

MA 004-497-1

République Algérienne Démocratique et Populaire
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE



Université Saad Dahleb Blida

Faculté des Sciences

DÉPARTEMENT D'INFORMATIQUE

Projet de fin d'étude pour l'obtention du diplôme master 2

OPTION

Génie des Systèmes Informatique



Clustering pour la prolongation de la
durée de vie d'un réseau de capteurs
sans fil

Organisme d'accueil : CDTA

Le président du jury : Mr Kawech A.H

Présenter par :
BOUNSSAIRI Adel
RAHMANI Khaled

Encadreur :
Mlle MERABTINE Nassima
Promoteur :
Mr BENYAHIA MOHAMED

Promotion : 2017

MA-004-497-1

Remerciements

C'est avec un grand plaisir que nous réservons ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Nous tenons tout d'abord à remercier *ALLAH* qui nous a aidé et nous a donné la patience et le courage durant ces longues années d'études.

Nous tenons à exprimer notre gratitude et nos remerciements les plus sincères à Mlle MERABTINE Nassima d'avoir accepté de nous accueillir au sein de l'équipe "Antennes-capturs" au Centre de Développement des Technologies Avancées. Nous la remercions pour l'excellent encadrement ainsi que pour la confiance et la grande autonomie qu'elle nous a accordée tout en étant toujours disponible malgré ses nombreuses contraintes.

Nous tenons également à exprimer notre gratitude à Mr. BENYAHYA Mohammed pour son aide précieuse et ses conseils.

Nos remerciements vont également à nos familles pour leurs prières, leur soutien et leurs mots d'encouragement, depuis le début de cette année, mais également pendant tout notre cursus universitaire.

Dédicace

Je dédie ce modeste travail :

A mes très chers parent

*Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération
pour les sacrifices que vous avez consenti pour mon instruction et mon bien être*

A ma grand-mère que Dieu le Tout Puissant la garde en bon santé

A mes tantes pour leur soutien tout au long de cette année

A mes frères Imad eddin, Sidali, Issam et Abdelghafar

et ma sœur adorée Ikram

*A mes frères et sœurs avec lesquels j'ai partagé le bon et le mal Ahmed, Abdelillah, Fateh, Ali,
Abdelhak et Sidali...*

A mon cher binôme, ami et surtout frère Khaled

Et toute personne que je connais et qui m'est chère.

Adel

Dédicace

Je dédie mon mémoire :

A mon père Djelali et ma mère Safia vous êtes pour moi une source de vie car sans vos sacrifices, votre tendresse et votre affection je ne pourrais arriver jusqu'au bout. Je me réjouis de cet amour filial. que dieu vous garde et vous bénisse

A mes frères : Mohamed, Lotfi

A mes sœurs : Soumia, Moufida, Nabila, Dalale

A mes cousins et amis : Mounir, Mohamed, Seddik, Arafet, Yazid, Ayoub et Ali

À tous les étudiants de la promotion 2016/2017 Option : GSI

Je le dédie également à mon cher binôme et ami Adel qui a partagé avec moi le meilleur et le pire pour mener à terme ce modeste travail.

Khaled

Résumé

Les réseaux de capteurs sans fil (*RCSFs*) représentent une technologie émergente qui vise à améliorer la vie des êtres humains. Leur utilisation ne devrait cesser d'augmenter et ceci dans de nombreux domaines qu'ils soient scientifiques, industriels, militaires ou encore environnementaux. Cependant, la taille miniature des micro-capteurs constitue une limitation importante, principalement en terme d'autonomie d'énergie et donc de durée de vie. C'est pourquoi de nombreux travaux portent aujourd'hui sur la gestion de l'énergie consommée par les micro-capteurs. La clusterisation du réseau se révèle comme une bonne technique, qui permet d'optimiser la consommation énergétique, surtout dans les réseaux de capteurs déployés à grande échelle.

Notre projet de fin d'étude consiste à développer un nouveau protocole de clustering pour les réseaux de capteurs sans fil. L'objectif principal de notre travail est l'optimisation de la consommation énergétique afin de prolonger la durée de vie du réseau. Pour réaliser ce projet, nous avons commencé par une étude approfondie des protocoles de clustering. Cette étude nous a permis de concevoir un nouveau protocole nommé "2OSC" pour "Optimized One-Step Clustering" qui tente de remédier aux inconvénients recensés au niveau des protocoles de clustering existants. Les résultats de comparaison par simulation ont montré que notre protocole 2OSC surpasse les protocoles *LEACH-C*, *TPSO-CR* et *ETPSO-CR* en termes d'efficacité énergétique et longévité du réseau et ce pour les réseaux homogènes et hétérogènes.

Mot clés : Réseaux de capteurs sans fil, clustering, *PSO*, passage à l'échelle, efficacité énergétique, durée de vie, *MATLAB*.

Abstract

The Wireless Sensor Network (*WSN*) is an emerging technology that aims to deliver innovative capabilities. It can be used in many fields, scientific, logistic, and military or health. However, the sensors size constitutes an important limitation, mainly in terms of energy autonomy and therefore lifetime because of how smaller the battery is. This is why many studies are focused on the management of the energy consumed by micro-sensors. The clustering of the network is a good technique, which allows to optimize the energy consumption, especially in the networks of sensors deployed on a large scale.

Our final project is to develop a new clustering protocol for wireless sensor networks, based on the centralized clustering process. The main objective of our work is to optimize energy consumption in order to extend the network lifetime, In order to reach the final project, we began with a thorough study of the hierarchical routing and centralized clustering protocols. This study allowed us to design a new protocol named "2OSC" "Optimized One-Step Clustering" that solve the disadvantages identified in the existing protocols. The simulation comparison results showed that our 2OSC protocol surpasses the *LEACH-C*, *TPSO-CR* and *ETPSO-CR* protocols in terms of energy efficiency and network longevity for homogeneous and heterogeneous networks.

Keywords : wireless sensor networks, clustering, *PSO*, Scaling, energy efficiency, lifetime, *MATLAB*.

TABLE DES MATIÈRES

| | |
|---|-----------|
| Introduction générale | 1 |
| I Etat de L'art | 3 |
| 1 Généralités sur les réseaux de capteurs sans fil | 4 |
| 1.1 Introduction | 5 |
| 1.2 Micro-capteur | 5 |
| 1.2.1 Présentation | 5 |
| 1.2.2 Architecture d'un micro-capteur | 6 |
| 1.2.3 Caractéristiques d'un micro-capteur | 8 |
| 1.3 Réseaux de capteurs sans fil | 9 |
| 1.3.1 Définition | 9 |
| 1.3.2 Architecture de communication | 9 |
| 1.3.3 Pile protocolaire | 11 |
| 1.3.4 Paradigme de communication | 12 |
| 1.4 Facteurs et contraintes de conception d'un réseau de capteur sans fil | 13 |
| 1.4.1 Facteurs Matériels | 13 |
| 1.4.2 Facteurs conceptuelles | 14 |
| 1.5 Domaines d'application | 15 |
| 1.6 Conclusion | 18 |
| 2 Algorithmes de clusterisation pour les RCSFs | 19 |
| 2.1 Introduction | 20 |

| | | |
|-------|--|----|
| 2.2 | Concepts de base de clustering | 20 |
| 2.2.1 | Définition | 20 |
| 2.2.2 | Critères de clustérisation | 21 |
| 2.2.3 | Objectifs de clustering | 22 |
| 2.2.4 | Défis de clustering | 23 |
| 2.3 | Protocoles de Clustering dans les RCSFs | 24 |
| 2.3.1 | Low Energy Adaptive Clustering Hierarchy (LEACH) | 24 |
| 2.3.2 | Centralized Low Energy Adaptive Clustering Hierarchy (LEACH-C) | 27 |
| 2.3.3 | Dynamic round time based fixed LEACH (DRTBFL) | 28 |
| 2.3.4 | Enhanced Adaptive Re-clustering Protocol (ADRP) | 31 |
| 2.3.5 | Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network (TPSO-CR) | 33 |
| 2.3.6 | Enhanced Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network (ETPSO-CR) | 36 |
| 2.4 | Comparaison | 37 |
| 2.5 | Conclusion | 37 |

II Propositions et évaluations 38

3 Proposition et conception d'un protocole de clustering 39

| | | |
|-------|---|----|
| 3.1 | Introduction | 40 |
| 3.2 | Hypothèses | 40 |
| 3.3 | Modèle de consommation d'énergie | 41 |
| 3.4 | Description de la solution proposée | 42 |
| 3.4.1 | Modélisation du problème de clustering dans les RCSFs | 42 |
| 3.4.2 | Particle swarm optimization (PSO) | 44 |
| 3.4.3 | Fonctionnement général du protocole proposé | 47 |
| 3.4.4 | Algorithme de clustering centralisé basé sur PSO | 49 |
| 3.5 | Conclusion | 53 |

