paquets de données suivant un taux de transmission prédéterminé. Dans ce cas, les protocoles de routage établissent au préalable les liens de communication.

Dans le modèle orienté-événement, les nœuds capteurs envoient les paquets de données lors d'une détection d'un événement (par exemple, détection d'incendie dans une forêt) et pour le modèle orienté-requête, ils les envoient lors d'une réception d'une requête générée par le nœud puits. Les applications orienté-événement et orienté-requête sont critiques et généralement intolérantes aux délais (c'est-à-dire temps réel). Suivant ces modèles, les protocoles de routage établissent les liens de communication à la demande.

Le modèle hybride concatène les trois modèles précédents.

## **2.7. HÉTÉROGÉNÉITÉ**

Généralement, les nœuds d'un RCSF sont homogènes ayant les mêmes capacités de calcul, de mémoire et de ressources énergétiques. Ces nœuds pourront être rapidement épuisés puisqu'ils réalisent plusieurs tâches à la fois comme le captage, le traitement et le routage de données. Pour y remédier, une solution envisagée par certaines applications consiste à intégrer des nœuds spéciaux plus puissants que les autres et qui seront chargés d'effectuer les tâches les plus coûteuses en termes de ressources énergétiques. Cependant, l'intégration d'un ensemble de nœuds hétérogènes dans un seul réseau impose de nouvelles contraintes liées au routage de données. En effet, les données récoltées par ces nœuds peuvent être soumises à des fortes qualités de service, et peuvent suivre des modèles de transmission de données différents. Par conséquent, la conception des protocoles de routage doit prendre en compte les différents types de nœuds, et les contraintes qui en résultent [23].

## **3. MÉTRIQUES DE ROUTAGE**

Cette section étudie les métriques communes utilisées pour mesurer l'efficacité des protocoles de routage. Un calcul de métrique est un algorithme qui traite un coût associé à un certain chemin de routage. Les protocoles de routage permettent aux nœuds de comparer les métriques calculées afin de déterminer les routes optimales à emprunter. Plus la métrique est optimale, plus le protocole de routage considère que la probabilité d'atteindre le nœud puits à travers ce nœud intermédiaire est grande.

Plusieurs métriques peuvent affecter le routage en termes d'énergie, délai, longueur du chemin, etc. De plus, elles peuvent être considérées seules ou combinées (hybrides).

### **3.1. MÉTRIQUES POUR LA CONSOMMATION ÉNERGÉTIQUE**

Les protocoles de routage utilisent cet ensemble de métriques pour minimiser la consommation d'énergie pendant le routage [23 ,24]. L'idée est de calculer l'énergie

disponible (ED) pour chaque nœud du réseau et l'énergie nécessaire (EN) pour les transmissions des paquets entre une paire de nœuds.

Les routes entre les nœuds et le puits sont établies et chacune d'elles est caractérisée par la somme des ED des nœuds qui la constituent et par la somme des EN des liaisons qui la construisent. La consommation d'énergie suit plusieurs approches dont on peut citer :

- **Par considération de puissance**

La route choisie est celle caractérisée par la somme des ED la plus élevée.

- **Par considération du cout**

La route choisie est celle caractérisée par la plus petite somme des EN.

- **Par considération de puissance et du cout**

Cette métrique est la combinaison des deux métriques précédentes. La route choisie est celle caractérisée par la plus petite somme des EN et la plus grande somme des ED.

### 3.2. NOMBRE DE SAUTS

Les protocoles de routage utilisent cette métrique pour minimiser le nombre de sauts pendant le routage. L'idée est de calculer le nombre de nœuds intermédiaires pouvant être traversés lors d'une transmission d'un paquet du nœud source vers le nœud puits. La route choisie est celle qui contient un nombre minimum de nœuds (minimum de sauts) [24].

### 3.3. PERTE DE PAQUETS

Les protocoles de routage utilisent cette métrique dans le but de minimiser le nombre de paquets de données perdus lors du transfert depuis une source vers une destination pendant le routage. L'idée est de calculer le ratio des paquets perdus et des paquets émis transitant dans le réseau. Autrement dit, on calcule le nombre de paquets perdus sur le nombre de paquets transmis lors d'une transmission. Dans le cas où le taux de perte de paquets est élevé, il est nécessaire de mettre en place des mécanismes qui permettent de minimiser les collisions.

### 3.4. DÉLAI DE BOUT-EN-BOUT (EED)

L'EED (End-to-End Delay) est le temps moyen nécessaire pour qu'un paquet de données soit acheminé à partir de la source vers la destination. Cette technique est parmi les métriques les plus connues dans les réseaux sans fil. Les protocoles de routage l'utilisent pour minimiser le temps de propagation des paquets de données échangés pendant le routage [23].

## 4. LES CAUSES DE LA SURCONSOMMATION DE L'ÉNERGIE DANS UN RCSF

Puisque l'énergie est l'épine dorsale d'un RCSF, nous allons détailler les causes principales de la consommation de l'énergie sur un capteur afin de mieux comprendre les pistes sur lesquelles il faut agir pour rationaliser l'utilisation de cette source précieuse.

### 4.1. LA COMMUNICATION

La transmission de données entre capteurs consomme un taux important d'énergie de ses batteries, cela s'explique par le fait que :

- Les capteurs se mettent souvent en écoute passive (Idle mode) pour pouvoir accéder au medium de transmission (la technique CSMA) ce qui génère une consommation importante sans pour autant garantir un accès équitable pour l'ensemble des capteurs.
- Le sans-fil se base sur la diffusion du signal sur un rayon de transmission, et par conséquent, l'ensemble des capteurs situés à cet intervalle reçoivent le signal envoyé sans qu'ils soient destinataires de ce dernier message (Overhearing).
- La densité de déploiement des capteurs est très élevée ce qui rend le taux de collisions, entre messages envoyés, très important, et par conséquent, la nécessité d'une retransmission des messages affectés par ces collisions.
- Un RCSF est constitué de plusieurs capteurs autonomes sans aucune synchronisation centralisée, cela exige l'utilisation des messages de contrôle pour pouvoir gérer l'ensemble du réseau. Ces messages ne comprennent pas de données recensées par les capteurs ce qui les rend une source de gaspillage dans le cas d'une utilisation abusive.
- Pour optimiser les dépenses de l'énergie, des améliorations aux différents niveaux de la pile protocolaire peuvent être opérées à savoir la couche physique, la couche MAC et la couche réseau [24]. Une solution plus complète sera celle qui intègre des améliorations sur les différentes couches dans un même protocole (solution de type CrossClayer).

### 4.2. LE TRAITEMENT LOCAL DE DONNÉES

Le traitement des données consomme moins d'énergie par rapport à la communication, mais cela devient coûteux en cas d'utilisation d'algorithmes complexes ou bien en cas de traitement d'un volume considérable de données. Beaucoup de protocoles préconisent l'utilisation de fonctions d'agrégation de données sur chaque nœud ou bien sur des nœuds spécifiques CH (ClusterHeads) afin de réduire la taille des données transmises. La grande

densité de déploiement des capteurs génère souvent des données hautement corrélées qui nécessitent un filtrage pour éviter des transmissions redondantes [25].

### **4.3. L'ACQUISITION DE DONNÉES**

La fonction principale du RCSF est de surveiller une zone géographique définie en utilisant des sondes spécifiques aux applications destinataires des données. La fréquence des prélèvements opérés affecte considérablement la consommation de l'énergie sur un capteur. Par conséquent, une bonne planification de la périodicité des acquisitions de données permet une consommation d'énergie plus rigoureuse tout en répondant efficacement aux exigences des applications implémentées.

### **4.4. LA MOBILITÉ**

C'est une question clé sur les RCSF car les capteurs sont souvent embarqués sur des dispositifs mobiles comme les voitures ou les animaux. Selon le degré de cette mobilité, la topologie du réseau change d'une manière très fréquente. Ainsi, connaître ses voisins et reconfigurer le réseau nécessitent un nombre de messages important, donc une dépense plus importante d'énergie. D'autres types de RCSF compliquent d'avantage la tâche, en considérant à la fois la mobilité des capteurs et la mobilité de la station de base.

Par conséquent, la préservation de l'énergie sur un RCSF nécessite une considération individuelle ou bien globale des facteurs cités ci-dessus. Dans ce qui va suivre, nous allons présenter une classification des solutions du routage de données les plus répandues pour répondre à ces exigences [26].

## **5. CLASSIFICATION DES PROTOCOLES DE ROUTAGE DANS LES RCSF**

Récemment, les protocoles de routage conçus pour les RCSF ont été largement étudiés. Les méthodes employées peuvent être classifiées suivant plusieurs critères qui sont illustrés dans la figure suivante :

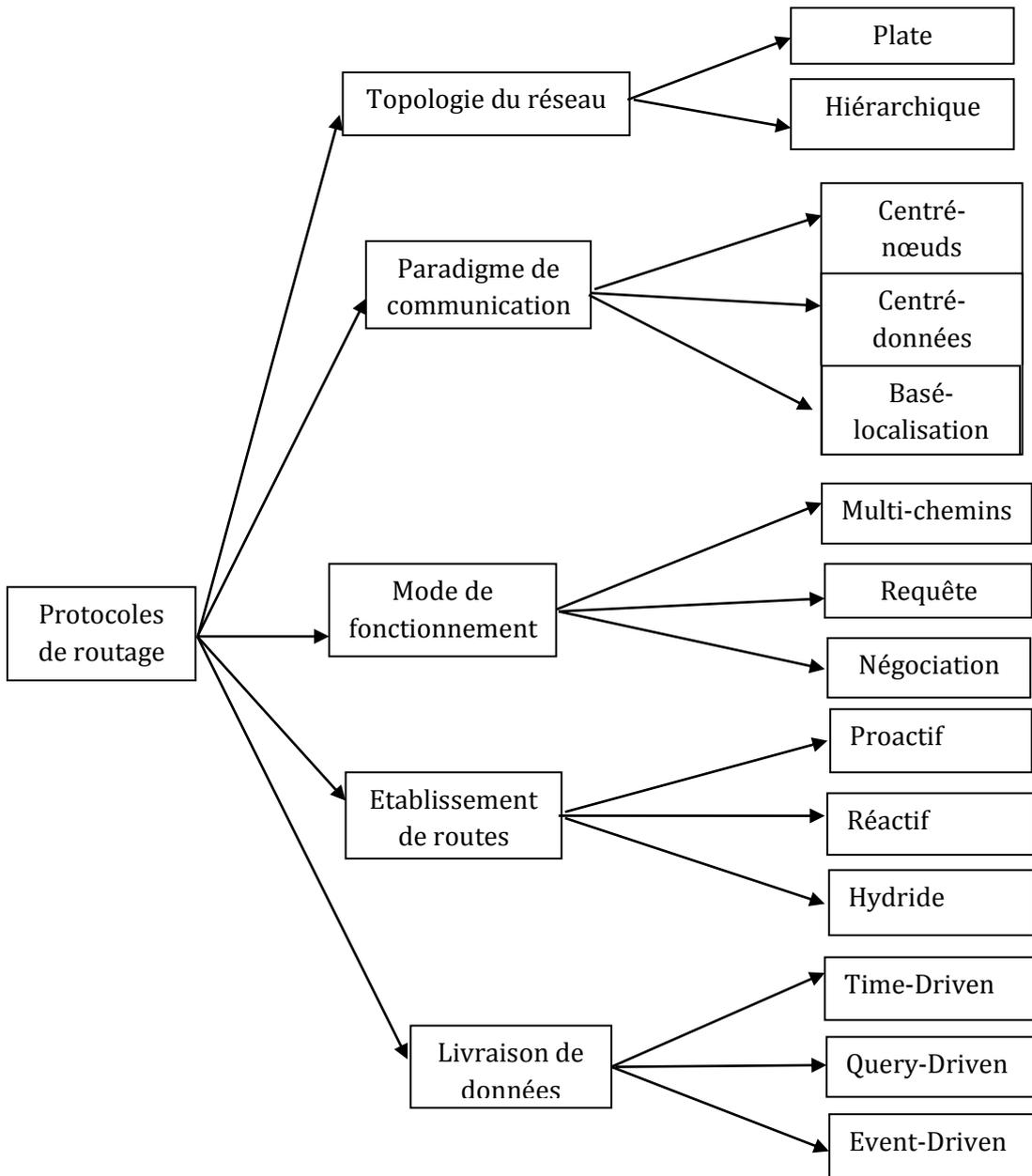


Figure 2.1 : Classification des protocoles de routage dans les RCSF.

### 5.1. CLASSIFICATION SELON LA TOPOLOGIE DU RÉSEAU

La topologie détermine l'organisation logique adaptée par les protocoles de routage afin d'exécuter les différentes opérations de découverte de routes et de transmission de données. Elle joue un rôle significatif dans le fonctionnement d'un protocole. La topologie peut être plate ou hiérarchique.

### 5.1.1. Topologie plate

Dans cette topologie tous les nœuds sont considérés homogènes et communiquent entre eux sans aucun autre intermédiaire, et seul le nœud puits est chargé de la collecte de données issues des différents nœuds capteurs afin de les transmettre vers les centres de traitement. Au cas où la destination (D) ne fait pas partie du voisinage de la source (S), les données seront transmises en utilisant des sauts multiples comme l'illustre la figure 2.2.

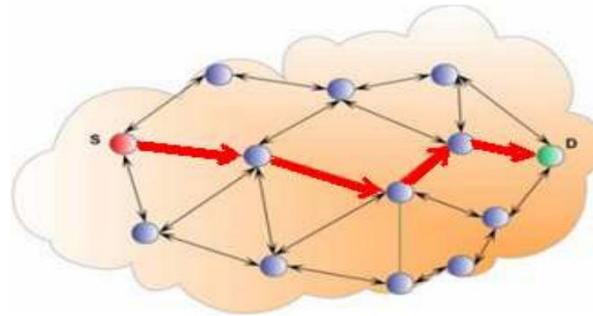


Figure 2.2 : Topologie plate [25].

Les topologies plates sont caractérisées par la simplicité des algorithmes exécutés par les protocoles de routage. Et comme les RCSF souffrent des changements brusques de la topologie, une organisation plate permet la possibilité de construire différents chemins des sources vers le nœud puits.

Cependant, les réseaux plats présentent des inconvénients comme celui défini par le problème de Hotspot [27, 28]. En effet, tous les nœuds sont homogènes et il n'y a que le nœud puits qui est chargé de la récolte d'informations, et, ces dernières passent forcément par les nœuds qui entourent le nœud puits et qui seront de ce fait épuisés. Par ailleurs, les nœuds doivent accomplir plusieurs tâches en même temps ce qui pourrait rapidement épuiser leurs ressources énergétiques et dégrader ainsi les performances du réseau.

Plusieurs protocoles rentrent dans cette catégorie comme : Directed Diffusion et SAR (Sequential Assignment Routing).

### 5.1.2. Topologie hiérarchique

Les réseaux hiérarchiques associent des rôles différents aux nœuds du réseau. Ils supposent des nœuds spéciaux plus puissants que les autres qui sont chargés d'effectuer les tâches les plus coûteuses en termes d'énergie afin d'alléger la charge sur les nœuds plus contraints en ressources énergétiques qui se consacrent uniquement au captage. De ce fait, des ensembles de ces derniers sont construits et gérés par les nœuds spéciaux appelés chefs

d'ensembles. Dans ce cas, le routage devient plus simple, puisqu'il s'agit de passer par les chefs pour atteindre le nœud puits qui leur sont directement attachés.

Comme le montre la figure 2.3, il existe deux configurations possibles pour les ensembles construits. Dans la première configuration, les membres d'un ensemble ne communiquent qu'avec leurs chefs de groupes, en obtenant ainsi un modèle basé sur les groupes. Dans la seconde, ils construisent des listes et les membres d'un ensemble utilisent d'autres nœuds comme passerelles appelés Leaders pour transmettre leurs données à leurs chefs en obtenant ainsi un modèle basé sur les listes (chaînes).