4 Implémentation et Testes 54

| | | |
|-----|--------------------------------------|----|
| 4.1 | Introduction | 55 |
| 4.2 | Environnement expérimental | 55 |
| 4.3 | Implémentation | 56 |

| | | |
|-------|---|-----------|
| 4.4 | Tests | 57 |
| 4.4.1 | Configuration de scénario de tests. | 57 |
| 4.4.2 | Résultats | 58 |
| 4.5 | Conclusion : | 70 |
| | Conclusion générale | 71 |

TABLE DES FIGURES

| | | |
|-----|---|----|
| 1.1 | Quelques modèles des micro-capteurs | 6 |
| 1.2 | Les composants d'un micro-capteur | 7 |
| 1.3 | Schéma d'un réseau de capteurs sans fil | 9 |
| 1.4 | Architectures de communication de <i>RCSF</i> | 10 |
| 1.5 | Pile Protocolaire | 11 |
| 1.6 | Utilisation des <i>RCSFs</i> dans le domaine militaire | 16 |
| 1.7 | Utilisation des <i>RCSFs</i> dans le domaine médicale | 16 |
| 1.8 | Utilisation des <i>RCSFs</i> dans le domaine environnemental | 17 |
| 1.9 | Utilisation des <i>RCSFs</i> dans le domaine industriel | 18 |
| 2.1 | Architecture de protocole <i>LEACH</i> | 25 |
| 2.2 | Ordonnancement des rounds de protocole <i>LEACH-C</i> | 28 |
| 2.3 | Architecture de protocole <i>ADRP</i> | 31 |
| 2.4 | Fonctionnement de protocole <i>ADRP</i> | 32 |
| 3.1 | L'ordonnancement des rounds | 48 |
| 3.2 | Organigramme du Protocole 2OSC. | 49 |
| 3.3 | Organigramme qui montre l'utilisation de <i>PSO</i> pour la Clusterisation. | 52 |
| 4.1 | Architecture d'implémentation du protocole proposé. | 57 |
| 4.2 | Durée de vie d'un réseau homogène (100 Nœuds) | 60 |
| 4.3 | Durée de vie d'un réseau homogène de 200 Nœuds. | 61 |
| 4.4 | Durée de vie d'un réseau homogène de 300 Nœuds. | 61 |
| 4.5 | Durée de vie d'un réseau hétérogène de 100 Nœuds. | 62 |

| | | |
|------|--|----|
| 4.6 | Durée de vie d'un réseau hétérogène de 200 Nœuds. | 62 |
| 4.7 | Durée de vie d'un réseau hétérogène de 300 Nœuds. | 63 |
| 4.8 | Les nœuds vivants dans un réseau homogène de 100 nœuds. | 63 |
| 4.9 | Les nœuds vivants dans un réseau homogène de 200 nœuds. | 64 |
| 4.10 | Les nœuds vivants dans un réseau homogène de 300 nœuds. | 64 |
| 4.11 | Les nœuds vivants dans un réseau hétérogène de 100 nœuds. | 65 |
| 4.12 | Les nœuds vivants dans un réseau hétérogène de 200 nœuds. | 65 |
| 4.13 | Les nœuds vivants dans un réseau hétérogène de 300 nœuds. | 66 |
| 4.14 | Le taux de paquets reçus au niveau de la <i>SB</i> dans des réseaux homogènes. | 67 |
| 4.15 | Le taux de paquets reçus au niveau de la <i>SB</i> dans des réseaux hétérogènes. | 67 |
| 4.16 | Nombre des <i>CHs</i> dans chaque round | 69 |
| 4.17 | Variation des durées de rounds | 69 |

LISTE DES TABLEAUX

| | | |
|-----|---|----|
| 2.1 | Comparaison entre quelques protocoles de clustering | 37 |
| 4.1 | Paramètres du réseau de capteurs | 57 |
| 4.2 | Paramètre du modèle énergétique | 58 |
| 4.3 | Paramètres de <i>PSO</i> | 58 |
| 4.4 | Tableau d'amélioration dans un réseau homogène | 68 |
| 4.5 | Tableau d'amélioration dans un réseau hétérogène | 68 |

LISTE DES ABRÉVIATIONS

- ADC** : Analog to Digital Converter
- CDMA** : Code Division Multiple Access
- CH** : Cluster-Head
- ETPSO-CR*** : Enhanced Two-tier Particle Swarm Optimization for Clustering and Routing in wireless sensor network
- FND** : First Node Dead
- GEAR** : Geographic and Energy Aware Routing
- GenSeN** : Generator for Sensor Networks
- GPS** : Global Positioning System
- HND** : Half Node Dead
- LEACH*** : Low-Energy Adaptive Clustering Hierarchy
- LEACH-C*** : Centralized LEACH
- LAECH-F*** : Fixed LEACH
- LND** : Last Node Dead
- MAC** : Media Access Protocol
- MATLAB*** : MATrix LABoratory
- NS-3*** : Network Simulator 3
- OSI** : Open Systems Interconnection
- PSO** : Particule Swarm Optimization
- RCSF** : Réseaux de Capteurs Sans Fil
- SB** : Station de base
- TDMA** : Time Division Multiple Access
- TPSO-CR** : Two-tier Particle Swarm Optimization for Clustering and Routing in wireless

sensor network

WSN : Wireless Sensor Network

2OSC : 2 Optimized One-Step Clustering

INTRODUCTION GÉNÉRALE

Les progrès réalisés d'une part dans le domaine de la micro-électronique et d'autre part dans les technologies de communication sans fil ont permis de produire, à coût raisonnable, des micro-capteurs communicants et peu consommateurs en énergie. Dans le besoin d'observer et de contrôler les environnements physiques, ces nœuds capteurs ont l'objectif de récolter des grandeurs telles que la luminosité, le mouvement, la température, la pression barométrique, etc., éventuellement de les traiter, et de les acheminer vers une destination finale appelée station de base (*SB*). Ces nœuds capteurs doivent d'être autonomes, d'une taille miniature et peuvent être déployés d'une manière dense et aléatoire dans le champ surveillé. Les micro-capteurs communiquent entre eux à travers un canal sans fil, ainsi formant un réseau de capteurs sans fil (*RCSF*) ou Wireless Sensor Network (*WSN*). Ce type de réseaux a une très grande variété d'applications dans le domaine environnemental, médical, industriel et militaire.

En raison des contraintes de miniaturisation, les micro-capteurs sont généralement dotés de ressources très limitées en terme de capacité de calcul, d'espace de stockage de données, de débit de transmission et surtout d'énergie. En effet, les nœuds capteurs sont généralement alimentés via des batteries non remplaçables ni rechargeables. Par conséquence, la durée de vie d'un nœud capteur est essentiellement dépendante de la durée de vie de sa batterie. C'est pour cela qu'un usage prudent et efficace de la batterie est primordial afin de prolonger la durée de vie du réseau.

Ces dernières années, les recherches se sont penchées sur la proposition des nouveaux protocoles visant l'optimisation de la consommation énergétique et donc la prolongation de la durée de vie du réseau. Le clustering est l'une des techniques efficaces qui améliorent la durée de vie des réseaux de capteurs sans fil déployés à grande échelle. Il consiste à regrouper les micro-capteurs dans un ensemble de clusters et de désigner dans chacun un nœud particulier qui aura

le rôle du chef de cluster (*CH*). Ce dernier doit collecter les données de tous les membres de son cluster, faire l'agrégation des données et les transmettre vers la *SB*. Le clustering permet d'optimiser la consommation énergétique grâce à la minimisation du nombre de messages circulant dans le réseau (puisque seuls les *CHs* se chargeront de la transmission de données à la *SB*), l'agrégation qui permet de minimiser le nombre des paquets à transmettre, la localisation de la recherche et la maintenance des routes, etc.

Dans le cadre de notre sujet de fin d'étude, nous nous intéressons à la clusterisation d'un *RCSF*. Notre travail consiste en l'exploration de quelques protocoles de clustering proposés spécifiquement pour les *RCSFs*, afin de concevoir un nouveau protocole de clustering qui tentera de remédier aux inconvénients recensés au niveau des protocoles de clustering existants. Ce nouveau protocole doit répondre aux objectifs suivants :

1. Il doit prolonger la durée de vie du réseau au maximum possible.
2. Il doit être scalable, i.e. il doit maintenir ses performances lors du passage à l'échelle.
3. Le protocole doit fonctionner correctement pour les deux types de réseaux : homogène et hétérogène.

Pour ce faire, ce mémoire est organisé en quatre chapitres. Le premier chapitre présente un aperçu général sur les réseaux de capteurs sans fil, notamment sur leur architecture, classifications et leurs domaines d'applications.

Le deuxième chapitre est consacré à l'étude des concepts de base de clustering, sa classification, ses défis, objectifs et avantages. Nous allons par la suite citer quelques algorithmes de clustering proposés dans la littérature, en expliquant leur principe de fonctionnement et en discutant leurs avantages et inconvénients.

Le troisième chapitre résume les étapes de notre conception d'un nouvel algorithme nommé *2OSC* pour "Optimized One-Step Clustering", qui vise à remédier à quelques problèmes de clustering comme la re-clusterisation périodique du réseau et la durée de round statique dont l'objectif est l'optimisation de la consommation énergétique.

Le quatrième chapitre est consacré à l'implémentation de l'algorithme proposé ainsi que sa comparaison, par simulation, avec trois solutions de l'état de l'art. Les métriques de comparaison incluent la durée de vie du réseau, le nombre de nœuds vivants et le taux de délivrance de paquets au niveau de la station de base.

Enfin, la conclusion générale de ce mémoire synthétisera nos principales contributions et donnera quelques perspectives à notre travail.

Première partie

Etat de L'art

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

1.1 Introduction

Les réseaux de capteurs sans fil (*RCSFs*) ont fait l'objet de nombreux travaux de recherche dans ces dernières années grâce aux progrès réalisés dans les domaines de la micro-électronique, la micro-mécanique et des technologies de communication sans fil.

Ce type de réseau est amené à résoudre divers problèmes. Aujourd'hui, on peut facilement envisager un large éventail d'applications des réseaux de capteurs sans fil : la surveillance environnementale, les maisons intelligentes, la sécurité, la santé, etc. Les réseaux de capteurs auront, dans le futur, un impact significatif dans notre vie quotidienne.

Dans ce premier chapitre, nous présentons les réseaux de capteurs sans fil avec un plan méthodologique. Nous commençons par présenter les micro-capteurs sans fil. Nous passerons par la suite à la présentation de leur architecture, caractéristiques et types. Enfin, nous discutons l'architecture et les spécificités des réseaux de capteurs sans fils ainsi que leurs domaines d'application.

1.2 Micro-capteur

1.2.1 Présentation

Un micro-capteur est un appareil autonome d'une taille minuscule, capable d'effectuer trois tâches complémentaires : la mesure d'une valeur physique, le traitement de cette mesure et la communication. Parmi les différents types de mesures enregistrées par les capteurs, nous pouvons citer : la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la position, la pression, la présence d'un gaz, la vision (capture d'image), le son, etc.

Dans les dernières années, l'importance des micro-capteurs a encouragé les grands fournisseurs à ouvrir le marché à l'industrie. Le fournisseur le plus connu est : *Xbow* (aussi appelé *Crossbow*) qui est considéré jusqu'à aujourd'hui une référence dans la fabrication des micro-capteurs, parmi ses fameux micro-capteurs nous trouvons *Mica2* et *MicaZ* (voir figure 1.1).

La technologie des capteurs a beaucoup évolué. Actuellement, il existe une large gamme de type de capteurs. Leur architecture et leur taille changent selon le type d'application au quelle ils sont destinés.

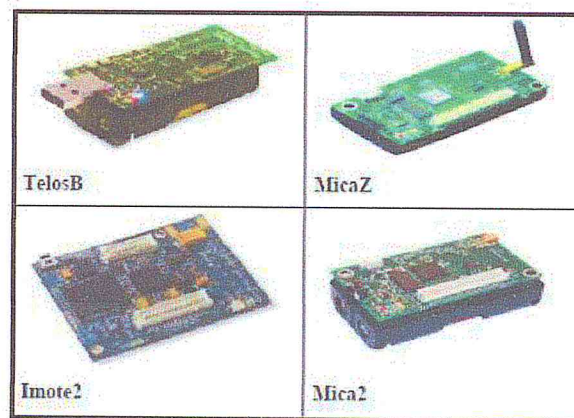


FIGURE 1.1 – Quelques modèles des micro-captureurs

1.2.2 Architecture d'un micro-captureur

Nous distinguons les deux parties qui composent un micro-captureur :

Architecture matérielle

Un capteur est composé de quatre modules de base comme représentée dans la figure 1.2 : un module d'acquisition (captage), un module de traitement, un module de communication et une source d'énergie.

Il existe des micro-captureurs qui sont dotés d'autres composants additionnels (optionnels), dépendant des applications, tels qu'un système de localisation, un générateur d'énergie et un mobilisateur [1], [3].

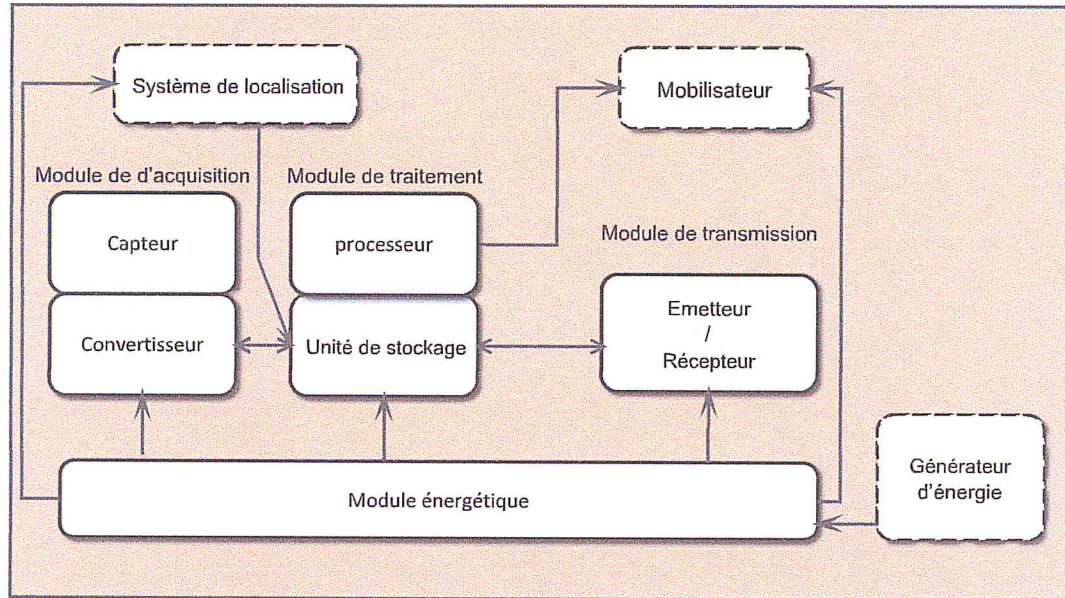


FIGURE 1.2 – Les composants d'un micro-capteur

(a) Module de d'acquisition :

Il se compose de deux sous unités, l'unité de captage et un convertisseur analogique numérique *CAN*. La première unité prélève l'information de l'environnement local, tant dis que la deuxième convertit l'information relevée (les signaux mesurés par le capteur) en signaux numériques, qui sont par la suite traités par le module de traitement.

(b) Module de traitement :

Ce module est généralement composé d'une petite unité de stockage pour stocker les données collectées, et d'un micro-contrôleur qui fonctionne à l'aide d'un système d'exploitation pour traiter les données. Le module de traitement assure aussi le routage de paquets.

(c) Module de transmission :

Il est composé d'un émetteur/récepteur permettant la communication entre les différents nœuds sur un médium sans fil.

(d) Module énergétique :

Il correspond le plus souvent à une batterie ou une pile, dont les ressources limitées deviennent la contrainte fondamentale. L'accès à ces batterie pour changer ou charger est difficile, voir même impossible dans certaines applications [2].

Architecture logicielle

Plusieurs plate-formes logicielles ont été également développées spécifiquement pour les réseaux de capteurs sans fil. En effet, la contrainte énergétique des micro-capteurs exige l'utilisation des systèmes d'exploitation (*OS*) légers. Parmi ces *OS*, nous pouvons mentionner : *TinyOS*, *Contiki* et *LiteOS*. *TinyOS* est l'un des *OS* les plus utilisés et les plus populaires dans le domaine des *RCSFs*.

1.2.3 Caractéristiques d'un micro-capteur

Il existe plusieurs caractéristiques qui spécifient un micro-capteur, parmi eux nous citons :

(a) **Sources d'énergie :**

L'alimentation d'un micro-capteur est assurée par une batterie de capacités limitées. Ceci fait que l'énergie est l'une des problématiques majeures des micro-capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible (hostiles ou inaccessibles par les êtres humains). Il faut donc que les micro-capteurs économisent au maximum l'énergie dans le but de prolonger leur durée de vie [2].

(b) **Taille :**

La taille relativement petite d'un micro-capteur lui permet de se loger dans beaucoup d'endroits avec discrétion. Néanmoins, elle peut poser des problèmes puisque une taille minuscule limite les ressources (la performance du processeur, la capacité mémoire, la capacité de batterie).

(c) **Coût de construction :**

Il est nécessaire d'avoir un faible coût, car dans plusieurs applications, le nombre des micro-capteurs à déployer peut atteindre des milliers.