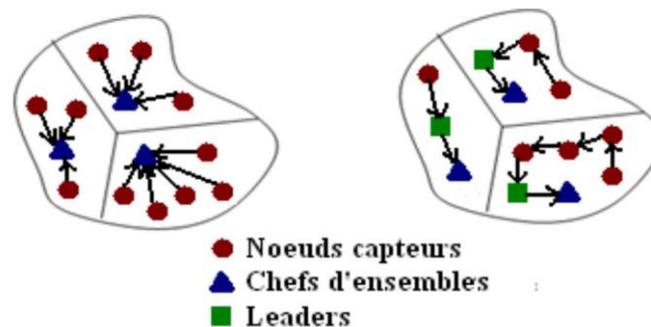


Figure 2.3 : Configurations pour les RCSF découpés en ensembles [27].

L'avantage du routage hiérarchique est que les données d'un ensemble vont être combinées par le chef d'ensemble avant leur arrivée au nœud puits ce qui allège le travail de ce dernier, ainsi que celui des nœuds qui l'entourent. De plus, contrairement aux réseaux plats, un réseau hiérarchique possède une forte scalabilité. En effet, l'ajout des nœuds ne dégrade pas les performances du réseau car le réseau peut gérer seulement les nouveaux nœuds (par exemple, en les groupant et les associant à un chef d'ensemble) sans qu'il affecte tous les nœuds restants du réseau.

Cependant, les nœuds élus comme chefs d'ensembles consomment plus d'énergie que les autres nœuds. S'ils jouent toujours le rôle d'un chef d'ensemble, ils vont être épuisés à un moment donné. Par conséquent, le réseau va être divisé ce qui implique le découpage du réseau en secteurs inaccessibles. Parmi les protocoles qui utilisent cette topologie : LEACH (Low-Energy Adaptive Clustering Hierarchy), PEGASIS (Power-Efficient Gathering in Sensor Information Systems), LCH (Layered Clustering Hierarchy protocol) [29].

## **5.2. CLASSIFICATION SELON LA MÉTHODE D'ÉTABLISSEMENT DE ROUTES**

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en catégories : protocoles proactifs, réactifs ou hybrides [30].

### **5.2.1. Protocoles proactifs**

Les protocoles de routage proactifs établissent au préalable les meilleures routes pour chaque nœud vers toutes les destinations possibles. Ces protocoles maintiennent en permanence une vision globale de l'état du réseau grâce à une gestion périodique des tables de routage et l'échange des messages de contrôle. Ceci induit un contrôle excessif d'autant plus qu'ils sont particulièrement utilisés pour les réseaux denses. De plus, ils présentent un autre inconvénient dû à la sauvegarde des routes même si elles ne sont pas utilisées.

### **5.2.2. Protocoles réactifs**

Les protocoles de routage réactifs maintiennent des routes à la demande. Lorsque le réseau a besoin d'une route, une procédure de découverte est lancée. Une fois la route n'est plus utilisée, elle sera immédiatement détruite ce qui permet une conservation d'énergie. Cependant, le routage à la demande induit une lenteur à cause de la durée nécessaire à rechercher les chemins, ce qui peut dégrader les performances des applications interactives.

### **5.2.3. Protocoles hybrides**

Les protocoles hybrides combinant les deux techniques précédentes utilisent des méthodes proactives pour l'établissement de la route dans le proche voisinage (par exemple le voisinage à deux ou trois sauts) et des méthodes réactives au delà de la zone de voisinage [31].

## **5.3. CLASSIFICATION SELON LES PARADIGMES DE COMMUNICATION**

Le paradigme de communication détermine la manière dont les nœuds sont interrogés. Dans les RCSF, il existe trois paradigmes de communication :

### **5.3.1. Centré-nœuds**

Ce paradigme est celui employé dans les réseaux conventionnels, où il est nécessaire de connaître et d'identifier les nœuds communicants (comme l'adresse IP). Les réseaux Ad-hoc utilisent ce genre de paradigme, qui s'intègre bien avec l'utilisation de ce type d'environnement. Cependant pour les réseaux de capteurs, un routage basé sur une identification individuelle des nœuds ne reflète pas l'usage réel du réseau. Pour cela, un autre

paradigme a été introduit : Centré-données. Néanmoins, le paradigme Centré-nœuds n'est pas à écarter totalement, car certaines applications nécessitent une interrogation individuelle des capteurs.

### 5.3.2. Centré-données

Dans les RCSF, la donnée est généralement plus importante que le nœud lui-même. De ce fait, le routage et l'identification, dans ce paradigme, se font en fonction des données disponibles au niveau des capteurs. Ainsi le système peut être vu comme une base de données distribuée, où les nœuds forment des tables virtuelles, alimentées par les données captées.

Le protocole Directed Diffusion (DD) est un exemple des protocoles de routage Centré-données.

### 5.3.3. Basé-localisation

Dans cette approche, les positions des nœuds représentent le moyen principal d'adressage et de routage. Dans ce cas, le routage s'effectue grâce à des techniques géométriques afin d'acheminer l'information d'une zone géographique vers une autre.

Ce type de mécanismes nécessite le déploiement d'une solution de positionnement, dont le degré de précision requis dépend de l'application ciblée. L'utilisation du GPS reste trop coûteuse dans un RCSF. Pour cela, plusieurs méthodes de localisation et de positionnement de capteurs ont été développées telles que la triangulation et la localisation par centroïdes.

On peut citer les protocoles Geographic Adaptive Fidelity (GAF) et Geographic and Energy Aware Routing (GEAR) comme exemples de protocoles position centric [32].

## 5.4. CLASSIFICATION SELON LE MODE DE FONCTIONNEMENT DU PROTOCOLE

Le mode de fonctionnement définit la manière avec laquelle les données sont propagées dans le réseau. Selon ce critère, les protocoles de routage peuvent être classifiés en quatre catégories : routage basé sur la Qualité de Service « QoS » (Quality of Service based routing), routage basé sur les requêtes (Query-based routing), routage multi-chemins (Multi-path routing), et routage basé sur la négociation (Negociation based routing).

### 5.4.1. Routage basé sur les multi-chemins

Dans cette catégorie, les protocoles de routage utilisent des chemins multiples plutôt qu'un chemin simple afin d'augmenter la performance du réseau. La fiabilité d'un protocole peut être mesurée par sa capacité à trouver des chemins alternatifs entre la source et la destination en cas de défaillance du chemin primaire. Pour cette raison, certains protocoles

construisent plusieurs chemins indépendants, c.-à-d. ils ne partagent qu'un nombre réduit (voire nul) de nœuds. Malgré leur grande tolérance aux pannes, ces protocoles requièrent plus de ressources énergétiques et plus de messages de contrôle.

#### **5.4.2. Routages basé sur les requêtes**

Dans ce type de routage, le puits génère des requêtes afin d'interroger les capteurs. Ces requêtes sont exprimées soit par un schéma valeur-attribut ou bien en utilisant un langage spécifique (par exemple SQL : Structured Query Language). Les nœuds qui détiennent les données requises doivent les envoyer au nœud demandeur à travers le chemin inverse de la requête. Les requêtes émises par le puits peuvent aussi être ciblées sur des régions spécifiques de réseau.

#### **5.4.3. Routage basé sur la négociation**

En détectant le même phénomène, les nœuds capteurs inondent le réseau par les mêmes paquets de données. Ce problème de redondance peut être résolu en employant des protocoles de routage basés sur la négociation. En effet, avant de transmettre, les nœuds capteurs négocient entre eux leurs données en échangeant des paquets de signalisation spéciales, appelés méta données. Ces paquets permettent de vérifier si les nœuds voisins disposent déjà de la donnée à transmettre. Cette procédure garantit que seules les informations utiles seront transmises et élimine la redondance des données.

#### **5.4.4. Routage basé sur la qualité de service**

Dans les protocoles de routage basés sur la QoS, le réseau doit équilibrer entre la consommation d'énergie et la qualité de données. En particulier, le réseau doit satisfaire certaines métriques de QoS, par exemple, retard, énergie, largeur de bande passante, etc.

Les protocoles de cette approche sont très recommandés pour les applications de surveillance (centrales nucléaires, application militaires, etc.) [33].

### **5.5. CLASSIFICATION SELON LE MODELE DE LIVRAISON DE DONNÉES**

Il est possible de distinguer trois modèles de livraison de données : time-driven, query-driven et event-driven.

#### **5.5.1. Time-driven**

Cette approche consiste à la livraison des données de façon périodique. Cet aspect permet aux capteurs de se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur

dispositif de capture qu'à des instants particuliers. Ainsi, la durée de vie du réseau va être allongée.

Le modèle time-driven est approprié pour des applications qui nécessitent un prélèvement périodique des données. Par exemple, cela est utile dans des applications de monitoring (feu, météo).

### 5.5.2. Query-driven

Dans les applications query-driven, la collecte d'informations sur l'état de l'environnement et la livraison des données sont initiées par des requêtes envoyées généralement par le nœud puits.

### 5.5.3. Event-driven

Ce modèle est généralement adopté dans les applications temps-réel où les capteurs doivent réagir immédiatement à des changements soudains des valeurs captées. Dans ce cas, le protocole de routage doit être réactif et doit donner des réponses rapides à l'occurrence d'un certain nombre d'évènements [34, 35].

## 6. ROUTAGE HIERARCHIQUE DANS LES RCSF

### 6.1. LES TACHES D'UN PROTOCOLE DE ROUTAGE HIERARCHIQUE

Un protocole de routage hiérarchique doit spécifier plusieurs tâches que nous pouvons classer selon :

#### 6.1.1. L'algorithme de clustering utilisé

Plusieurs algorithmes sont proposés dans la littérature, un classement en trois types de ces algorithmes est proposé dans [36] en se basant sur le type d'algorithme de contrôle exécuté :

- **Centralisé** : l'algorithme est exécuté sur le nœud qui a une vue globale du réseau, généralement, au niveau du puits. Ce type d'algorithme est peu à cause de l'overhead généré suite aux transmissions exécutées pour pouvoir garder la vue globale du réseau et la dynamique de la topologie qui fait que cette vue soit très variable. La vue obtenue par un nœud peut ne plus être valide à l'instant où le nœud la récupère à cause des pannes et des imprévisibles déconnexions.
- **Distribué** : l'algorithme est exécuté en coopération au niveau de chaque nœud du réseau. La synchronisation des tâches de contrôle est obtenue en échange des messages de contrôle. Ce type d'algorithmes minimise la communication relative à la sauvegarde de la vue globale du réseau, car chaque nœud décide, indépendamment des

autres, de son rôle et fait connaître sa décision par l'envoi de message. Cependant, l'efficacité de ces algorithmes dépend de la taille et du nombre de ces messages de synchronisation.

- **Géographique** : l'algorithme de la tâche de contrôle à exécuter par un nœud dépend de sa position géographique.

### 6.1.2. La réélection des ClusterHeads

Par sa fonction, le nœud CH consomme plus d'énergie par rapport aux autres nœuds du réseau. Le ClusterHead coordonne le fonctionnement des nœuds membres de son cluster et agrège leurs données, de ce fait, il dissipe plus d'énergie créant un déséquilibre de la distribution de l'énergie sur le réseau. Pour pallier à ce problème, une rotation de ce rôle de CH est organisée au sein du cluster ou bien au sein du réseau entier. La rotation est effectuée périodiquement ou bien en fonction de la consommation de l'énergie du nœud CH [36].

### 6.1.3. La nature des clusters générés

Les algorithmes de Clustering utilisés peuvent générer deux types de cluster : des clusters disjoints et des clusters interconnectés (Overlapped). Dans le premier type, un nœud ne peut appartenir qu'à un et un seul cluster à la fois (le cas le plus fréquent) ; sauf que pour des applications spécifiques telles que le routage inter-cluster, la localisation et la synchronisation des nœuds ; les clusters interconnectés sont utilisés. Ce type de Clustering permet aux nœuds d'appartenir à un ou plusieurs clusters à la fois.

### 6.1.4. La communication intra-cluster

La communication entre le nœud CH et les autres nœuds membres du cluster peut se faire, soit en un seul saut soit, en plusieurs sauts. Dans le cas d'une communication directe (en un saut), les paquets de données sont envoyés directement au CH. Cela suppose que les nœuds membres sont capables d'atteindre le CH en utilisant une transmission assez puissante pour une bonne réception de données. Ce type de communication engendre une consommation importante d'énergie si la distance, entre le CH et les nœuds, est grande.

Pour réduire la consommation de l'énergie, une communication en plusieurs sauts, de petites distances, est utilisée, dans ce cas chaque membre du cluster envoie ses données au plus proche membre de son cluster jusqu'à l'aboutissement au CH. Des communications de type « Unicast » sont souvent utilisées pour réduire le nombre de collisions. On trouve aussi des techniques employées au niveau MAC pour garantir un accès équitable et sans erreurs, à savoir, CDMA (Code Division Multiple Access) en utilisant un code pour chaque cluster, TDMA (Time Division Multiple Access) en allouant pour chaque nœud du cluster un temps

spécifique (dit Frame) pour envoyer ses données, ou bien, FDMA (Frequency Division Multiple Access) où chaque nœud utilise une fréquence spécifique pour l'envoi de données.

### 6.1.5. La communication inter-cluster

Les CH communiquent avec la (les) station(s) de base soit directement, soit en deux ou plusieurs sauts via des nœuds appelés généralement des « Nœuds Gateway ». Ces nœuds peuvent être des CH ou bien des nœuds membres d'un cluster. L'utilisation de la communication en multi-sauts permet de réduire la consommation d'énergie et d'augmenter la scalability du réseau.

### 6.1.6. Le niveau d'agrégation de données

Selon le type des capteurs utilisés, l'agrégation de données peut se faire à chaque nœud du réseau ou bien uniquement au niveau des CH. L'agrégation des données permet de réduire la taille des données échangées entre les nœuds, et par conséquent réduire l'énergie dépensée. Plusieurs techniques d'agrégation sont utilisées à savoir des fonctions élémentaires comme la somme, la moyenne, l'écart type, etc. ou bien des fonctions plus complexes spécifiques aux applications utilisées [25].

## 6.2. PROTOCOLES HIÉRARCHIQUES EXISTANTS

Plusieurs exemples de protocoles hiérarchiques sont décrits dans la littérature, nous allons présenter les plus référencés :

### 6.2.1. Le protocole LEACH (Low-Energy Adaptive Clustering Hierarchy)

Le protocole LEACH est le plus populaire des protocoles de routage hiérarchique, proposé par Heinzelman et al, LEACH utilise un algorithme distribué où chaque nœud décide d'une manière autonome s'il sera ClusterHead ou non en calculant aléatoirement une probabilité  $p_u$  et en la comparant à un seuil  $T(u)$  ; puis, il informe son voisinage de sa décision. Chaque nœud non ClusterHead décide du cluster à joindre en utilisant un minimum d'énergie de transmission (i.e. le plus proche). L'algorithme se déroule en plusieurs rounds et pour chaque round, une rotation du rôle du ClusterHead est initiée selon la probabilité  $p_u$  choisie et comparé à la formule suivante du seuil :

$$T(u) = \begin{cases} \frac{P}{1 - p \times \left( r \times \text{mod} \left( \frac{1}{p} \right) \right)} & \text{si } u \in G \\ 0 & \text{sinon} \end{cases}$$

Où :

$p$  : le pourcentage des CHs sur le réseau (généralement 5%).

$r$  : numéro du round en cours.

$G$  : l'ensemble des nœuds qui n'était pas CH dans les  $(1/p)$  rounds précédentes.

LEACH suppose que chaque nœud du réseau peut communiquer directement avec le puits ; alors que, les nœuds non ClusterHead ne peuvent communiquer qu'avec leurs ClusterHead choisi, en utilisant la technique TDMA instaurée par ce dernier. Cette technique permet de minimiser les collisions en allouant à chaque nœud un temps privé pour transmettre ses données vers son CH [3].

### **6.2.2. Le protocole PEGASIS (Power-Efficient Gathering in Sensor Information Systems)**

Il est considéré comme une optimisation de LEACH, proposé par Lindsey et autres en 2002, PEGASIS regroupe les nœuds du réseau sous forme d'une longue chaîne en se basant sur le principe qui stipule qu'un nœud ne peut communiquer qu'avec le nœud le plus proche de lui. Ainsi, il ajuste sa radio pour une communication très courte pour conserver son énergie. Pour communiquer avec le puits, le processus est organisé en rounds ; au cours de chaque round un seul nœud est autorisé à communiquer avec le puits directement. Ce privilège est accordé à l'ensemble des nœuds du réseau à tour de rôle [37].