(d) **Mémoire :**

La capacité de mémoire des micro-capteurs est considérée comme l'une de ses importantes caractéristiques. Elle est composée de deux types de mémoires, à savoir : la mémoire du programme, à partir de laquelle le micro-contrôleur exécute les programmes, et la mémoire des données utilisée pour le stockage des données mesurées. La taille de cette mémoire est généralement limitée.

1.3 Réseaux de capteurs sans fil

1.3.1 Définition

Un réseau de capteurs sans fil (Wireless Sensor Network (*WSN*)) est un réseau informatique. Il est considéré comme un type particulier des réseaux Ad-hoc. Un *RCSF* est composé d'un ensemble de dispositifs très petits, nommés nœuds capteurs ou micro-capteurs, variant de quelques dizaines d'éléments à plusieurs milliers. Un réseau de capteurs peut être mis en place dans le but de surveiller une zone géographique plus ou moins étendue pour détecter l'apparition de phénomènes ou mesurer une grandeur physique (température, pression, vitesse, etc.).

Dans ce réseau, les nœuds sont capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds n'est pas nécessairement prédéterminée. Ils sont soit posés à un endroit précis, soit dispersés aléatoirement à travers un champ de captage, qui définit le terrain d'intérêt pour le phénomène capté. Les données captées sont acheminées vers un point de collecte appelé station de base *SB* ou *sink*) via une communication sans fil grâce à un routage multi-saut, généralement [1]. La station de base peut être connectée à l'utilisateur du réseau via Internet ou un satellite voir figure 1.3.

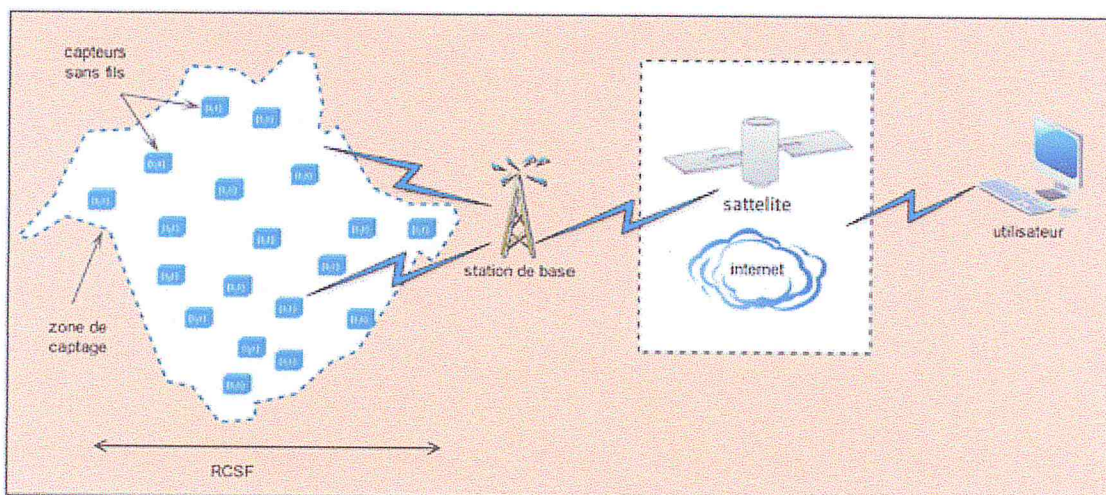


FIGURE 1.3 – Schéma d'un réseau de capteurs sans fil

1.3.2 Architecture de communication

Par architecture de communication, nous désignons la façon dont les nœuds-capteurs transmettent l'information à la station de base. Nous citons ci-après les différentes architectures

existantes, qui sont illustrées dans la figure 1.4 :

(a) **Architecture plate :**

Chaque nœud transmet l'information directement vers la station de base.

(b) **Architecture multi-sauts :**

Chaque nœud transmet l'information à la *SB* à travers d'autres nœud capteur i.e via un routage multi-sauts.

(c) **Architecture Hiérarchique :**

Dans une architecture hiérarchique, les nœuds forment des clusters. Dans chacun, un nœud principal appelé Cluster-Head (*CH*), aura le rôle de collecter les informations des autres nœuds et les transmettre vers la *SB*.

(d) **Architecture Arborescente :**

Cette architecture ressemble à celle présentée ci-dessus avec une seule différence : chaque *CH* transmet l'information à d'autres *CH* jusqu'à arriver à la *SB*, i.e. routage multi-sauts entre les *CH*s.

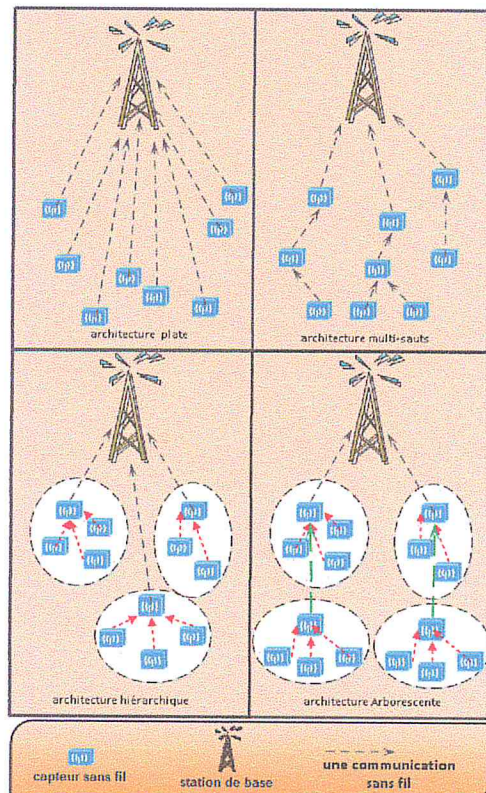


FIGURE 1.4 – Architectures de communication de *RCSF*

1.3.3 Pile protocolaire

Dans le but de bien acheminer les données vers l'utilisateur final (task manager), le procédé de communication suit le fameux modèle *OSI* et comporte cinq couches : application, transport, réseau, liaison de données et physique, ainsi que trois plan de gestion : énergétique, mobilité et tâches.

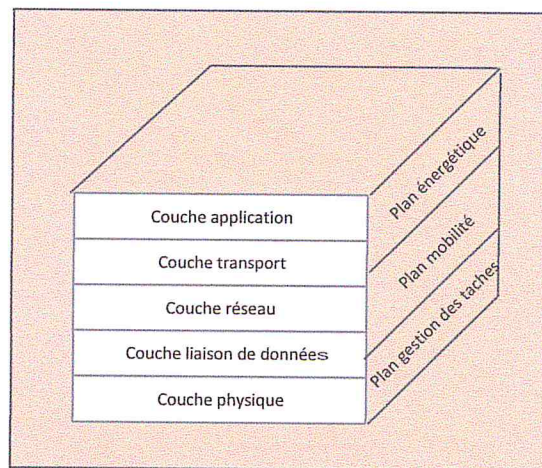


FIGURE 1.5 – Pile Protocolaire

Le but du modèle en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous), et donc elle utilise les services des couches inférieures et en fournit à celle du niveau supérieur.

Ci-dessous nous présentons la fonctionnalité principale de chaque couche :

(a) **Couche application :**

C'est la couche qui permet d'obtenir une connectivité de bout en bout entre un réseau de données et l'utilisateur à l'aide d'un logiciel d'application qui changera d'un micro-capteur à un autre afin d'obtenir un résultat fiable.

(b) **Couche transport :**

Elle s'occupe de la segmentation, transfert et ré-assemblage des données ainsi que le contrôle du flux de données et la gestion des éventuelles erreurs de transmission.

(c) **Couche réseau :**

Le but principal de cette couche est d'assurer le routage des données des micro-capteurs vers la destination souhaitée. Le routage dépend de l'architecture du *RCSF* et du protocole. En général, les protocoles de routage conçus pour les *RCSFs* diffèrent de ceux

proposés pour les réseaux filaires ou avec une infrastructure par les caractéristiques suivantes :

- Les micro-capteurs doivent s'auto-organiser à cause de l'absence d'un système d'adressage global.
- Plusieurs micro-capteurs peuvent produire la même information (redondance), donc le protocole de routage doit prendre cette caractéristique en considération.
- Trouver un chemin redondant pour tolérer les pannes des micro-capteurs.
- Le protocole de routage doit prendre en considération la contrainte énergétique.

(d) **Couche liaison de données**

Contient deux sous couches MAC et Liaison. Cette couche sert à faire le multiplexage des données, contrôler l'accès au média, minimiser le taux de collisions entre les données diffusées par les nœuds voisins, etc.

(e) **Couche physique :**

Elle s'occupe de transmettre l'ensemble des bits cryptés sur un support physique.

En plus de ces cinq couches, la pile protocolaire des réseaux de capteur sans fil comprend trois autres plans de gestion :

(a) **Plan énergétique :**

Par souci d'économie d'énergie, le plan énergétique gère la consommation d'énergie d'un nœud capteur pendant ses opérations principales : captage, calcul et transmission.