### **6.2.3. Le protocole HEED (Hybrid Energy-Efficient Distributed clustering)**

Proposé par O. YOUNIS et autres en 2004, HEED s'exécute d'une manière distribuée comme LEACH à la différence du mode d'élection des CHs où il utilise une formule qui fait intervenir l'énergie et le coût de la communication. Ainsi, seuls les nœuds ayant une énergie résiduelle plus importante peuvent devenir des CHs. HEED est caractérisé par :

- Une meilleure distribution des CHs sur le réseau, car la probabilité que deux nœuds adjacents deviennent CH en même temps, est minime à la différence de LEACH.
- La consommation d'énergie est différente d'un nœud à un autre.
- La probabilité d'élire un CH peut être ajustée pour assurer une connectivité inter-CH avec un rayon de transmission prédéfini [38].

### **6.2.4. Le protocole LCH (Layered Clustering Hierarchy protocol)**

Y. Wang et B. Ramamurthy proposent, en 2007, un protocole de Clustering basé sur une hiérarchisation en forme de couches dites LCH qui fixe comme objectif la conservation de l'énergie et la scalability du réseau de capteurs.

Dans le protocole LCH les nœuds sont organisés en couches selon leurs distances, en nombre de sauts, par rapport au puits ; chaque couche exécute un algorithme distribué LEACH-like où un nombre défini de ClusterHeads est choisi. Ensuite, les nœuds non CH choisissent leurs clusters en se basant sur le coût de la communication. Les nœuds CH reçoivent les données envoyées par les membres de leurs clusters, les agrègent et les envoient vers la couche supérieure.

Une rotation des CH est effectuée pour équilibrer la consommation de l'énergie et garantir une durée de vie plus longue du réseau.

Le protocole LCH permet des performances meilleures que LEACH en éliminant la contrainte de la possibilité de communication directe entre le puits et l'ensemble des nœuds du réseau, en adoptant une communication en multi-sauts [39].

### 6.2.5. Le protocole KOCA (K-hop Overlapping Clustering Algorithm)

A. Youssef et autres proposent en 2009, un algorithme original pour construire des clusters Overlapping (qui se chevauchent) pour renforcer la robustesse du réseau, et pour répondre à des problématiques spécifiques, à savoir, la transmission inter-cluster, la localisation des nœuds et la synchronisation temporelle.

La construction des clusters overlappings d'une manière distribuée est un problème NP-difficile (une des classes de la théorie de la complexité) ; l'heuristique KOCA résout ce problème d'une façon aléatoire et distribuée. KOCA déclare être d'une complexité  $O(1)$  (une complexité constante : temps d'exécution constant) ; indépendamment de la taille du réseau.

Le problème de formation des clusters overlapping est formulé par KOCA comme une construction d'un ensemble de CHs  $\langle S \rangle$  qui satisfait trois conditions :

- **La couverture** : chaque nœud est soit un CH ou à une distance de  $k$ -sauts d'un CH;
- **L'overlapping** : pour chaque cluster, il existe au moins un autre cluster qui chevauche avec lui avec un degré  $\leq$  à un seuil  $\theta$  ;
- **La connectivité** : l'ensemble  $\langle S \rangle$  est connecté.

La solution la plus appropriée est de trouver un Minimal K-Dominating Set (MKDS) ; sauf qu'aucune solution exacte n'est disponible pour résoudre ce problème, ce qui explique l'utilisation de l'heuristique KOCA.

KOCA divise les nœuds, selon leurs rôles, en trois (03) types :

- Nœud ClusterHead (CH) : sauvegarde la topologie de son cluster, les informations sur les clusters voisins et les chemins pour les atteindre en utilisant une table (Adjacents

Clusters AC\_table) de la forme {CHID, BNL} (CHID: ID du CH, BNL: liste des nœuds BN pour atteindre ce cluster) ;

- Nœud frontière (Boundarynode BN) : appartient à plusieurs clusters en même temps, il joue le rôle d'un Gateway entre clusters voisins.
- Nœud simple (Normal Node) : appartient à un et un seul cluster.

Chaque nœud contient une table CH\_table qui sauvegarde les informations sur les CHs joignables. Si la table contient plus d'une entrée, le nœud est considéré comme BN.

L'algorithme KOCA permet une couverture et une connexion meilleures (robustesse) du réseau. Il peut contrôler la taille des clusters en variant le nombre de sauts  $k$ . Il est d'une complexité de l'ordre de  $O(1)$  indépendamment de la taille du réseau. Il génère des clusters équilibrés en taille. Ce qui permet une bonne distribution de charge sur les clusters [40].

### 6.2.6. Le protocole MCR (Multi-hop Clustering Routing protocol)

Q. Yang et autres proposent, en 2011, le protocole de Clustering MCR, basé sur l'utilisation des nœuds Gateway pour parvenir à une livraison de données vers une station de base éloignée avec un coût raisonnable en énergie.

Le protocole MCR utilise un principe de transmission inter-cluster à deux sauts, i.e., les nœuds CHs ne communiquent pas directement avec le puits, mais, ils utilisent un nœud intermédiaire (Gateway) qui se trouve dans une zone couverte par la station de base. Pour les nœuds non couverts par le puits, le protocole propose d'utiliser le même principe que le protocole HEED pour la construction des clusters.

Le protocole MCR prolonge la durée de vie des nœuds CHs, et par conséquent celle du réseau, en adoptant une communication inter-cluster en deux sauts et une réélection basée sur l'énergie résiduelle [41].

## 6.3. SYNTHÈSE DES PROTOCOLES HIÉRARCHIQUES

Nous avons présenté, dans ce chapitre, deux classes de protocoles hiérarchiques, la première classe concerne les protocoles employant une communication directe entre les ClusterHeads et la station de base dont LEACH, PEGASIS et HEED qui sont considérés comme un échantillon représentatif de cette classe.

LEACH est parmi les premiers protocoles à adopter le Clustering comme solution au problème de la consommation de l'énergie sur les RCSF, cette technique permet de réduire, considérablement, la consommation en énergie par rapport aux protocoles à topologie plate [3].

Le protocole PEGASIS, adopte une technique de routage différente de LEACH en construisant une chaîne qui relie l'ensemble des nœuds du réseau. Dans ce cas, un nœud ne peut communiquer qu'avec un autre nœud voisin, et un seul nœud de la chaîne est autorisé à communiquer avec la BS. En effet, cette technique permet une consommation d'énergie plus réduite que LEACH ; sauf que, la longueur de la chaîne peut accroître, considérablement, les délais de livraison de données, et le seul nœud autorisé à communiquer avec la BS peut devenir un passage d'étranglement au sein du réseau.

Le protocole HEED, de son côté, utilise des Clusters à un saut comme LEACH, sauf qu'il se diffère, de ce dernier, par sa technique d'élection des CHs qui utilise deux paramètres : L'énergie résiduelle des nœuds et le coût de communication.

Pour améliorer d'avantage la consommation d'énergie, une deuxième classe de protocoles hiérarchiques est présentée ; elle adopte une communication inter-Clusters à multi-sauts au lieu d'une communication directe entre les CHs et la BS. Nous avons présenté les protocoles : LCH [39], COKA [40] et MCR [41] comme échantillon de cette classe.

Le protocole LCH adopte une hiérarchisation du réseau en couches, et sur chaque couche un nombre prédéfini de Clusters sont construits en utilisant l'algorithme de Clustering du protocole LEACH.

L'algorithme de Clustering KOCA adopte des K-Clusters overlapped (non disjoints), ainsi une communication intra-Cluster en multi-sauts est utilisée, cela permet de réduire le nombre des Clusters générés tout en couvrant une surface plus importante du réseau.

Le protocole MCR prolonge la durée de vie des nœuds CHs, et par conséquence celle du réseau, en adoptant une communication inter-cluster en deux sauts et une réélection basée sur l'énergie résiduelle.

Enfin, nous présentons un tableau comparatif (Tableau 1) résumant les principales remarques de synthèse des protocoles hiérarchiques à multi-sauts étudiés [35], [36] [3].

<b>Critères de comparaison</b>	<b>Solution1 LCH 2007</b>	<b>Solution 2 COKA 2009</b>	<b>Solution 3 MCR 2011</b>
<b>Type de réseau</b>	Statique, dense, Homogène	Statique, uniforme, Homogène	Statique, homogène
<b>Nombre de Puits</b>	1	1	1
<b>Localisation du Puits</b>	/	/	Loin de la zone du captage

<b>Performances visées</b>	Optimisation d'énergie et scalability	Optimisation d'énergie, Robustesse, Scalability	Optimisation d'énergie, durée de vie
<b>Algorithme de Clustering</b>	LEACH-like	Original	HEED-like
<b>Type</b>	Distribué	Distribué	Distribué
<b>Complexité</b>		O(1)	O(1)
<b>Nature de Clusters générés</b>	Dynamiques, Disjoints en nombre prédéfini par couche (Layer)	Statiques, Over lapping, nombre de clusters over lapping prédéfini, équilibres	Dynamiques, Disjoints
<b>Election des ClusterHeads</b>	Aléatoire, Probabiliste (énergie non prise en compte)	Aléatoire, Probabiliste (énergie non prise en compte)	Aléatoire, Probabiliste (énergie prise en compte)
<b>Localisation des ClusterHeads</b>	/	Centre du cluster	/
<b>Réélection des ClusterHeads</b>	Périodique	Pas de rotation	Energy-driven
<b>Communication intra-cluster</b>			
<b>Nature</b>	Direct(un saut)	Multi sauts	Direct
<b>MAC utilisé</b>	TDMA	TDMA, CSMA/CA	TDMA
<b>Communication inter-cluster</b>			
<b>Nature</b>	Multi sauts	Multi sauts	02 sauts
<b>MAC utilisé</b>	CDMA	CSMA/CA	CDMA
<b>Gateway</b>	Possible (nœud n'appartenant pas au cluster)	Nœuds appartenant a plusieurs clusters	Nœuds à proximité du puits
<b>Mobilité</b>	Non	Non	Non

Tableau 2.1 : Tableau comparatif des solutions de routage hiérarchique multi-sauts.

## 7. CONCLUSION

Dans ce chapitre, nous avons présenté les différents protocoles de routage existants pour les RCSF. Nous avons présenté le routage hiérarchique qui est un type particulier de ces protocoles, et les techniques utilisées dans ce type de routage.

Plusieurs protocoles de routage hiérarchique ont été proposés pour les RCSF. Parmi ces protocoles, on distingue LEACH qui est un protocole particulièrement intéressant pour les RCSF. En effet, il constitue un standard sur lequel est basée la conception de plusieurs protocoles de routage. LEACH constitue l'objet principal de notre étude.

Dans le chapitre suivant, nous allons présenter une étude détaillée sur le protocole LEACH, présenter les différentes améliorations portées sur lui, et nous présenterons notre proposition qui consiste en une amélioration des performances de LEACH.

**CHAPITRE 03 :  
AMÉLIORATION DES  
PERFORMANCES DU  
PROTOCOLELEACH**

## 1. INTRODUCTION

Les protocoles de routage hiérarchiques sont considérés comme étant des protocoles très adaptés pour les RCSF en termes d'efficacité énergétique. Le routage hiérarchique reconstruit le réseau en créant une hiérarchisation logique des nœuds pour une meilleure distribution des tâches.

Le protocole LEACH proposé par Heinzelman et al [3] est le plus populaire des protocoles de routage hiérarchique. Il combine l'efficacité en consommation d'énergie et la qualité de l'accès au média. Cependant, ce protocole présente quelques insuffisances en termes d'efficacité énergétique.

Plusieurs améliorations du protocole LEACH ont été proposées par les chercheurs, dont l'objectif principal est le prolongement du temps de vie du réseau ainsi que la gestion efficace de la consommation énergétique.

Dans la première partie de ce chapitre, nous présenterons une description détaillée du protocole LEACH, son architecture de communication et son fonctionnement. Par la suite nous étudierons certaines améliorations existantes de LEACH pour prolonger la durée de vie du réseau. Et en fin nous présenterons notre proposition qui consiste à une amélioration des performances de LEACH par la combinaison de deux approches existantes.

## 2. LE PROTOCOLE LEACH (LOW-ENERGY ADAPTIVE CLUSTERING HIERARCHY)

### 2.1. PROTOCOLE MAC UTILISÉ PAR LEACH

Pendant son fonctionnement, le protocole LEACH appelle certains schémas des protocoles MAC qui seront détaillés dans cette section pour mieux comprendre son déroulement. Les nœuds doivent avoir une certaine capacité de calcul pour supporter différents protocoles MAC. Comme les RCSF ont des caractéristiques distinctes de tout autre type de réseaux sans fil, les protocoles MAC conçus pour ces derniers ne sont pas toujours applicables dans les RCSF. Deux versions des protocoles MAC pour l'accès au media sont alors proposées pour les RCSF : l'accès aléatoire et l'allocation fixe.

#### 2.1.1. Accès aléatoire

Les schémas à accès aléatoire sont à base de contention. Dans ces schémas, les nœuds qui possèdent des données à transmettre doivent essayer d'obtenir l'autorisation pour l'accès au media tout en réduisant les collisions avec les transmissions des données des autres nœuds. Le schéma d'accès multiple avec surveillance de porteuse CSMA (Carrier Sense Multiple Access) sur lequel se base le protocole LEACH est l'un des schémas d'accès aléatoire.

Lorsqu'un nœud veut transmettre un message, il examine le média pour vérifier s'il est libre ou occupé par un autre nœud. Dans le cas où le média est libre, ce nœud pourra émettre son message afin d'éviter les collisions. Cela dit, des nœuds peuvent émettre des données en même temps, ce qui mène à des collisions. Il est nécessaire donc que celles-ci soient détectées et que la récupération de données soit effectuée et que ces données soient retransmises. Si les retransmissions se passent encore en même temps, d'autres collisions vont se produire.

Une solution à ce problème consiste à introduire que chaque nœud attende un délai aléatoire avant de retransmettre ses données, ce qui réduit la probabilité d'une autre collision.

### **2.1.2. Allocation fixe**

Les schémas à allocation fixe permettent d'allouer pour chaque nœud le média de transmission suivant des intervalles de temps (schéma TDMA) ou un schéma de codage particulier (schéma CDMA).

Étant donné que chaque nœud est attribué en exclusivité à un intervalle, il n'y a presque pas de collisions entre les données. Toutefois, les schémas à allocation fixe s'avèrent inefficaces lorsque tous les nœuds n'ont pas de données à transmettre. En effet, ces intervalles sont affectés à des nœuds qui n'ont pas besoin de les utiliser.

- **TDMA**

Le schéma d'accès multiple à répartition de temps ou TDMA (Time Division Multiple Access) permet de diviser le temps en intervalles (time-slot) attribués à chaque nœud. Ainsi, un seul nœud a le droit d'accès au canal (il utilise toute la plage de la bande passante du canal), mais doit émettre ses données pendant les intervalles de temps qui lui sont accordés [42].

- **CDMA**

Le schéma d'accès multiple par répartition en code ou CDMA (Code Division Multiple Access) permet de côtoyer plusieurs nœuds simultanément. En effet, il ne divise ni la plage de fréquences ni l'intervalle de temps. Ainsi, des nœuds peuvent émettre leurs données continuellement et selon une large plage de fréquence. Le protocole CDMA utilise des techniques d'étalement de spectre afin d'éviter les collisions entre les transmissions simultanées des nœuds [29].

## **2.2.ARCHITECTURE DE COMMUNICATION DE LEACH**

LEACH est un protocole autoorganisateur basé sur le clustering adaptatif, qui utilise la rotation randomisée des têtes de grappe pour distribuer équitablement la charge d'énergie entre les nœuds capteurs dans le réseau. Il est considéré comme étant l'une des premières

approches de routage hiérarchique basées sur le clustering. LEACH est fondé sur deux hypothèses de base :

- La station de base est fixe et est placée loin des capteurs,
- Tous les nœuds du réseau sont homogènes et limités en énergie.

L'idée derrière LEACH est de former des clusters des 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 cluster".
- **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.

Comme la figure 3.1 l'indique, les nœuds sont chargés de collecter des données, les envoyer à leurs CHs qui les agrègent et transmettent, à leur tour, les résultats d'agrégation au nœud puits selon une communication unicast (à un seul saut).