(b) **Plan mobilité :**

Il détecte et enregistre la mobilité des nœuds capteurs. En utilisant ces positions, les nœuds capteurs peuvent connaître qui sont leurs voisins. Parfois à cause de la destruction ou l'ajoute de quelques nœuds la nécessité de faire une auto-organisation est assurée par le plan mobilité.

(c) **Plan de gestion des tâches :**

Il planifie les événements tel que le captage dans une zone de couverture spécifique.

1.3.4 Paradigme de communication

Le mode d'envoi des informations par les micro-capteurs au niveau d'un *RCSF* change d'une application à une autre, par rapport au but du réseau. Dans un *RCSF*, nous distinguons les quatre paradigmes de communication suivants :

(a) **Événementiel :**

Les micro-capteurs ne transmettent l'information à la station de base que lorsqu'un événement se produit dans un point de la zone de couverture; par exemple, un changement de température.

(b) **Périodique :**

La majorité des applications de *RCSFs* utilisent ce mode de communication où les micro-capteurs captent et envoient l'information à la station de base dans chaque intervalle du temps.

(c) **Requête/réponse :**

La station de base envoie une requête à un sous ensemble du réseau pour demander une information particulière. Cet ensemble de nœuds répond à cette requête par l'envoi des informations demandées.

(d) **Hybride :**

Quand une application utilise plusieurs paradigmes de communication, décrits plus haut, à la fois.

1.4 Facteurs et contraintes de conception d'un réseau de capteur sans fil

La conception et la mise en place d'un réseau de capteur sans fil sont influencées par plusieurs facteurs, qui sont considérés comme métriques d'indication de performance d'un réseau de capteur sans fil. On peut généraliser ces facteurs en deux grandes familles : matériels et conceptuels.

1.4.1 Facteurs Matériels

Nous pouvons citer les facteurs matériels suivants :

(a) **Taille :**

Un micro-capteur est un matériel de petite taille, caractérisé par des modestes ressources.

(b) **Puissance de calcul, portée et bande passante :**

En plus de l'énergie, les nœuds-capteurs ont aussi une capacité de traitement et de mémoire limitée. Ainsi, bien que dans certaines applications les *RCSFs* peuvent couvrir

une large zone de captage, mais un micro-capteur sans fil ne peut couvrir généralement que de 10 à 75 mètres. En outre, Afin de minimiser l'énergie consommée lors de transfert de données entre les nœuds, les micro-capteurs opèrent à bas débit (quelques dizaines de Kb/s).

1.4.2 Facteurs conceptuelles

Nous pouvons citer les facteurs conceptuelles suivants :

(a) **Tolérance aux pannes :**

Plusieurs problèmes peuvent survenir lors du déploiement des micro-capteurs dans un champ de captage : panne physique ou logicielle, interférence environnementale, etc. La tolérance aux pannes est la capacité de maintenir le fonctionnement d'un réseau de capteur après n'importe quel problème dans le réseau.

(b) **Scalability ou Passage à l'échelle :**

Le nombre de nœuds déployés dans une application peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec une très grande densité de nœuds-capteurs. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter-nodales, et nécessite que la station de base soit équipée de mémoire suffisante pour stocker les informations reçues. Un réseau de capteurs doit être scalable c'est-à-dire, possède la faculté d'accepter un très grand nombre de nœuds qui collaborent ensemble afin d'atteindre un objectif commun.

(c) **Topologie :**

Le changement de la topologie est l'un des aspects originaux des réseaux de capteurs sans fil. En effet, ce changement topologique du réseau peut engendrer des problèmes de connectivité ou, au pire, une défaillance totale du *RCSF*. Ce changement topologique est due à :

- **Mobilité des nœuds :**

les nœuds capteurs peuvent être mobiles (cas des nœuds-capteurs robots) ou peuvent être fixés sur des entités mobiles. Les nœuds capteurs peuvent aussi bouger sous l'effet de facteurs externes tel que le vent.

- **Ajout ou suppression des micro-capteurs :**

A cause de l'autonomie énergétique limitée des nœuds ou même à cause d'une

défaillance, les micro-capteurs peuvent devenir hors service changeant ainsi la topologie du réseau. De même, l'ajout de nouveaux nœuds au réseau peut aussi remettre en cause la topologie du réseau.

(d) **Agrégation de données :**

Dans les réseaux de capteurs, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement. Ceci peut engendrer la réception d'informations redondantes par la station de base. Pour optimiser la consommation d'énergie, il faut réduire la quantité d'informations redondantes transmises par les micro-capteurs.

L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation de données. Avec cette technique, les nœuds intermédiaires agrègent l'information reçue de plusieurs sources.

1.5 Domaines d'application

L'évolution et la variété de types des capteurs (humidité, température, pression, etc.), leur dimension, et l'outil de communication sans fil ont permis aux *RCSFs* d'être une technologie indispensable dans plusieurs domaines.

Ci-dessous nous citons les grands domaines d'application :

(a) **Militaire :**

Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Les *RCSFs* peuvent être déployés dans une zone stratégique afin d'analyser le mouvement de l'ennemi, détecter les zones infectées par des radiations nucléaires, etc (voir figure 1.6).

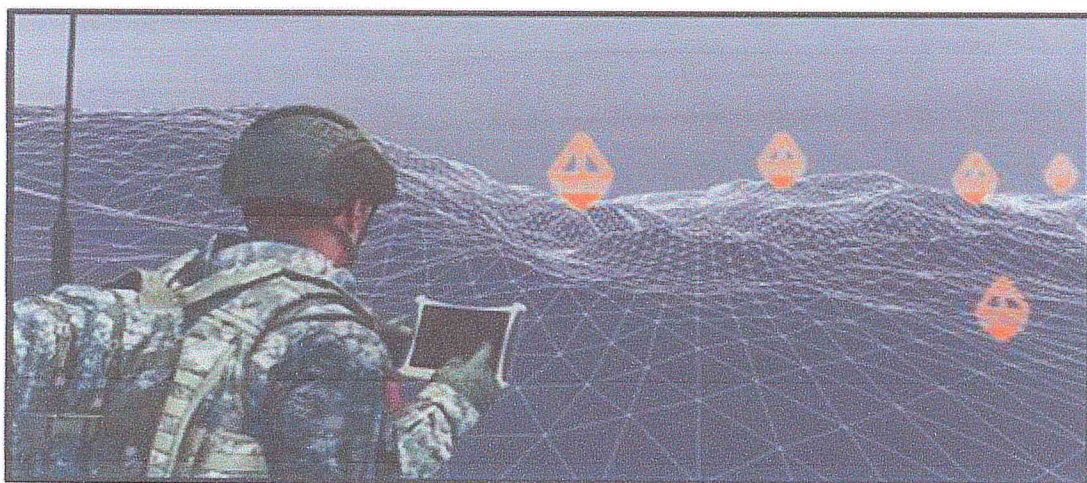


FIGURE 1.6 – Utilisation des *RCSFs* dans le domaine militaire

(b) Médicale :

L'utilisation des réseaux de capteur dans le domaine médical aide à contrôler l'état, en temps réel, des malades cancéreux ou diabétiques (Bio-Surveillance), de suivre le mouvement physiothérapie à l'aide des micro-capteurs qui sont soit avalés ou implantés dans le corps (figure 1.7), de contrôler l'état des personnes invalides (handicaps) en cas de faiblesse de leurs organes, et de faciliter les diagnostics des maladies. Cela permet de maintenir la santé des patients.



FIGURE 1.7 – Utilisation des *RCSFs* dans le domaine médicale

(c) **Environnemental :**

Dans ce domaine, les *RCSFs* sont utilisés pour étudier les phénomènes et détecter des catastrophes naturelles comme illustré dans la figure 1.8, telles que le tremblement de terre, tsunami, etc.

En général, les applications environnementales peuvent être divisées en cinq catégories :

- Application Métrologique.
- Application géologique.
- Surveillance d'énergie.
- Surveillance de pollution.
- Surveillance d'habitat.



FIGURE 1.8 – Utilisation des *RCSFs* dans le domaine environnemental

(d) **Industriel :**

Les *RCSFs* se trouvent largement dans le domaine industriel à cause de leur efficacité d'augmenter le rendement de production. Les applications dans le domaine industriel se divisent en deux catégories :

- L'acquisition des données :

L'objectif de ce genre d'applications est de collecter des informations diverses sur l'environnement tel que la température, l'humidité, etc.

- Supervision :

Ces applications sont très importantes dans le domaine industriel, car elles permettent, par exemple, de superviser l'état d'amortissement d'un équipement en vue de prédire les défauts et d'anticiper leur réparation ou leur remplacement (figure 1.9).