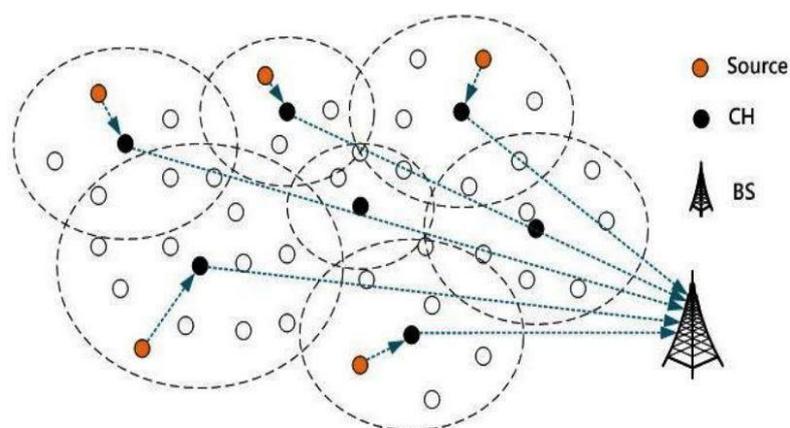


Figure 3.1 : Architecture de communication de LEACH [43].

Les CHs ont pour mission d'assurer les fonctions les plus coûteuses en énergie, à savoir la communication avec le nœud puits qui est supposé éloigné, ainsi que tous les traitements de données (agrégation, fusion et transmission de données) afin de réduire la quantité des données transmises. Cela permet d'économiser l'énergie puisque les transmissions sont

uniquement assurées par les CHs plutôt que par tous les nœuds du réseau. Par conséquent, LEACH réalise une réduction significative de la dissipation d'énergie [44].

### 2.3.ALGORITHME DÉTAILLÉ DE LEACH

L'algorithme se déroule en « rounds » qui ont approximativement le même intervalle de temps déterminé au préalable. Chaque round est constitué d'une phase d'initialisation et d'une phase de transmission.

#### 2.3.1. Phase d'initialisation

Comme l'indique la figure 3.2, la phase d'initialisation est composée de 3 sous-phases : d'annonce, d'organisation des groupes et enfin d'ordonnancement.

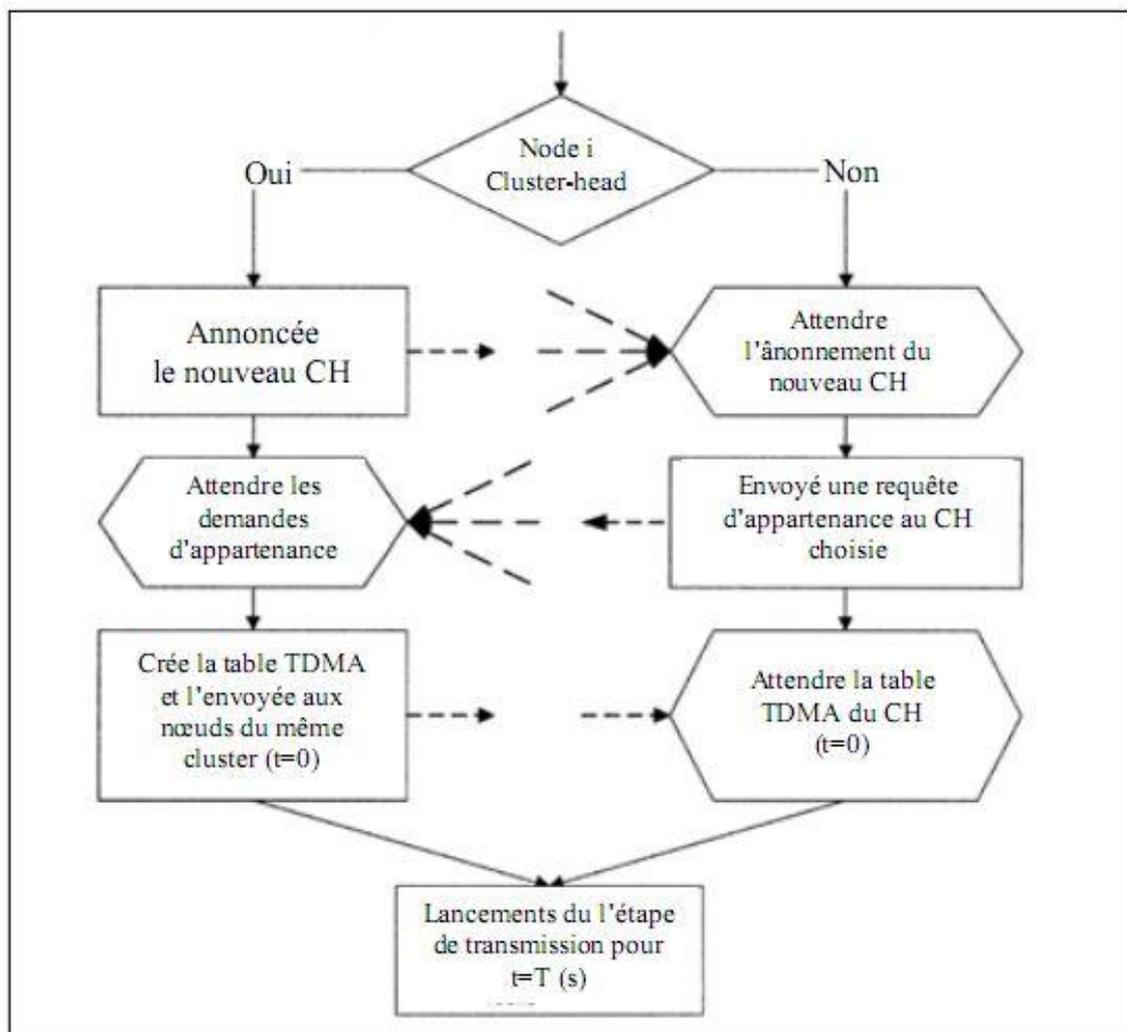


Figure 3.2 : Opérations de l'étape d'initialisation de LEACH [44].

##### 2.3.1.1.Phase d'annonce

Avant de lancer cette phase, on désire avoir un certain nombre de CHs. Ce nombre, que l'on note  $K$ , est fixe et il est inchangé durant tous les rounds. On estime que le pourcentage optimal du nombre de CHs désirés devrait être de 5% à 15% du nombre total de

nœuds [43]. Si ce pourcentage n'est pas respecté, cela mènera à une grande dissipation d'énergie dans le réseau. En effet, si le nombre de CHs est très élevé, on aura un nombre important de nœuds (CH) qui se consacrent aux tâches très coûteuses en ressources énergétiques. Ainsi, on aura une dissipation d'énergie considérable dans le réseau. De plus, si le nombre de CHs est très petit, ces derniers vont gérer des groupes de grandes tailles. Ainsi, ces CHs s'épuiseront rapidement à cause de travail important qui leur est demandés.

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

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

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

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

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

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

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

$$\sum_{i=1}^N C_i(t) = N - K * (r \bmod N/K) \dots \dots (2)$$

Utilisant l'équation (1) et (2), le nombre de CHs par round est :

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

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

$$P_i(t) = \frac{E_i(t)}{E_{\text{total}}(t)} K \dots \dots (3)$$

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

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

Les équations (2) et (3) seront égales si les nœuds commencent avec la même énergie. De plus, en utilisant l'équation (3), les nœuds requièrent des informations sur toute l'énergie disponible dans le réseau.

### 2.3.1.2. Phase d'organisation de groupes

Après qu'un nœud soit élu CH, il doit informer les autres nœuds non-CH de son nouveau rang dans le round courant. Pour cela, un message d'avertissement ADV contenant l'identificateur du CH est diffusé à tous les nœuds non-CH en utilisant le protocole MAC CSMA pour éviter les collisions entre les CHs. La diffusion permet de s'assurer que tous les nœuds non-CH ont reçu le message. Par ailleurs, elle permet de garantir que les nœuds appartiennent au CH qui requière le minimum d'énergie pour la communication. La décision est basée donc sur l'amplitude du signal reçu ; le CH ayant le signal le plus fort (i.e. le plus proche) sera choisi. En cas d'égalité des signaux, les nœuds non-CH choisissent aléatoirement leur CH.

Chaque membre informe son CH de sa décision. Une fois que le CH ait reçu la demande, il lui envoie un message d'acquiescement Join- REQ [29].

### 2.3.1.3. Phase d'ordonnancement

Après la formation des groupes, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données au sein de son groupe. Il crée un ordonnanceur (schedule) TDMA et assigne à chaque nœud membre un slot de temps durant lequel il peut transmettre ses données. L'ensemble des slots assignés aux nœuds d'un groupe est appelé frame. La durée de chaque frame diffère selon le nombre de membres du groupe.

Par ailleurs, afin de minimiser les interférences entre les transmissions dans des groupes adjacents, chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA. Il le transmet par la suite à ses membres afin de l'utiliser pour leurs transmissions. [46].

### 2.3.2. Phase de transmission

Cette phase est plus longue que la phase précédente, et permet la collecte de données captées. En utilisant l'ordonnanceur TDMA, les membres émettent leurs données captées pendant leurs propres slots. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leur énergie. Ces données sont ensuite agrégées par les CHs qui les fusionnent et les compressent, et envoient le résultat final au nœud puits. Après un certain temps prédéterminé, le réseau va passer à un nouveau round. Ce processus est répété jusqu'à ce que tous les nœuds du réseau seront élus CHs, une seule fois, tout au long des rounds précédents. Dans ce cas, le round est réinitialisé à 0.

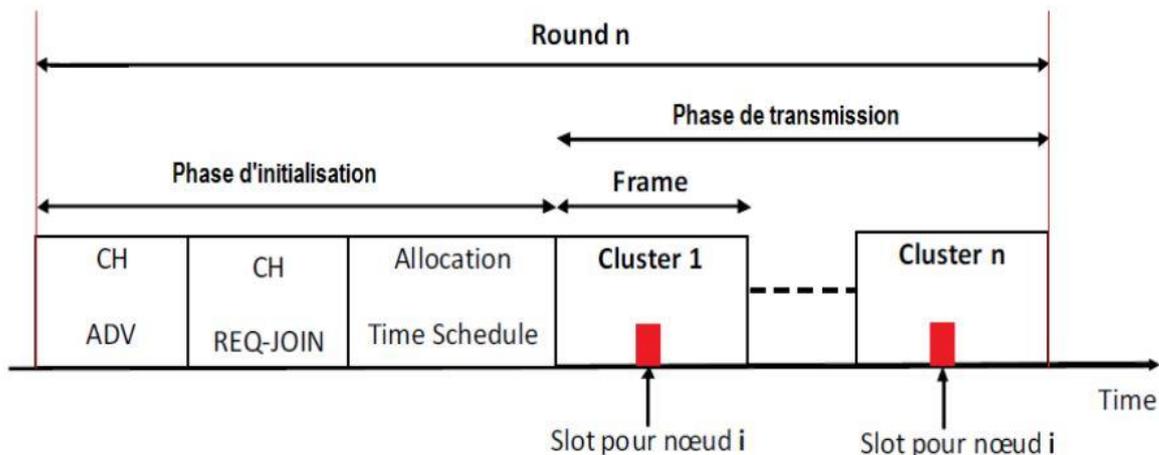


Figure 3.3 : Répartition du temps et différentes phases pour chaque round [45].

## 2.4. AVANTAGES ET INCONVÉNIENTS DE LEACH

### 2.4.1. Avantages

Le protocole LEACH présente les avantages suivants :

- L'auto-configuration des clusters se fait indépendamment de la station de base (algorithme distribué).
- La consommation d'énergie est partagée sur l'ensemble des nœuds prolongeant ainsi la durée de vie du réseau.
- L'utilisation des techniques TDMA/CDMA permet d'avoir une hiérarchie et de réaliser des clusterings sur plusieurs niveaux. Ces derniers permettent d'économiser davantage l'énergie.
- Rotation randomisée de la grappe effectuée par "les stations de base" ou "les têtes de cluster".
- Compression locale (agrégation) : Les nœuds CHs 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 [29].

### 2.4.2. Inconvénients

LEACH a les inconvénients suivants :

- Sans justifier leur choix, les auteurs fixent le pourcentage optimal de CHs pour le réseau à 5% du nombre total des nœuds. Néanmoins, la topologie, la densité et le nombre de nœuds peuvent être différents dans d'autres réseaux.
- Aucune suggestion n'est faite à propos du temps de réélection des CHs (temps des itérations).
- Les CHs les plus éloignés de la station de base meurent rapidement par rapport à ceux qui sont proches de la station.
- On pourra ne pas avoir des CHs durant un round si les nombres aléatoires générés par tous les nœuds du réseau sont supérieurs à la probabilité  $P_i(t)$ .
- L'utilisation d'une communication à un seul saut au lieu d'une communication multi-sauts diminue l'énergie des nœuds.
- Le CH est toujours allumé et lorsqu'il meurt, le groupe deviendra inutile car les données recueillies par les nœuds du cluster ne sauront jamais atteindre la station de base.

- LEACH ne garantit pas une distribution homogène des CHs sur le réseau, car le seul critère d'élection du CH est une probabilité aléatoire. Cela n'empêche pas une concentration des CHs dans une région limitée au détriment de l'ensemble du réseau.
- Le protocole LEACH n'est pas sécurisé, il est très vulnérable même aux simples attaques [47].

Dans la section suivante, nous étudions certaines améliorations existantes du protocole LEACH pour prolonger la durée de vie des capteurs.

### **3. AMÉLIORATIONS EXISTANTES DE LEACH**

Nous avons vu, dans la section précédente que le protocole LEACH présente beaucoup d'avantages en termes d'efficacité énergétique. Mais, il présente aussi quelques insuffisances qui permettent la dissipation de l'énergie du réseau.

Par conséquent, un grand nombre d'améliorations de LEACH ont été proposées par de nombreux chercheurs. Nous présenterons, ici, quelques algorithmes proposés et qui sont portés sur l'amélioration de la méthode de sélection des cluster-heads.

#### **3.1.LEACH-C (LEACH CENTRALISED)**

LEACH choisit seulement un nombre fractionnel de nœuds capteurs pour servir en tant que chefs de cluster. Le reste des nœuds rejoignent leurs clusters appropriés selon la force du signal des CHs. La limitation de LEACH est qu'il n'offre aucune garantie sur le placement ou le nombre des CHs. Par conséquent, pour une meilleure performance, une amélioration du protocole LEACH, qui utilise un algorithme de clustering centralisé, appelée LEACH-Ca été proposée par Wu and Wang [48].

LEACH-C est un algorithme de clustering centralisée similaire à LEACH. Dans LEACH-C, chaque nœud envoie les informations sur sa position actuelle et son niveau d'énergie à la station de base. La station de base utilise cette information globale du réseau et construit les meilleurs clusters (groupes) qui utilisent uniquement le moins d'énergie possible pour la transmission de données. Cet algorithme à besoin de GPS (Système de Positionnement Global) ou d'une autre technique de localisation. La station de base ne permet qu'aux nœuds ayant suffisamment d'énergie de concourir dans le choix des CHs, puis diffuse l'information à tous les nœuds du réseau. LEACH-C utilise un algorithme de seuil déterministe qui utilise la quantité d'énergie dans le nœud et l'information sur sa récente participation en tant que CH. L'algorithme de contrôle est centralisé pour produire les meilleurs groupes par la distribution des nœuds CHs partout dans le réseau [49].

### **3.2.EEE-LEACH(Energy Efficient Extended LEACH)**

EEE-LEACH (Energy Efficient Extended LEACH) proposé par Wairagu G. Richard [50], est une approche qui utilise la technique de clustering (regroupement) à plusieurs niveaux pour améliorer l'efficacité énergétique du réseau en réduisant la distance de la communication radio.

Dans ce regroupement multi-niveau, une première couche de clusters est formée entre les nœuds et la station de base en utilisant l'algorithme LEACH. Dans la première couche les CHs sont sélectionnées, puis les nœuds normaux envoient leurs propres données à leurs CHs correspondants et les CHs utilisent le mécanisme de fusion de données pour agréger les données reçues.

Dans la deuxième couche de clustering, les cluster-heads maîtres (MCH) sont élus. Lorsque les CHs fusionnent et agrègent les données, ils cherchent les MCHs les plus proches, en calculant la distance entre eux, et puis envoient les données agrégées au MCH correspondant. Une fois le MCH reçoit les données des CHs les plus proches, il cumule toutes les données reçues, les compresse et les transmet à la station de base (BS).