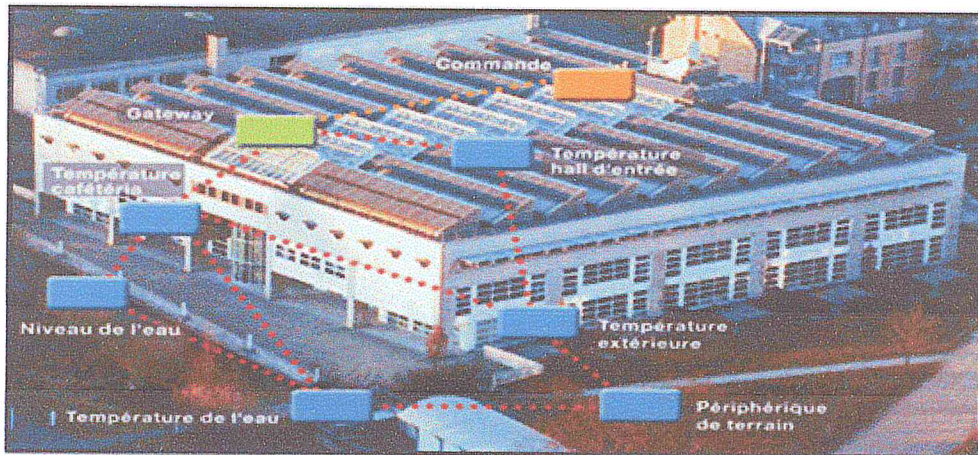


FIGURE 1.9 – Utilisation des *RCSFs* dans le domaine industriel

1.6 Conclusion

Nous avons essayé à travers ce chapitre de présenter les concepts généraux liés aux réseaux de capteurs sans fils, en commençant par la présentation du micro-capteur, son architecture et ses caractéristiques. D'autres points des *RCSFs* ont été présentés aussi tels que : la pile protocolaire, les spécificités des *RCSFs* et leurs domaines d'application.

Les réseaux de capteurs sont une nouvelle technologie, qui permet de faciliter le suivi et le contrôle à distance de l'environnement physique avec une meilleure précision. Néanmoins, les *RCSFs* présentent certaines contraintes fondamentales, principalement, l'énergie limitée des micro-capteurs ce qui limite leur durée de vie.

La clusterisation s'avère comme une technique efficace permettant l'amélioration de la durée de vie des réseaux de capteurs sans fil déployé à grande échelle. Dans cette optique, nous étalons dans le chapitre suivant le concept de clustering ainsi que quelques protocoles de clustering à titre représentatif.

CHAPITRE 2

ALGORITHMES DE CLUSTERISATION POUR LES RCSFS

2.1 Introduction

Comme nous l'avons vu dans le chapitre précédent, les *RCSFs* sont caractérisés par de modestes ressources matérielles, notamment celle énergétique, ce qui constitue leur principal handicap. Ceci est due au fait que les nœuds capteurs sont généralement alimentés via des batteries non remplaçables ni rechargeables. Par conséquent, la durée de vie d'un micro-capteur est essentiellement dépendante de la durée de vie de sa batterie. D'autre part, plusieurs applications des *RCSFs* nécessitent le déploiement d'un grand nombre de micro-capteurs dispersés aléatoirement autour du terrain d'intérêt du phénomène capté. Les contraintes matérielles des *RCSFs* combinées à la densité du déploiement compliquent davantage les tâches ordinaires des *RCSFs*, telles que le routage des données, et rendent la prolongation de la durée de vie du réseau un challenge réel. La clusterisation d'un *RCSF* s'avère comme une technique efficace permettant l'amélioration de la durée de vie des réseaux de capteurs sans fil, surtout ceux déployés à grande échelle.

L'objectif de ce chapitre est la présentation des concepts de base de clustering, ses avantages et challenges ainsi que la description du principe de fonctionnement de quelques protocoles de clustering proposés spécifiquement pour les *RCSFs*.

2.2 Concepts de base de clustering

2.2.1 Définition

Le clustering est une technique qui consiste à partitionner le réseau en un ensemble de groupes appelés clusters [18]. Dans chaque cluster on trouve un micro-capteur principal désigné comme chef de cluster (Cluster-Head), et un ensemble de micro-capteurs (Non-Cluster-Head) qui sont dans la portée de communication de leurs *CHs* (en cas d'utilisation de la communication directe).

Le *CH* a le rôle d'une station de base locale, il reçoit les informations collectées de tous ses membres, fait quelques possibles traitements d'agrégation et envoie les données vers une destination suivante qui est soit la station de base ou un autre *CH* (selon le mode de communication utilisé : directe ou multi-sauts) [17].

Le clustering permet d'optimiser la consommation énergétique menant à une meilleure durée de vie du réseau. Ceci grâce à la minimisation du nombre de messages circulant dans le réseau. En effet, au lieu que chaque nœud envoie sa propre données, seuls les *CHs* se chargeront de la transmission de données à la station de base. De plus, la taille des données peut être de loin

minimisée grâce à l'agrégation de données au niveau des *CHs*. La localisation de la recherche et la maintenance des routes au sein du cluster et la réduction de la taille des tables du routage est un autre avantage. Ainsi, les chefs de clusters peuvent planifier les activités du cluster de sorte que les nœuds passeront en mode de veille la plupart du temps, qui est un autre moyen pour conserver l'énergie.

2.2.2 Critères de clustérisation

Il y a deux grandes étapes dans la clustérisation qui sont la sélection de *CHs* et la formation de clusters.

(a) Les principales préoccupations de la sélection des *CHs* sont :

- **Distance entre le *CH* et la *SB*** : Garantir que la distance entre la *SB* et les *CHs* est courte revient à minimiser l'énergie dissipée par les *CHs*. Ceci est justifié par le fait que l'énergie consommée pour envoyer un message est fortement liée à la distance vers la destination. En effet, la puissance du signal doit être augmentée pour atteindre une destination éloignée, et donc plus d'énergie consommée.
- **Capacité des *CHs*** : Comme nous l'avons dit précédemment, les *CHs* effectuent des tâches supplémentaires par rapport aux autres nœuds, telles que l'agrégation et la transmission de données qui peuvent épuiser leur énergie. Ceci fait de la rotation du rôle du *CH* entre les différents nœuds du réseau (connu sous le terme "re-clustering") une nécessité afin d'éviter la mort prématurée des *CHs*. Conséquemment, le fonctionnement général des protocoles de clustering est divisé en rounds. Au début de chacun, un re-clustering du réseau doit avoir lieu.
- **Distribution des *CHs*** : Afin que les *CHs* ne soient pas concentrés dans une seule région laissant le reste du réseau non-couvert, la position des nœuds est un autre critère du choix des *CHs*. Un autre inconvénient de la concentration des *CHs* dans une seule région est que les nœuds non-*CHs* qui se trouvent loin de la zone de concentration des *CHs* auront besoin de dissiper une grande quantité d'énergie pour communiquer avec leur *CHs* respectifs. De ce fait, un bon algorithme de clustering doit garantir une bonne distribution des *CHs* sur le réseau.
- **Énergie résiduelle** : Les nœuds ayant une grande valeur d'énergie résiduelle sont plus favorisés pour jouer le rôle du *CH*.

- **Durée de round** : C'est le temps nécessaire pour faire le re-clustering du réseau. En effet, bien que le re-clustering est une étape inévitable pour assurer l'équilibrage de charge entre les différents nœuds du réseau ; cependant, il entraîne un gaspillage important de l'énergie (messages échangés pour refaire le clustering). C'est pour cela qu'il faut bien paramétrer la durée de round.
- (b) Les critères considérés pour l'affectation des membres aux clusters sont :
- **Distance entre le nœud et le CH** : La distance entre un nœud et son *CH* influence directement sur l'énergie consommée par ce nœud pour transmettre l'information captée à son *CH*. Conséquemment, il est préférable d'affecter chaque nœud à son *CH* le plus proche géographiquement pour optimiser la consommation énergétique des nœuds.
 - **Taille de cluster** : Le processus d'affectation des membres aux clusters dépend également de la taille du cluster. La création des clusters de tailles optimales a comme but l'équilibrage de charge du travail entre les *CHs*. En effet, un *CH* trop chargé risque d'épuiser son énergie très rapidement.

2.2.3 Objectifs de clustering

Le clustering a plusieurs objectifs qui peuvent varier selon l'application, ci-dessous nous citons les objectifs les plus pertinents.

- **Longévité maximale du réseau** : Depuis que les nœuds-capteurs sont limités en énergie, la durée de vie du réseau est une préoccupation majeure, en particulier pour les applications des *RCSFs* dans des environnements difficiles où le remplacement ou la recharge des batteries est quasiment impossible. Lorsque les *CHs* sont plus riches en ressources que les autres nœuds-capteurs, il est impératif de minimiser l'énergie pour la communication intra-cluster (entre les membres et les *CHs*) [21]. Si possible, les *CHs* doivent être placés à proximité de la plupart des nœuds-capteurs de leurs clusters [20]. Par contre, lorsque les *CHs* sont des nœuds-capteurs réguliers, leur durée de vie peut être étendue en limitant leurs charges.
- **Tolérance aux fautes** : les réseaux de capteurs sont non fiables, et les micro-capteurs peuvent être déployés dans des environnements hostiles ce qui les rend vulnérables aux dégâts matériels comme les cassures. La tolérance aux fautes dans un réseau de capteurs

est la capacité de ce dernier à maintenir ses fonctionnalités sans interruption et d'éviter la faille totale du système malgré la présence de quelques défaillances. Un protocole de clustering doit avoir l'aptitude de maintenir son fonctionnement correct en cas de pannes d'un *CH* ou de l'un de ses membres.