Le nombre total de CHs et MCHs est principalement décidé en utilisant une valeur fractionnelle  $p$  prédéterminée (valeur de probabilité de l'élection) pour les CHs et  $p_m$  (valeur de probabilité de l'élection pour les MCHs). Dans l'algorithme EEE-LEACH, le nombre de MCHs est sélectionné comme inférieur au nombre de CHs afin de réduire la distance de communication globale entre le nœud capteur et la station de base.

Les diverses expériences effectués par les auteurs du protocole EEE-LEACH [50] prouvent que le réseau à un temps de vie et une efficacité énergétique meilleures que le protocole LEACH.

### **3.3.DISTANCE BASED CLUSTER FORMATION TECHNIQUE FOR LEACH**

Ce protocole est proposé par Snehal Kole et al. [51], au cours de la formation de clusters dans l'algorithme LEACH, certains nœuds doivent sélectionner des CHs qui ont une plus longue distance avec la BS comparées à eux. Dans ce cas, les données sont envoyées au CH vers la direction inverse de la BS, et donc les données doivent parcourir une plus longue distance pour atteindre la BS. Ces transmissions sont appelés transmissions supplémentaires et cela affecte sur la durée de vie de réseau en dépensant l'énergie du nœud, comme le montre la figure 3.4.

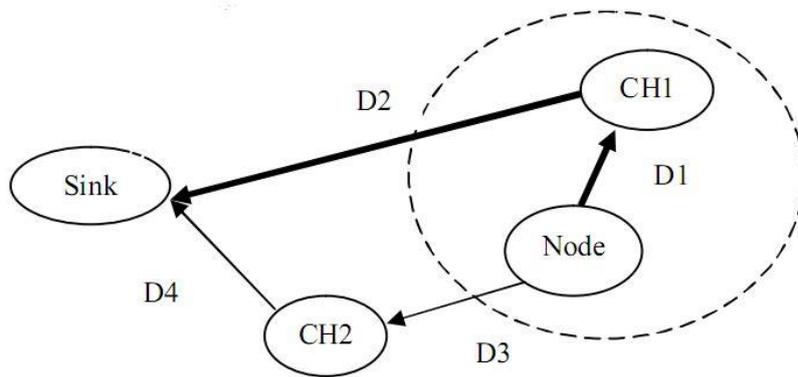


Figure 3.4 : Transmissions supplémentaires [51].

Dans la figure ci-dessus, il existe deux chemins du nœud vers le nœud puits (Sink): Path1 = (Nœud-CH1, CH1-Sink) et Path2 = (Nœud-CH2, CH2-Sink). La longueur de Path1 est supérieure à la longueur de Path2 c.à.d.  $(D1 + D2) > (D3 + D4)$  où  $D1 < D3$ . Selon LEACH, le nœud rejoint le CH le plus proche de lui CH1 même si la distance résultante :  $(D1+D2) > (D3+D4)$ .

Pour prolonger la durée de vie du réseau et résoudre ce problème de transmission supplémentaire, l'algorithme proposé estime la distance entre le nœud et le premier CH plus la distance entre ce CH et le nœud puits, comparée à la distance entre le nœud et le deuxième CH plus la distance entre ce CH et le nœud puits, durant la formation de clusters. En considérant le même exemple, si un nœud rejoint le CH2 au lieu de CH1, même si CH1 est plus proche du nœud, alors la transmission supplémentaire peut être minimisée, puisque :  $(D3+D4) < (D1+D2)$ , comme le montre la figure 3.5 :

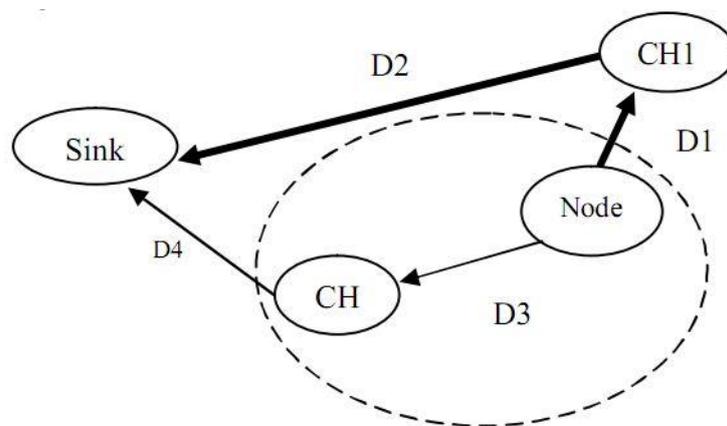


Figure 3.5 : Exemple montrant la minimisation de la transmission supplémentaire [51].

Le pseudo code suivant présenté dans [51] décrit le fonctionnement du protocole « DISTANCE BASED CLUSTER FORMATION TECHNIQUE FOR LEACH » :

- **Pseudo code:**

**Notations:**

S: Set of nodes in network

CH: Cluster Head

K: Number of Cluster heads

$S_{CH}$  : Set of Cluster Heads

$S_{nCH}$  : Set of non Cluster Heads

**Algorithm:**

```

For every node n in S
do {
r ← random (0,1)
if (  $E_{init}(n) > 0$  &  $r \bmod (1/p_{opt}) \neq 0$  ) then
Compute T(n)
if (  $r < T(n)$  &  $K < \text{no. of CH}$  ) then
CCH {n} = TRUE
else
CCH {n} = FALSE
end if
end if
if (CH {n} = TRUE) then
send  $ADV_i$  to all  $n_i \in S$ 
for every CH in  $S_{CH}$ 
if {  $\text{dist}(n, CH_i) + \text{dist}(CH_i, BS) < \text{min\_dist}$  }
setmin_dist =  $\text{dist}(n, CH_i) + \text{dist}(CH_i, BS)$ 
send join( $ID_i$ ) to all  $n_i \in S_{nCH}$ 
CH sends TDMA slots to  $S_{nCH}$  in Cluster (C)
 $S_{nCH}$  send data to  $S_{CH}$ 
 $S_{CH}$  aggregate data & send it to BS
}

```

Dans cette solution, la distance entre le nœud et la station de base est considérée comme un facteur important dans la formation de clusters, ce qui permet de réduire certaines transmissions supplémentaires dans le protocole LEACH initial. L'analyse des résultats de simulation présentée dans [51] prouve l'amélioration des performances du protocole LEACH en termes de consommation d'énergie et durée de vie du réseau.

### 3.4. MODLEACH (MODIFIED LEACH)

Le protocole MODLEACH est proposé par D.Mahmood et al [52]. Il introduit deux techniques pour augmenter la durée de vie du réseau et le débit de transmission.

Dans le protocole LEACH, le changement de CHs est effectué à chaque round (tour) et une fois un nœud a été élu comme CH, il ne pourra pas l'être durant  $1/p$  (où  $p$  est la probabilité d'un nœud à être CH), tours suivants, c.à.d. jusqu'à ce que tous les autres nœuds du réseau passent par le rôle de CH. Dans chaque tour, les CHs seront remplacés et tous le processus de formation de groupes est repris.

La première technique proposée dans le protocole MODLEACH consiste à une nouvelle méthode de remplacement de CHs. Si le CH courant n'a pas dépensé une grande énergie durant ce tour et a suffisamment d'énergie restante, il restera CH pour le prochain tour. Donc, au commencement de chaque tour, le protocole vérifie si l'énergie restante du CH est supérieure à un seuil prédéfini, il ne sera pas remplacé et il restera CH pour le prochain tour, ce qui permet d'économiser l'énergie dépensée dans la transmission de paquets nécessaires à la formation de nouveaux clusters. Si l'énergie restante du CH est inférieure au seuil, le CH sera remplacé par un autre nœud en utilisant l'algorithme de LEACH.

La deuxième technique consiste à l'utilisation de deux différentes puissances des signaux de transmission. Le protocole utilise des amplifications de différentes puissances pour transmettre les données, en se basant sur la distance de la destination. L'énergie d'amplification minimale nécessaire pour la communication inter-cluster (entre les CHs) ou pour la communication entre le CH et la BS, et l'énergie d'amplification nécessaire pour la communication intra-cluster (entre les nœuds d'un cluster) ne peuvent pas être les mêmes. L'utilisation d'une faible puissance d'énergie pour les transmissions intra-cluster par rapport aux transmissions inter-cluster ou entre le CH et la BS, conduit à économiser beaucoup d'énergie.

Les résultats présentés par [52] montrent que le protocole MODLEACH donne un meilleur débit, une augmentation du temps de vie du réseau, et un meilleur régime de remplacement de CHs, par rapport à LEACH.

## 4. AMÉLIORATION PROPOSÉE « DistModLeach »

Après avoir étudié les différentes améliorations portées sur le protocole LEACH, nous avons essayés de combiner les avantages de deux algorithmes existants. Le premier est « Distance Based Cluster Formation Technique For Leach » proposé par Snehal Kole et al. [51] et le deuxième est « MODLEACH » proposée par D.Mahmood et al. [52].

Dans la solution « Distance Based Cluster Formation Technique For Leach », la distance entre le nœud et la station de base est considérée comme un facteur important dans la formation de clusters, ce qui permet de réduire certaines transmissions supplémentaires dans le protocole LEACH initial. Et par conséquent l'amélioration des performances du protocole LEACH en termes de consommation d'énergie et durée de vie du réseau.

Dans la solution « MODLEACH », un CH ne sera pas remplacé et il restera CH pour le prochain tour s'il a suffisamment d'énergie, ce qui permet d'économiser l'énergie dépensée dans la transmission de paquets nécessaires à la formation de nouveaux clusters.

Nous proposons de concevoir un nouveau protocole « DistModLeach » qui combine les deux approches précédentes. Notre proposition « DistModLeach » consiste à l'utilisation du protocole « Distance Based Cluster Formation Technique For Leach » à la base, et de remplacer la phase de sélection de CHs par celle proposée dans « MODLEACH ».

### 4.1. LES GRANDES ETAPES DU PROTOCOLE « DistModLeach »

Le déroulement de notre protocole hybride « DistModLeach » est similaire à celui de LEACH, avec quelques modifications portées sur certaines étapes de l'algorithme. L'algorithme se déroule en « rounds » qui ont à peu près le même intervalle de temps déterminé au préalable. Chaque round est constitué d'une phase d'initialisation dans laquelle les clusters sont formés et les CHs sont élus, suivie d'une phase de transmission où les données collectées sont transmises aux CHs qui vont à leur tour les transmettre à la station de base.

Les nœuds doivent être tous synchronisés de façon à participer à la phase d'initialisation en même temps.

#### 4.1.1. Phase d'initialisation

La phase d'initialisation est composée de 3 sous-phases : d'annonce, d'organisation des groupes et enfin d'ordonnancement.

##### 4.1.1.1. Phase d'annonce

La phase d'annonce commence par l'annonce du nouveau round par le nœud puits, et, par la prise de décision locale d'un nœud pour devenir CH. Chaque nœud  $i$ , vérifie s'il a été élu

comme CH dans le tour précédent ou pas. Si le nœud a été CH dans le tour précédent, il vérifie si son énergie restante est supérieure à un seuil prédéfini, si c'est le cas donc le nœud se réélue comme CH. Si l'énergie du nœud est insuffisante, le nœud ne se désigne pas comme CH.

Si le nœud  $i$  n'a pas été CH dans le tour précédent, la prise de décision locale pour devenir CH est similaire à celle de LEACH, avec une certaine probabilité  $P_i(t)$  au début du round  $r+1$  qui commence à l'instant  $t$ . Le nœud  $i$  génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à  $P_i(t)$ , le nœud deviendra CH durant le round  $r+1$ .  $P_i(t)$  est calculé en fonction de  $K$  et de round  $r$ , et à la différence de LEACH la condition  $P_i(t)=0$  si le nœud  $i$  a déjà été CH durant l'un des  $(r \bmod N/K)$  rounds précédents, est enlevée.

$$P_i(t) = \begin{cases} \frac{K}{N - K * (r \bmod N/K)} \end{cases}$$

Où  $N$  est le nombre total de nœuds dans le réseau.

Après un certain temps, si l'énergie restante dans la majorité des nœuds (plus que  $N-K$ ) est inférieure au seuil prédéfini, c.à.d. le nombre de CHs devient insuffisant, l'algorithme proposé est remplacé par celui de LEACH.

#### **4.1.1.2. Phase d'organisation**

Dans la phase d'organisation, si un nœud a déjà été CH dans le tour précédent et n'a pas été remplacé par un autre, donc il n'a pas besoin d'informer les autres nœuds du réseau, et le cluster formé dans le tour précédent restera le même.

Si un nœud a été CH dans le tour précédent et n'a pas été élu pour le tour courant, il doit informer les nœuds qui appartenaient à son cluster pour les libérer.

Si un nouveau nœud est élu CH, il doit informer les autres nœuds non-CH de son nouveau rang dans le round courant d'une façon similaire à LEACH. Un message d'avertissement ADV contenant l'identificateur du CH est diffusé à tous les nœuds non-CH en utilisant le protocole MAC CSMA pour éviter les collisions entre les CH.

A la différence de LEACH la décision d'un nœud d'appartenir à un CH, se fait premièrement en vérifiant si le nœud appartient déjà à un CH ou pas. Si le nœud n'appartient à aucun CH, la décision se fait en utilisant la proposition du protocole (Distance Based Cluster Formation Technique for LEACH). Pour chaque CH, le nœud calcule la distance entre lui et le CH plus la distance entre ce CH et le nœud puits. Le CH choisit est celui qui

correspond à la distance minimale calculée par le nœud. En cas d'égalité de distances, les nœuds non-CH choisissent aléatoirement leur CH.

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

#### **4.1.1.3. Phase d'ordonnancement**

La phase d'ordonnancement est la même que celle de LEACH. Après la formation des groupes, chaque CH agit comme un centre de commande local pour coordonner les transmissions des données au sein de son groupe, en utilisant un ordonnanceur (schedule) TDMA.

#### **4.1.2. Phase de transmission**

Une fois que l'étape d'initialisation est terminée, la phase de transmission commence. Dans cette phase, les données captées sont transmises à la station de base, cela se fait de la même manière que dans le protocole LEACH en utilisant l'ordonnanceur TDMA. Les nœuds membres émettent leurs données captées pendant leurs propres slots. Ces données sont ensuite agrégées par les CHs qui les fusionnent et les compressent, et envoient le résultat final au nœud puits.

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

### **4.2.PSEUDO CODE**

Le pseudo code suivant décrit le fonctionnement de notre protocole proposé «DistModLeach» :

#### **Notations:**

n : nœud

CH : cluster head

CHch = cluster head choisi

BS : station de base

CH {n} : l'état d'un nœud (la valeur initiale est faux)

dist : la distance entre deux nœuds

Mindist (n) : la distance minimale entre nœud, CH et SB

$ERES_n$  : l'énergie restante d'un nœud

seuil : le seuil d'énergie

$P_n(t)$  : probabilité d'un nœud pour être CH

cluster : un groupe de nœuds avec un CH

**Algorithme :**

/\* Ses étapes sont appliquées à tous les nœuds du réseau \*/

Répéter{

/\* Génération du nombre aléatoire \*/

$r \leftarrow \text{random}(0,1)$

/\* Tester si le nœud a été CH ou pas \*/

Si (CH {n} = vrai){

/\* Comparaison de l'énergie restante du nœud et le seuil d'énergie \*/

Si( $ERES_n > \text{seuil}$ )

/\* Le nœud reste CH \*/

CH {n} = vrai

Sinon {

/\* Le nœud devient non CH \*/

CH {n} = faux

Envoyer les messages de libération aux nœuds du cluster

}}

Sinon

{

/\* Calcul du nombre aléatoire  $P_n(t)$  \*/

Calculer  $P_n(t)$

/\* Comparaison entre le nombre aléatoire et la probabilité d'un nœud pour être CH \*/

Si ( $r < P_n(t)$ ) {

/\* le nœud devient CH \*/

CH{n} = vrai

/\* Informer le réseau \*/

Envoyer les messages d'avertissement à tous les nœuds

Recevoir les demandes d'appartenance

Envoyer les acquittements aux nœuds du cluster

}

```

Sinon
/* Le nœud n'est pas un CH*/
CH {n} = faux
}

Si (CH {n} = faux & CH {n} ∉ cluster)
{
Pour chaque CH dans le réseau {
/* Chercher le CH qui correspond à la distance minimale entre : nœud, CH
et BS */

Si (dist(n,CH)+dist(CH,BS)<Mindist(n)) {
Mindist(n) = dist(n,CH)+dist(CH,BS)
CHch = CH
}}

Envoyer une demande d'appartenance à CHch
}

Transmission de données
}

```

## 5. CNCLUSION

Dans ce chapitre, nous avons présenté le protocole de routage hiérarchique LEACH, ainsi que les solutions les plus importantes proposées pour améliorer la formule de sélection de CHs utilisée par le protocole LEACH, afin de prolonger la durée de vie des RCSF.