- **Augmentation de connectivité** : La connectivité inter-cluster est une exigence importante dans de nombreuses applications. Le but de la connectivité peut être seulement limité à assurer la disponibilité d'un chemin de tous les *CHs* à la *SB*.

2.2.4 Défis de clustering

Le clustering joue un rôle important pour l'amélioration de performances des réseaux de capteur sans fil et l'augmentation de la durée de vie du réseau. Néanmoins, cette solution est confrontée à une multitude de challenges, parmi lesquels nous citons :

- **Ressources matérielles limitées** : Comme tous les algorithmes dédiés aux réseaux de capteurs, ceux de clustering doivent prendre en considération les contraintes matérielles des micro-capteurs, surtout celles énergétiques. Ceci en minimisant, le plus que possible, le nombre de messages échangés pour la création et la maintenance des clusters et pour le routage (en choisissant les chemins optimaux, faisant des agrégations, etc.).
- **Tolérance aux pannes** : Dans de nombreuses applications, les *RCSFs* sont déployés dans des environnements hostiles et donc les nœuds sont généralement exposés à un risque accru de dommages physiques. L'expiration d'énergie étant une autre raison de dysfonctionnement des micro-capteurs. La panne d'un *CH* cause le partitionnement du réseau et la perte des données de son cluster. Par conséquent, des mécanismes de tolérances aux pannes doivent être mises en œuvre.
- **Dynamique du réseau** : Bien que la plupart des architectures des *RCSFs* supposent que les nœuds-capteurs sont stationnaires, il existe quelques applications nécessitant des micro-capteurs mobiles. A ce niveau, nous pouvons définir deux types de mobilité : les *CHs* et les autres membres du réseau. La mobilité des nœuds rendrait très difficile le clustering, car l'attachement des nœuds aux clusters va changer de façon dynamique. Bien évidemment, un algorithme de clustering efficace doit soutenir la mobilité des nœuds ou des *CHs*.

2.3 Protocoles de Clustering dans les RCSFs

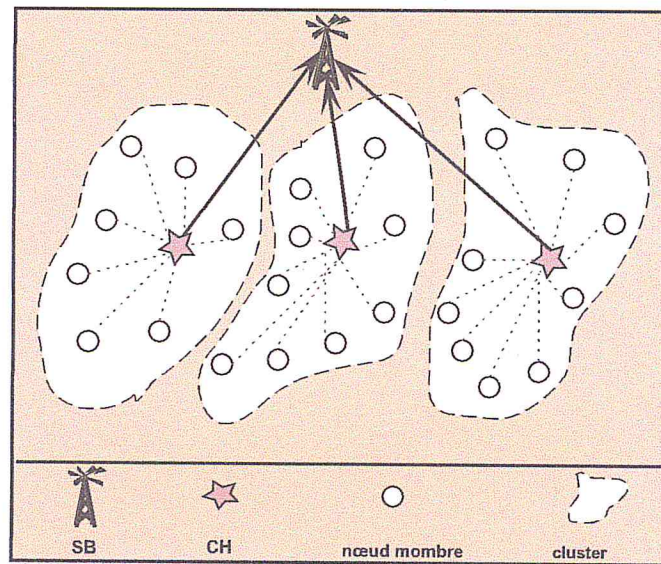
Au cours de ces dernières années, une large gamme de protocoles de clustering a été proposée par la communauté de chercheurs. Selon la méthodologie de clustering, les protocoles proposés peuvent être classés en deux grandes catégories : protocoles centralisés et distribués. Dans la première catégorie, la station de base gère l'algorithme de clustering en désignant les *CHs* et attribuant les membres de chaque cluster. Par contre, dans un protocole distribué, un nœud décidera de rejoindre un cluster ou devenir *CH* en se basant sur sa vue locale uniquement (informations sur le voisinage d'un seul ou deux saut(s) maximum). Les approches centralisées nécessitent que des informations globales sur la topologie du réseau soient envoyées à la *SB*, ce qui limite leur efficacité dans les réseaux à très grande échelle. Néanmoins, les protocoles centralisés fournissent une meilleure qualité de clustering en ce qui concerne le choix des *CHs*, l'affectation des membres aux clusters, etc. Ceci grâce à la connaissance totale de la topologie. En revanche, les algorithmes distribués sont plus adaptés aux réseaux à grande échelle mais peuvent avoir de mauvais choix de *CHs*, une mauvaise distribution de *CHs* dans le réseau, etc. à cause de la non connaissance de la topologie.

Nous résumons dans la section suivante quelques protocoles de clustering :

2.3.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

Architecture et fonctionnement de LEACH

LEACH est l'un des premiers et des plus populaires algorithmes de clustering proposés pour les *RCSFs* [7] [8]. C'est un algorithme hiérarchique, probabiliste, et distribué sans aucun contrôle centralisé. Il consiste à partitionner le réseau en groupes (clusters) montré dans la figure 2.1, ainsi qu'il utilise la rotation randomisée des *CHs* pour distribuer équitablement la charge d'énergie entre les nœuds du réseau. Les nœuds envoient leurs données aux *CHs* qui transmettent à leur tour ces données vers la *SB*. Les *CHs* font des fonctions de traitements (agrégation, fusion, compression par, etc.) sur les données avant de les transmettre en suivant une communication directe (à un seul saut).

FIGURE 2.1 – Architecture de protocole *LEACH*

LEACH s'exécute en deux grandes phases : la phase de configuration du réseau, « set-up phase » et la phase de transmission de données « steady-phase ».

(a) **La phase de configuration :**

Lors de cette phase, les nœuds *CH* sont élus et les clusters sont formés. L'élection des *CHs* se déroule comme suit :

Chaque nœud choisit un nombre de manière aléatoire entre 0 et 1. Si cette valeur est inférieure à un seuil T_n le nœud devient *CH*. Le seuil est défini comme suit :

$$T_n = \begin{cases} \frac{p}{1-p * r \bmod (\frac{1}{p})} & \text{si } n \in G \\ 0 & \text{Sinon} \end{cases} \quad (2.1)$$

Où p est le pourcentage de clusters souhaité, r est le round courant, G représente l'ensemble des nœuds qui n'ont pas été encore élus comme *CH* dans les derniers rounds.

Chaque nœud qui sera désigné comme *CH* pour le round courant diffuse un message d'initialisation au reste des nœuds. Les autres nœuds doivent maintenir leurs récepteurs à l'écoute afin d'entendre les messages diffusés par les *CHs*, et chacun de ces nœuds décide quel cluster rejoindre pour le round courant en se basant sur la puissance des messages (signaux) reçus de la part des *CHs*. Plus un nœud est proche du *CH*, plus le signal reçu sera puissant. Après que chaque nœud ait choisi quel cluster rejoindre, le *CH* doit être informé des nœuds composant son cluster. Chaque nœud non-*CH* transmet une requête de ralliement vers le *CH*.

Les *CHs* construisent des tables *TDMA* (Time Division Multiple Access) en se basant sur le nombre des nœuds dans leurs clusters. Dans ces tables on associe à chaque nœud un laps de temps durant lequel il peut transmettre ses messages. De cette manière, chaque nœud qui tente de faire une transmission durant son time-slot ; il est sûr que le canal est propre pour lui.

(b) **La phase de transmission de données :**

Lors de cette phase les nœuds envoient leurs données au *CH* du cluster auquel ils appartiennent en respectant le programme *TDMA* qui a été déjà mis en place.

Une fonction de traitement sera effectuée par le *CH* après que toutes les données sont reçues, pour plus de conservation d'énergie. Après que la transmission des données soit terminée, la phase de configuration du réseau aura lieu, comme expliqué plus haut.