Nous avons proposé une amélioration des performances du protocole LEACH appelée « DistModLeach », qui consiste en la combinaison du protocole « Distance Based Cluster Formation Technique for Leach » avec une partie du protocole « MODLEACH » afin d'avoir une moindre consommation énergétique et prolonger la durée de vie du réseau.

Dans le chapitre suivant nous allons passer à l'implémentation des algorithmes « LEACH », « Distance Based Cluster Formation Technique for Leach », et l'algorithme « DistModLeach » que nous avons proposé, afin d'évaluer leurs performances et comparer les résultats, en utilisant le simulateur NS2.

# **Chapitre 04 :**

## **Réalisation**

## 1. INTRODUCTION

Nous avons présenté dans le chapitre précédent, deux améliorations existantes du protocole de routage LEACH, ainsi que notre amélioration proposée « DistModLeach » qui consiste en une combinaison de ces deux solutions afin de minimiser la consommation d'énergie dans les RCSF.

Nous allons présenter dans ce chapitre, les simulations effectuées pour évaluer les performances des protocoles précédents et consolider nos affirmations. Pour cela, nous avons choisi le simulateur NS2 qui est le plus populaire des simulateurs dans le domaine académique pour le domaine des réseaux.

Dans la première partie de ce chapitre, nous présenterons le simulateur NS2 et les différents langages et outils utilisés pour la simulation. Dans la deuxième partie nous présenterons l'environnement de simulation, les paramètres de réseau, et les différentes étapes effectuées pour l'implémentation du réseau et des différents protocoles de routages. Dans la troisième partie nous présenterons les résultats de simulation en termes de consommation énergétique avec une discussion de ces résultats.

## 2. ENVIRONNEMENT DE SIMULATION

### 2.1. LES SIMULATEURS DES RCSF

Une simulation est le fait d'imiter sur un support informatique un phénomène physique, ce qui est possible à l'aide d'un simulateur. Un simulateur est un dispositif technique permettant de reproduire de façon virtuelle une situation. Il présente donc sous des conditions contrôlables et observables l'évolution du modèle du phénomène [53].

Un simulateur de réseau est un logiciel qui permet de construire un réseau virtuel de façon interactive sur écran puis d'observer son fonctionnement. Ce type de simulateur a été conçu pour rendre observable les principaux concepts associés aux réseaux : topologie, adressage physique et logique, domaines de collision et de diffusion, routage, filtrage, etc. [54].

Parmi les simulateurs qui implémentent des techniques d'accès propres aux réseaux de capteurs sans-fils, les trois simulateurs suivants sont les plus utilisés : le simulateur Avrora, le simulateur OMNET++, et le simulateur NS2.

### 2.2. LE SIMULATEUR NS2 (NETWORK SIMULATOR 2)

NS2 est un simulateur à événements discrets destiné à la recherche. Il est développé en collaboration avec plusieurs entreprises et centres de recherches, pour en citer quelques-uns : LBNL (Lawrence Berkeley National Laboratory), Xerox PARC (Palo Alto Research Center), UCB (Union chimique belge), et USC/ISI (University of SouthernCalifornia / Information

Sciences Institute), dans le cadre du projet VINT13 (qui étudie l'interaction entre différents protocoles), depuis 1995.

Aujourd'hui, NS2 est sûrement le simulateur de réseau le plus utilisé et est devenu avec le temps une référence dans ce domaine. Son avantage réside aussi dans le fait qu'il est multi-plateformes (UNIX et Windows, avec l'émulateur Cygwin qui permet d'émuler un environnement Linux sur Windows « Linux-like environment ») et que son utilisation soit gratuite. NS2 intègre un grand nombre de fonctionnalités pour l'étude des algorithmes de routage multipoint ou unipoint, des protocoles de transport, de session, et d'application (HTTP par exemple). Il gère aussi très bien la couche physique (couche 1) du modèle OSI avec différents systèmes de transmission, filaires ou non. Il reste cependant uniquement adapté aux petits réseaux.

NS2 est entièrement développé en C++ et son utilisation requiert une bonne maîtrise du langage TCL (Tool Command Language) qui est utilisé comme un interpréteur applicatif au langage C du simulateur. En effet, NS2 ne possède aucune interface graphique : toutes les simulations sont réalisées en ligne de commande. Toutefois, elle existe une extension NAM (Network Animator), qui permet de visualiser les résultats d'une simulation une fois achevée [55].

Une fois la simulation terminée, NS produit deux fichiers de traces qui visualisent la dynamique des systèmes simulés, et qui peuvent être interprétés en utilisant les outils : NAM (Network AniMator) et Xgraph.

Plusieurs versions de NS ont été élaborées, la plus récente est la version 3 qui ne supporte que du code en C++ ; néanmoins, la version 2 est la plus utilisée, jusqu'à présent, dans les recherches académiques.

#### ➤ **L'outil de visualisation NAM**

NS2 ne permet pas de visualiser le résultat des expérimentations. Il permet uniquement de stocker une trace de la simulation, de sorte qu'elle puisse être exploitée par un autre logiciel, comme NAM (Network AniMator).

NAM est un outil de visualisation qui présente deux intérêts principaux : représenter la topologie d'un réseau décrit avec NS2, et afficher temporellement les résultats d'une trace d'exécution NS2. Par exemple, il est capable de représenter des paquets TCP ou UDP, la séparation d'un lien entre nœuds, ou encore de représenter les paquets rejetés d'une file d'attente pleine. Ce logiciel est souvent appelé directement depuis les scripts TCL pour NS2, pour visualiser directement le résultat de la simulation [56].

➤ **Outils d'analyse « Awk, Xgraph »**

Le fichier de trace généré par NS2 est un fichier qui contient toutes les informations sur le réseau (temps, énergie, nombre de paquets, identifiants des nœuds, etc.) avec une extension (.tr). A partir de ce fichier et en utilisant le langage AWK on peut obtenir des graphes explicatifs sur l'information désirée.

Une fois le fichier de trace (.tr) est obtenu, on a besoin d'extraire les informations de ce fichier. Pour cela on utilise un programme écrit en langage AWK (Aho Weinberger Kernighan). Le langage AWK est un langage de traitement de lignes, disponible sur la plupart des systèmes Unix et sous Windows avec Cygwin. Il est principalement utilisé pour la manipulation de fichiers textuels pour des opérations de recherches, de remplacement et de transformations complexes. Il agit comme un filtre programmable prenant une série de lignes en entrée (sous forme de fichiers ou directement via l'entrée standard) et écrivant sur la sortie standard, qui peut être redirigée vers un autre fichier ou programme. Un programme AWK est composé de trois blocs distincts utilisables ou non pour le traitement d'un fichier (prétraitement, traitement, posttraitement). AWK lit sur l'entrée ligne par ligne, puis sélectionne (ou non) les lignes à traiter.

Le langage AWK permet d'extraire plusieurs informations à partir de fichier de trace comme par exemple le temps, le débit, le délai, l'énergie, le nombre de paquets reçus, des statistiques sur le nombre de paquets envoyés, etc.

La syntaxe de la commande AWK est :

```
awk [options] [programme] [fichier de sortie]>> [fichier généré]
```

Avec :

- options :
- -F séparateur : permet de modifier le séparateur de champs.
- -f fichier : lit le programme à partir d'un fichier.
- -v awkVar=\$shellVar : Permet de facilement intégrer des variables du shell dans le code Awk.

Dans notre cas c'est l'option -f (le code AWK est programmé dans un fichier), avec :

- Programme : c'est le programme AWK contenu dans un fichier .awk
- Fichier de sortie : c'est le fichier .tr généré par NS2.
- Fichier généré : c'est le fichier .xgr généré par la commande. [57]

Voici un exemple d'une commande AWK :

```
Awk -f energie_noeds.awk reseaux_final.tr >>graphenergie.xgr
```

Pour obtenir les graphes, à partir du fichier (.xgr) généré par la commande précédente, on a besoin de l'outil Xgraph qu'est une application X-Windows qui inclut :

- Un tracé interactif et graphique.
- L'animation.
- Une portabilité de corrections de bugs [58].

Il suffit d'exécuter la commande suivante, pour obtenir les graphes :

```
Xgraph [nom du fichier]
```

Par exemple :

```
Xgraph graphenergie.xgr
```

### 2.3. L'EXTENSION MANNASIM

Mannasim est un environnement de simulation de RCSF basé sur le simulateur de réseau NS2. Mannasim étend NS2 en introduisant de nouveaux modules pour la conception, le développement et l'analyse des différentes applications de RCSF.

L'objectif de Mannasim est de développer un cadre de simulation détaillée, qui peut modéliser avec précision les différents nœuds capteurs et applications tout en offrant un environnement de vérification pour les algorithmes et protocoles [59].

L'installation de l'outil Mannasim avec le simulateur NS2 permet d'offrir un environnement de simulation des RCSF.

### 2.4. CARACTERISTIQUES DE L'ENVIRONNEMENT DE SIMULATION UTILISE

Pour effectuer nos simulations, nous avons utilisé une machine qui a les caractéristiques techniques suivantes :

- Système d'exploitation : Windows 7 Édition Intégrale 64 bits. avec Service pack 1.
- Processeur : intel(R) Core(TM) i5 cpu @ 2.53GHz 2.53 GHz.
- Mémoire (RAM) : 4 Go.
- Le Simulateur NS version 2.35 avec le package Mannasim pour NS-2.35 et avec :
  - L'outil nam version 1.15
  - L'outil xgraph version 12.1

### 3. IMPLEMENTATION

#### 3.1. MODELISATION DU SYSTEME

Le tableau suivant résume l'ensemble des paramètres utilisés pour les simulations :

Paramètre	Valeur
<b>La surface du réseau</b>	100*100 m <sup>2</sup>
<b>Mac utilisé</b>	Mac/802_11
<b>Le nombre de nœuds</b>	100
<b>Le nombre de BS</b>	1
<b>La localisation de la BS</b>	(X=0 , Y=0, Z=0)
<b>L'énergie initiale des nœuds</b>	10 Joules
<b>L'énergie initiale de la BS</b>	100 Joules
<b>Taille du paquet de données</b>	50 Bytes
<b>Le temps de simulation</b>	100 secondes
<b>Les types de capteurs</b>	Mica 2
<b>La mobilité des nœuds</b>	Nœuds non mobiles

Tableau 4.1 : Les paramètres de simulation.

#### 3.2. IMPLEMENTATION

Après avoir préparé notre environnement d'implémentation, on entame l'étape d'implémentation qui consiste aux étapes suivantes :

- Implémentation de la structure et de la dynamique du réseau,
- Implémentation du protocole « Distance Based Cluster Formation Technique For Leach »,
- Et implémentation de notre protocole proposé « DistModLeach ».

##### ➤ Implémentation de la structure et de la dynamique du réseau

Pour implémenter la structure et la dynamique de notre réseau, nous avons écrit un programme en langage TCL qui est réparti sur 2 fichiers. Un pour décrire la structure de réseau et l'autre pour décrire son fonctionnement.

Voici un morceau de code qui décrit la structure de réseau :

```

set node_(0) [$ns_ node]
$node_(0) set X_ 1.0
$node_(0) set Y_ 1.0
$node_(0) set Z_ 0.0
$node_(0) color green
$ns_ at 0.0 "$node_(0) color green"
$ns_ at 0.0 "$node_(0) label SB_(0)"
$ns_ initial_node_pos $node_(0) 60
for {set i 1} {$i< $val(nn) } {inc i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0 ;
    $node_($i) color blue
    $ns_ at 0.0 "$node_($i) color blue"
    $ns_ at 0.0 "$node_($i) label Node_($i)"
}

```

Et voici un morceau de code qui décrit la configuration de la dynamique du réseau :

```

# Configuration des nœuds
$ns_ node-config -sensorNode ON \
-ahocRouting $val(rp) \    (LEACH)
-llType $val(ll) \    (LL)
-macType $val(mac) \    (Mac/802_11)
-ifqType $val(ifq) \    (Queue/DropTail)
-ifqLen $val(ifqlen) \    (200)
-antType $val(antenna) \    (Antenna/OmniAntenna)
-propType $val(prop) \    (Propagation/TwoRayGround)
-energyModel $val(en) \    (EnergyModel/Battery)
-phyType $val(netif) \    (Phy/WirelessPhy)
-channel $chan_1_ \    (Channel/WirelessChannel)
-topoInstance $topo \    (load_flatgrid $val(x) $val(y) )
-agentTrace $trace(agent) \    (ON)
-routerTrace $trace(router) \    (ON)

```

```

-macTrace $trace(mac) \      (ON)
-rxPower 0.5 \
-txPower 0.5 \
-initialEnergy $iniEnergy \
  -movementTrace $trace(movement) (ON)

```

Dans cette implémentation, on choisit le protocole LEACH comme protocole de routage. Le code du protocole LEACH est écrit en langage C++ et il est offert par l'outil Mannasim. Voici un morceau de code de LEACH contenu dans le fichier leachApp.cc :

```

double LeachApp::calculatePi()
{
    /*
     *  $P_i(t) = k / (N - k \bmod(r, N/k))$ 
     * where k is the expected number of clusters per
    round
     * N is the total number of sensor nodes in the
    network
     * and r is the number of rounds that have already
    passed.
     */
    register int n = config_.numberNodes_;
    register int k = config_.desiredClusters_;
    double thresh;
    if (hasBeenClusterHead())
        thresh = 0;
    else if (n - k * round_ < 1)
        thresh = 1;
    else
        thresh = (double) k / (n - k * round_);
    return thresh;
}

```

### ➤ Implémentation du protocole « Distance Based Cluster Formation Technique For Leach »

Pour implémenter le protocole « Distance Based Cluster Formation Technique For Leach », nous avons modifié dans le code de LEACH offert par Mannasim. Nous avons modifié dans le contenu des fichiers : rcagent.cc, rcagent.h et leachApp.cc

Voici un morceau de code de la modification effectuée sur le fichier rcagent.cc :

```
void RCAgent::recv(Packet* p, Handler*)
{
    hdr_cmn *hdr = HDR_CMN(p);
    hdr_rca *rca_hdr = HDR_RCA(p);
    if (app_ && (rca_hdr->rca_link_dst() < 0 || rca_hdr->rca_link_dst() == ((LeachApp *) app_)->sensor_node()->nodeid()))
    {
        //      printf("Receiving:  Link_dst    =  %x,    Type=%d
data_size=%d\n\tMeta    =  %s,    source    =  %d\n", rca_hdr->rca_link_dst(), rca_hdr->msg_type(), hdr->size(), rca_hdr->meta(), rca_hdr->rca_src());
        //  fflush(stdout);
        packetMsg_ = rca_hdr->msg_type();
        distEst_ = disttotal_ + rca_hdr->dist_est();
        ((LeachApp *) app_)->recv(packetMsg_, distEst_, rca_hdr->rca_link_dst(), hdr->size(), rca_hdr->meta(),
            rca_hdr->meta_size(),
            rca_hdr->rca_mac_src(), rca_hdr->rca_link_src());
    }
}
```

### ➤ Implémentation du protocole « DistModLeach »

Pour implémenter notre protocole « DistModLeach », nous avons remplacé la procédure de réélection de CHs dans « Distance Based Cluster Formation Technique For Leach », par la procédure de réélection de CHs proposée dans le protocole « MODLEACH ». Pour cela, nous avons modifié dans l'implémentation précédente de « Distance Based Cluster Formation Technique For Leach ». Nous avons modifié dans le contenu du fichier : leachApp.cc.

Voici un morceau de code de la modification effectuée sur leachApp.cc :

```
double LeachApp::calculatePi()
{
    /*
     *  $P_i(t) = k / (N - k \bmod(r, N/k))$ 
     * where k is the expected number of clusters per
    round
     * N is the total number of sensor nodes in the
    network
     * and r is the number of rounds that have already
    passed.
     */
    double thresh;
    if (hasBeenClusterHead())
    {
        if (((Battery *) sensor_node_ -> energy_model()) -
        > energy()) > 3)
            thresh = 1;
        else
            thresh = 0;
    }
    else
    {
        register int n = config_.numberNodes_;
        register int k = config_.desiredClusters_;
        if (n - k * round_ < 1)
            thresh = 1;
        else
            thresh = (double) k / (n - k * round_);
    }
    return thresh;
}
```

## 4. SIMULATION ET ANALYSE DES PERFORMANCES

### 4.1. METRIQUES D'EVALUATION

Pour pouvoir comparer les performances de notre protocole avec celles de « LEACH » et de « Distance Based Cluster Formation Technique For Leach », nous devons mesurer une métrique importante qui est la consommation énergétique par les nœuds du réseau.

#### ➤ Consommation énergétique

Nous nous sommes intéressés essentiellement à la consommation d'énergie des nœuds puisqu'elle constitue un paramètre essentiel pour la détermination de la durée de vie d'un RCSF. Nous analysons donc l'impact des améliorations effectuées sur le protocole de routage LEACH, et l'impact de la combinaison proposée de ces améliorations, sur l'énergie consommée par le réseau.

Pour faire cela, nous prenons comme critère, l'énergie consommée dans tout le réseau, l'énergie consommée dans la station de base (BS), et l'énergie consommée dans quelques nœuds du réseau (quelques exemples de noeuds) en fonction du temps.

### 4.2. RESULTATS DE SIMULATION

L'exécution de notre code Tcl, en modifiant à chaque fois le protocole de routage, donne les résultats suivants :

#### ➤ Fichier de sortie NAM

La visualisation du réseau est effectuée par l'outil NAM à partir du fichier de sortie (.nam) généré par NS2.

Voici quelques résultats de visualisation de réseau par l'outil NAM :

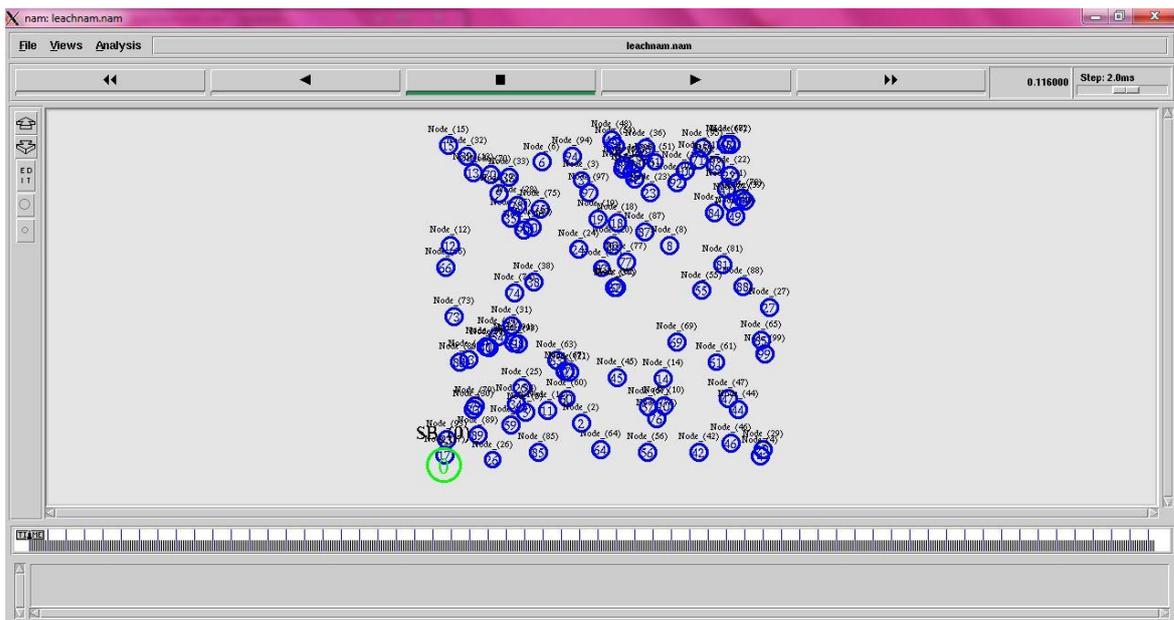


Figure 4.1 : Le réseau à l'instant 0.0 s.

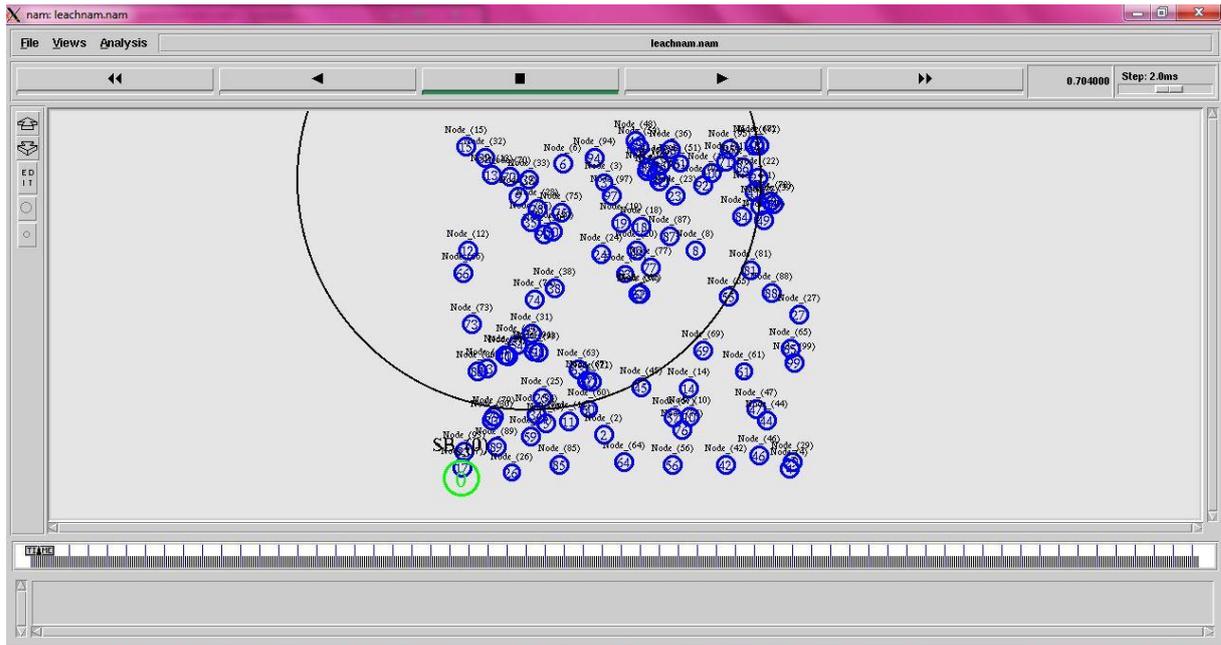


Figure 4.2 : Le réseau à l'instant 6.76 s.

### ➤ Fichier de trace

Pour extraire l'information sur le niveau d'énergie des nœuds du réseau en fonction du temps à partir du fichier de trace (.tr), on utilise des programmes écrits en langage AWK, et on utilise la commande AWK (présentée précédemment) qui permet de sauvegarder le résultat dans un fichier (.xgr).

Voici un morceau du code AWK utilisé pour extraire l'information sur la consommation d'énergie dans tous les nœuds du réseau en fonction du temps :

```
BEGIN {
# le debut du temps de simulation
    start_time = 0
    # le nombre de nœuds
for (j = 0; j < 101; j++) {
energy[j] = 0
}}
{
event = $1
time = $3
i = $9
#    r=receive    d=drop    s=send    f=forward
```

```
for (j = 0; j < 101; j++) {
  if (i == j) {
    if((event == "r" || event == "s" || event == "d" || event
    == "f")) {
      if (j == 0)
        energy[j]= 100 - $17
      else
        energy[j]= 10 - $17
    } }
  }
  energy_total = 0
  for (j = 0; j < 101; j++) {
    energy_total = energy_total + energy[j]
  }
  print(time,energy_total)
}
END {
}
```

En utilisant les programmes AWK précédents, l'outil Xgraph, et les fichiers de trace obtenus par la simulation de chaque protocole de routage, nous avons obtenus les graphes suivants :

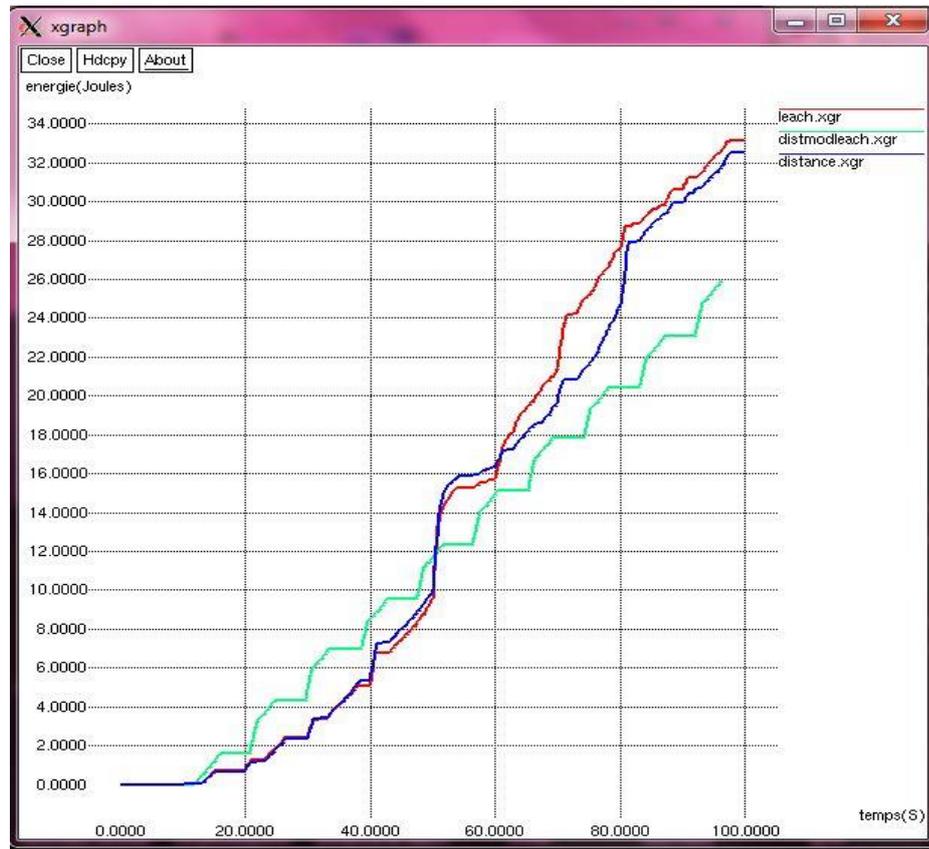


Figure 4.3 : Le niveau d'énergie consommée dans tout le réseau en fonction du temps.

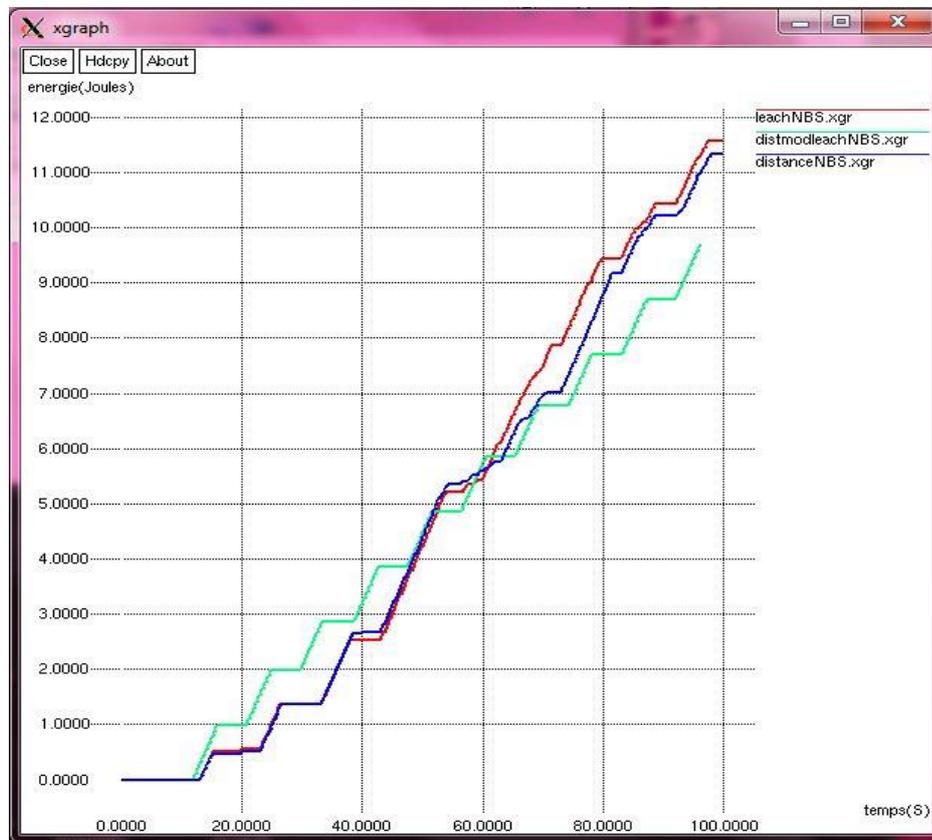


Figure 4.4 : Le niveau d'énergie consommée dans la BS en fonction du temps.

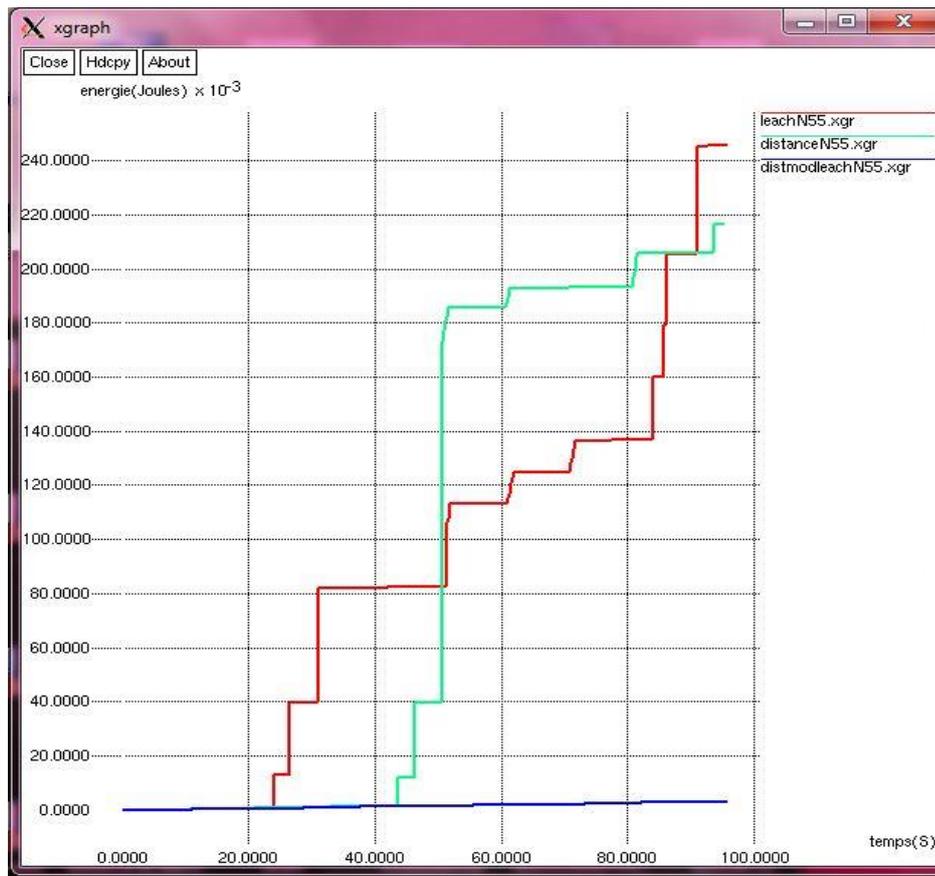


Figure 4.5 : Le niveau d'énergie consommée dans le nœud N°55 en fonction du temps.

Le tableau ci-dessous présente le niveau d'énergie consommée par les nœuds en fin du temps de simulation pour les trois protocoles « LEACH », « Distance Based Cluster Formation Technique For Leach », et « DistModLeach ».

<b>Energie</b> <b>Protocoles</b>	<b>Energie totale</b> <b>consommée par le</b> <b>réseau</b>	<b>Energie totale</b> <b>consommée par la</b> <b>BS</b>	<b>Energie totale</b> <b>consommée par le</b> <b>nœud N°55</b>
<b>LEACH</b>	33.111 (Joules)	11.5581 (Joules)	0.053536 (Joules)
<b>Distance Based</b> <b>Cluster Formation</b> <b>Technique For Leach</b>	32.5293(Joules)	11.3253(Joules)	0.257108(Joules)
<b>DistModLeach</b>	25.9072(Joules)	9.6964(Joules)	0.002983(Joules)

Tableau 4.2. Tableau comparatif d'énergie consommée en fin de simulation.

### 4.3. DISCUSSION

La présentation graphique illustrée dans les figures précédentes, nous permet de remarquer que l'énergie consommée par tout le réseau, l'énergie consommé par un capteur (nous avons pris comme exemple le nœud N° 55), et l'énergie consommée par la station de base, sont totalement différentes dans les protocoles « LEACH », « Distance Based Cluster Formation Technique For Leach », et notre protocole proposé « DistModLeach ».

Les graphes montrent que, après un certain temps de déroulement du réseau ( > 50 secondes) :

- L'utilisation du protocole « Distance Based Cluster Formation Technique For Leach » diminue la consommation d'énergie au niveau de tout le réseau, au niveau d'un nœud individuel, et au niveau de la station de base, par rapport au protocole « LEACH ». Mais les niveaux de la consommation énergétique sont proches entre les deux protocoles.
- L'utilisation de notre protocole proposé « DistModLeach » diminue la consommation d'énergie au niveau de tout le réseau, au niveau d'un nœud individuel, et au niveau de la station de base, par rapport aux deux protocoles « Distance Based Cluster Formation Technique For Leach » et « LEACH ». Et les niveaux de la consommation énergétique entre « DistModLeach » et les deux protocoles précédents ont une différence considérable.

## 5. CONCLUSION

Basés sur les résultats de la simulation, nous avons démontré que notre protocole proposé, qui consiste à combiner deux améliorations du protocole LEACH, améliore la dissipation d'énergie, augmente le gain d'énergie, et prolonge la durée de vie du réseau comparé au protocole « LEACH » et à l'amélioration de LEACH « Distance Based Cluster Formation Technique For Leach ».

### Conclusion générale

Les RCSFs constituent des sujets de recherche innovants pour diverses disciplines des sciences et techniques de l'information et de la communication mais avec toutefois des contraintes spécifiques qui permettent de relever certains défis. Parmi les problèmes posés à l'heure actuelle dans ce type de réseaux est la minimisation de la consommation énergétique des capteurs.

Dans ce projet de fin d'études, nous avons concentré notre étude sur la particularité du routage dans ce type de réseaux imposée par les contraintes sur les ressources énergétiques limités des réseaux de capteurs sans-fil.

Nous avons, présenté les différentes solutions de routage existantes selon différentes classifications. Nous avons focalisé, par la suite, notre étude sur la classe des protocoles de routage hiérarchique, et plus précisément sur le protocole LEACH qui est considéré comme le plus populaire des protocoles de routage hiérarchique.

Le protocole LEACH combine l'efficacité en consommation d'énergie et la qualité de l'accès au média. Cependant, ce protocole présente quelques insuffisances en termes d'efficacité énergétique. Pour cela, plusieurs améliorations du protocole LEACH ont été proposées par les chercheurs, dont l'objectif principal est le prolongement du temps de vie du réseau ainsi que la gestion efficace de la consommation énergétique.

A cet effet, nous avons proposé une amélioration du protocole LEACH que nous avons appelé « DistModLeach » et qui consiste en la combinaison de deux approches existantes, à savoir, l'amélioration « Distance Based Cluster Formation Technique for LEACH Protocol » et une partie de l'amélioration « MODLEACH », dans le but de minimiser la consommation énergétique et d'augmenter la durée de vie du réseau.

Nous avons utilisé le simulateur NS2 avec l'extension Mannasim pour les RCSF, afin d'effectuer les différentes simulations nécessaires et prouver l'efficacité de notre solution par rapport aux autres protocoles. Les résultats fournis par la simulation prouvent que notre protocole offre une meilleure gestion d'énergie par rapport aux deux protocoles « LEACH » et « Distance Based Cluster Formation Technique for LEACH Protocol ».

Suite à ce modeste travail, des perspectives s'offrent à nous, telle que la prise en compte de la deuxième partie de l'amélioration proposée dans « MODLEACH », qui consiste en l'utilisation de différentes puissances des signaux pour chaque type de transmission, dans la conception de notre protocole de routage pour les RCSF.

## Références

- [1] [http://moodle.utc.fr/file.php/498/SupportWeb/co/Module\\_RCSF\\_1.html](http://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_1.html)
- [2] <http://windows.microsoft.com/fr-fr/windows7/what-is-a-sensor>
- [3] W. HEINZELMAN, A. CHANDRAKASAN, and H. BALAKRISHNAN, «Energy-Efficient Communication Protocol for Wireless Micro sensor networks», In Proc. 33rd Hawaii International Conference on System Sciences (HICS '00), Janvier 2000.
- [4] I. F. Akyildiz, Weilian. Su, Y. Sankarasubramaniam, and E. I. Cayirci, « A survey on sensor networks », IEEE Communications Magazine, Vol. 40, No. 8, pp. 1021116, Août 2002.
- [5] Moufida DJAAFAR et Samra BENABDELKADER, « Simulation de la mobilité des nœuds sur une plateforme NS », Mémoire de fin d'études, Centre universitaire de Mila, 2012/2013.
- [6] Bécaye DIOUM, « Effets de la mobilité sur les protocoles de routage dans les réseaux ad hoc », Mémoire de fin d'études, Université MOULOUD MAMMERI de TIZI OUZOU, 2007.
- [7] Saloua CHETTIBI, « Protocole de routage avec prise en compte de la consommation d'énergie pour les réseaux mobiles Ad hoc », Rapport technique, Décembre 2008.
- [8] [https://interstices.info/jcms/c\\_22096/le-modele-de-pair-a-pair-profite-aux-reseaux-de-capteurs-tres-etendus](https://interstices.info/jcms/c_22096/le-modele-de-pair-a-pair-profite-aux-reseaux-de-capteurs-tres-etendus)
- [9] I. Khemapech, I. Duncan, and A. Miller, «A survey of wireless sensor networks technology», In PGNET, Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting, Juin 2005.
- [10] Mohamed AISSANI, « Surveillance de tout point d'une zone d'intérêt à l'aide d'un réseau de capteur multimédia sans fil », Mémoire de magistère, Oued-Smar Alger, Algérie, 2013.
- [11] Noureddine LASLA, « La gestion de clés dans les réseaux de capteurs sans-fil », Mémoire de magister, Institut National de formation en Informatique (I.N.I) Oued-Smar, Alger, 7 Juin 2008.

## Référence

[12] Equipe de Get 2005 Capt'Ad1hoc, « Sensor networks : State of the art », Rapport Technique, Telecom Paris, ENST Br, INT, INRIA, Mars 2006.

[13] [http://moodle.utc.fr/file.php/498/SupportWeb/co/Module\\_RCSF\\_33.html](http://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_33.html)

[14] Imrich Chlamtac, Iacopo Carreraset, and Hagen Woesner, « From internets to bionets: biological kinetic service oriented networks», The case study of Bionetic Sensor Networks, CREATE-NETResearch Consortium, Trento, Italy, 2005.

[15] Omar BESSAD, Nadia BATTAT, et Feriel MIMOUNE, «Le routage dans les réseaux ad hoc (OLSR et AODV)», Mémoire de licence, Université de Bejaia, Juin 2007.

[16] [http://moodle.utc.fr/file.php/498/SupportWeb/co/Module\\_RCSF\\_44.html](http://moodle.utc.fr/file.php/498/SupportWeb/co/Module_RCSF_44.html)

[17] Guy Pujolle, « Les Réseaux » », Livre, 5ème édition, 2006, ISBN: 2121211198719.

[18] B. Bougard, F. Catthoor, D. C. Daly, A. Chandrakasan, and W. Dehaene, « Energy efficiency of the IEEE 802.15.4 standard in dense wireless micro sensor networks: Modeling improvement perspectives », Proceedings of the Conference on Design, Automation and Test in Europe (DATE), 2005.

[19] A. Delye, V. Gauthier, M. Marot, et M. Becker, « Etat de l'art sur les réseaux de capteurs », Rapport de recherche, INT N05001RST GETINT, UMR5157 SAMOVAR, Institut National des Télécommunications, Evry, France, 2005.

[20] <http://fr.slideshare.net/mounayangui/zig-beeuw>

[21] E. H. Callaway, « Wireless sensor networks: Architectures and protocols » Livre, Crcpress edition, 2004.

[22] Julien Vaudour, « Élaboration de couches MAC et réseau pour un réseau de capteurs », Projet d'ingénieur I.I.E, 29 juin 2006.

[23] CHARIF Meryem et BENYAGOUB Amina, « Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil », Mémoire de fin d'études, 2012-2013.

## Référence

- [24] Khaled Bouchakour, «Routage hiérarchique sur les réseaux de capteurs sans fil : Protocole KhLCH (K-hop LayeredClusteringHierarchy) », Mémoire de magister, Ecole Nationale Supérieure de l'Informatique (ESI), 2012.
- [25] M.Guerroumi, N.Badache, and S.Moussaoui, « Data Dissemination and Power Management in Wireless Sensor Networks », Advances in Computing and Communications, Communications in Computer and Information Science, Volume 193, Part T, 593CT37, DOI: 13.1337/978C3CT42C2272TC4bT1, 2011.
- [26] T. Benmansour and S. Moussaoui, « GMAC: Group Mobility Adaptive Clustering Scheme for Mobile Wireless Sensor Networks », In proceeding of the 10 the International Symposium On Programming and Systems ISPS'2011. Alger, Algérie.
- [27] A. Bharathidasan and V. AnadSauPonduru, « Sensor networks: An overview », Rapport technique, Département d'informatique de Californie.
- [28] M. Lehssaini, « Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique», Thèse de Doctorat en informatique, Université de Tlemcen et Université de Franche-Comté, 2009.
- [29] Djallel Eddine Boubiche, «Protocole de routage pour les réseaux de capteurs sans fil», Mémoire de magistère, Université de l'Hadj Lakhdar, Batna, Algérie, 2008.
- [30] Violeta Felea, « Routage dans les réseaux de capteurs sans fil », Journées ResCom Strasbourg, Université de franche.comté, 9-10 Octobre 2008.
- [31] Yasmina Khalfaoui, «Routage dans les réseaux de capteur sans fils», Projet de fin d'étude, Centre universitaire Mustapha Stambouli, Mascara, 2006.
- [32] D. Niculescu, «Topics In Ad-Hoc Networks: Communication paradigms for Sensor Networks», NEC Laboratories America, IEEE Communications Magazine, Mars 2005.
- [33] J.N Al-Karaki and A. E. Kamal, «Routing Techniques in Wireless Sensor Networks: A Survey», Magazine: IEEE Communications, vol. 11, N 6, app.6-28, December 2004.
- [34] W. Bechkit, « Un nouveau protocole de routage avec conservation d'énergie dans les réseaux de capteurs sans fil », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, Juin 2009.

## Référence

- [35] A Amziane et M .Toumi « Plateforme d'évaluation de la tolérance aux pannes des protocoles de routage dans les réseaux de capteurs sans fil », Mémoire d'ingénieur, Ecole nationale Supérieure d'Informatique ESI, 2011.
- [36] Luis Javier García VILLALBA, Ana Lucila Sandoval OROZCO, Alicia Triviño CABRERA, and Cláudia Jacy Barenco ABBAS, «Routing Protocols in Wireless Sensor Networks», Sensors 2339, 9, 8399C8421, ISSN 1424C8223, 2009.
- [37] S. LINDSEY and C. RAGHAVENDRA, « PEGASIS: Power-Efficient Gathering in Sensor Information Systems », IEEE Aerospace Conference Proceedings, Vol. 3, 9-16 pp. 1125-1130, 2002.
- [38] O. YOUNIS and S. FAHMY, « HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks », IEEE Transactions on Mobil Computing, vol. 3, no. 4, pp.3TTC379, 2334, 2004.
- [39] Yong WANG and Byrav RAMAMURTHY, « Layered Clustering Communication Protocol for Wireless Sensor Networks », Proceedings of 16<sup>th</sup> International Conference on Computer Communications and Networks, 844–849, 2007.
- [40] Moustafa A. YOUSSEF, Adel YOUSSEF, and Mohamed F. YOUNIS, « Overlapping Multihop Clustering for Wireless Sensor Networks », IEEE Transactions On Parallel And Distributed Systems, Vol. 23, No. 12, December 2009.
- [41] Qi YANG, Yuxiang ZHUANG, and Hui LI, « An Multi-hop Cluster Based Routing Protocol for Wireless Sensor Networks », Journal of Convergence Information Technology, Volume T, Number 3, Mars 2011.
- [42] Lyes Khelladi et Nadjib Badache « Les réseaux de capteurs : état de l'art », Rapport de recherche, Algérie, Février 2004.
- [43] <http://www.mdpi.com/1424-8220/12/8/11113/htm>
- [44] Yasser Romdhane, « Evaluation des performances des protocoles S-MAC et Directed Diffusion dans les réseaux de capteurs », Projet de fin d'études, Ecole Supérieure des Communcations de Tunis (Sup'Com), 2006 / 2007.
- [45] <http://www.mdpi.com/2224-2708/2/2/235>

## Référence

- [46] Wendi Beth Heinzelman, « Application-Specific Protocol Architectures for Wireless Network », IEEE Transactions on Wireless Communications, Massachusetts Institute of Technology, June 2000.
- [47] M. Malik and Y. Singh, « Analysis of LEACH Protocol in Wireless Sensor Networks », International Journal of Advanced Research in Computer Science and Software Engineering, February 2013.
- [48] Xinhua Wu and Wang Sheng, « Performance comparison of LEACH and LEACH-C protocols by NS-2 », Proceedings of 9<sup>th</sup> International Symposium on Distributed Computing and Applications to Business, Engineering and Science, pp. 254-258, 2010.
- [49] Akramul Azim and Mohammad Mahfuzul Islam, « A Relay Node Based Hybrid Low Energy Adaptive Clustering Hierarchy for Wireless Sensor Networks », International Journal of Energy, Information and Communications, Vol. 3, No. 3, pp. 41-54, 2012.
- [50] Wairagu G. Richard, « Extending LEACH routing algorithm for Wireless Sensor Network », Data Communications Engineering, 2009.
- [51] Snehal Kole, Mr. K.N.Vhatkar, and Mr. V.V.Bag « International Journal of Application or Innovation in Engineering & Management (IJAIEEM) », Volume 3, Issue 3, March 2014, ISSN 2319 – 4847
- [52] D.Mahmood, N.Javaid, S.Mahmood, S.Qureshi, A.M.Memon, and T.Zaman, « Institute of Information Technology, Islamabad, Pakistan ».VOLUME: 04, ISSUE: 04, DECEMBER 2013.
- [53] <http://archives.reseaucerta.org/outils/simulateur/>
- [54] R. E. SHANNON, «Introduction to the art and science of simulation», Proceedings Of the 33th conference on Winter simulation (WSC'98), 1989.
- [55] M.Jamont, « Etude des simulations de routage pour les réseaux sans fil», Étude réalisée dans le cadre du Projet Personnel et Professionnel, Juin 2008
- [56] <http://y-baddi.developpez.com/tutoriels/ns2/>
- [57] <http://www.shellunix.com/awk.html>
- [58] <http://www.nsnam.com/2011/08/xgraph.html>
- [59] <http://www.mannasim.dcc.ufmg.br/>