Avantages et inconvénients

Dans ce qui suit, nous citerons quelques avantages et inconvénients du protocole *LEACH*. [9, 10]

• **Avantages :**

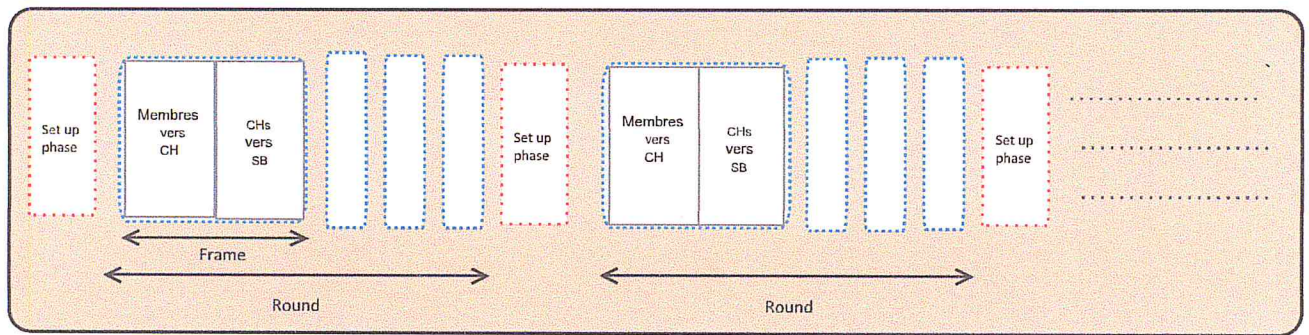
Parmi les avantages du protocole *LEACH* :

- L'auto-configuration des clusters se fait indépendamment de la station de base (algorithme distribué).
- La rotation des *CHs* permet de partager la charge de manière équitable entre les nœuds, et d'éviter la mort prématurée des *CHs*.
- L'utilisation du protocole *TDMA* permet de réduire la consommation d'énergie des nœuds puisque les nœuds sont actifs que durant leur slots.
- L'utilisation des techniques de fusion et d'agrégation de données réduit la quantité de trafic dans le réseau.

• **Inconvénients :**

En revanche, *LEACH* présente les inconvénients suivants :

- Les *CHs* les plus éloignés de la station de base meurent rapidement par rapport à ceux qui sont proches de la station. Ceci à cause de l'utilisation de la communication directe.

FIGURE 2.2 – Ordonnancement des rounds de protocole *LEACH-C*

Avantages et inconvénients

Dans ce qui suit, nous citerons quelques avantages et inconvénients du protocole *LEACH-C* [13],[14].

- **Avantages :**

Parmi les avantages du protocole *LEACH-C* :

- Une meilleure élection des *CHs* : non probabiliste, qui se base sur le niveau d'énergie et la localisation des nœuds dans le réseau faite par la station de base (centralisée).
- *LEACH-C* garantit une meilleure répartition des *CHs* dans le réseau.

- **Inconvénients :**

En revanche, *LEACH-C* présente les inconvénients suivants :

- La durée des rounds est aussi statique dans ce protocole.
- *LEACH-C* nécessite des informations de l'énergie résiduelle et de localisation de tous les nœuds qui consomme beaucoup d'énergie dans chaque rounds (re-clustering).

2.3.3 Dynamic round time based fixed LEACH (DRTBFL)

Fonctionnement de DRTBFL

Les protocoles *LEACH* et *LEACH-C* gaspillent beaucoup d'énergie dans :

- La re-clusterisation (la répétition de la phase de mise en place des clusters) dans chaque round.
- Le round toujours statique, cette durée fixe peut causer la mort prématurée des *CHs*.

Le protocole *LEACH-F* se montre comme une amélioration du protocole *LEACH* et *LEACH-C*, en répondant aux deux sources de gaspillage d'énergie citées précédemment. Dans *LEACH-F* [4], une fois les clusters sont formés, ils sont fixes. Autrement dit, aucune re-clusterisation n'est nécessaire au début de chaque round, ce qui mène à l'optimisation des dépenses énergétique et donc à la prolongation de la durée de vie.

LEACH-F utilise le même algorithme centralisé que *LEACH-C* pour former les clusters. La station de base diffuse ensuite des messages qui incluent l'*ID* de *CH* pour chaque nœud et l'ordre de faire tourner la position de *CH*. La position des nœuds dans la liste indique l'ordre de devenir *CH* dans les rounds suivants. Le processus de rotation se réinitialise lorsque tous les nœuds de la liste deviennent des *CHs*. De cette façon, le rôle du *CH* sera balancé entre les différents membres du même cluster, sans coût supplémentaire.

La phase de transmission de *LEACH-F* est la même que *LEACH* et *LEACH-C*, ainsi que le temps de round reste toujours fixe.

Pour répondre à la deuxième faiblesse concernant le round statique, un protocole nommé "Dynamic round time based fixed LEACH" [4] a été proposé comme une amélioration du protocole *LEACH-F*, les détails de ce protocole sont présentés ci-dessous :

- **modèle mathématique :**

L'idée de ce protocole consiste à définir une formule mathématique qui établit une relation entre la durée du round et l'énergie résiduelle des *CHs*, de sorte à éviter la mort prématurée des *CHs* au cours d'un round. En outre, tous les nœuds-capteurs membres sont synchronisés avec leurs *CHs* respectifs pour choisir le prochain *CH*, en suivant l'ordre de rotation envoyé par la *SB*. Autrement dit, la durée de round est la même dans chaque cluster, mais elle peut différer d'un cluster à un autre. Pour que la station de base puisse estimer la durée de chaque round, les auteurs de ce papier se sont basés sur la logique de fonctionnement de protocole *LEACH*. En effet, le round se divise en plusieurs frames. Dans chacun, tous les membres envoient leurs données à leurs *CHs* respectifs, et les *CHs* à leurs tours envoient les données vers la *SB*.

Soit $X_{frame/round}$ le nombre de frames par round et $E_{CH/frame}$ et $E_{nonCH/frame}$ sont les énergies consommées par frame pour les nœuds *CHs* et non-*CHs* respectivement. Alors, la consommation d'énergie des nœuds *CH* $E_{CH/round}$ et des nœuds non-*CH* $E_{non-CH/round}$ par round sont :

$$E_{CH/round} = E_{CH/frame} \times X_{frame/round} \quad (2.2)$$

$$E_{non-CH/round} = E_{non-CH/frame} \times X_{frame/round} \quad (2.3)$$

La formule du calcul de temps de chaque round garantit que l'énergie de chaque nœud suffise pour lui permettre d'être *CH* une fois et non-*CH* dans les autres $(N/K - 1)$ rounds où N représente la taille du réseau et K le nombre de clusters.

La démarche du calcul de la durée de round se résume dans ce qui suit :

$$E_{CH/round} + \left(\frac{N}{K} - 1\right)E_{non-CH/round} = E_{start} \quad (2.4)$$

$$X_{frame/round}((E_{CH/frame}) + \left(\frac{N}{K} - 1\right)E_{non-CH/frame}) = E_{start} \quad (2.5)$$

$$X_{frame/round} = \frac{E_{start}}{[(E_{CH/frame}) + \left(\frac{N}{K} - 1\right)E_{non-CH/frame}]} \quad (2.6)$$

Si l représente la taille du paquet et R_b représente le débit de paquet alors la durée de frame est :

$$t_{frame} = \frac{1}{R_b} \frac{N}{K} \text{ seconds} \quad (2.7)$$

Donc la durée de round se calcule comme suit :

$$t_{round} = X_{frame/round} \times t_{frame} \quad (2.8)$$

En remplaçant les équations 2.6, 2.7 dans 2.8, on obtient la formule suivante :

$$t_{round} = \frac{l}{R_b} \frac{N}{R} \frac{E_{start}}{[(E_{CH/frame}) + \left(\frac{N}{K} - 1\right)E_{non-CH/frame}]} \quad (2.9)$$

Avantages et inconvénients

Dans ce qui suit, nous citerons quelques avantages et inconvénients du protocole *DRTBFL* :

• Avantages :

Parmi les avantages du protocole *DRTBFL* :

- Ce protocole présente le même avantage du protocole *LEACH-F* qui concerne la minimisation du coût de re-clustering.
- Le temps de round dynamique minimise la mort prématurée des *CHs* ce qui permet de prolonger la durée de vie du réseau.

• Inconvénients :

En revanche, *DRTBFL* a les inconvénients suivants :

- Les clusters sont fixés durant toute la durée de vie et seul le rôle de *CH* est balancé entre les membres du même cluster. Par conséquent, les nœuds avec une faible quantité d'énergie résiduelle, ou qui sont loin de la station de base peuvent être élus comme *CHs*.

- Bien que ce protocole propose une technique pour rendre la durée de round dynamique. Cette technique est basée sur l'hypothèse que chaque nœud doit être *CH* une seule fois et non *CH* dans les autres $(N/K - 1)$ rounds. Néanmoins, cette hypothèse n'est pas nécessairement significative, car certains nœuds peuvent avoir une position défavorable qui les rend inappropriés pour agir comme *CH*.

2.3.4 Enhanced Adaptive Re-clustering Protocol (ADRP)

Architecture et fonctionnement de ADRP

Le problème majeur dans la plupart des protocoles de clustering des *RCSFs* est la re-clusterisation qui consomme beaucoup d'énergie, car les nœuds envoient des informations sur leur niveau d'énergie et leur localisation par une transmission directe ou par inondation à la station de base, au début de chaque round. Le protocole Enhanced Adaptive Re-clustering Protocol [5] essaye de remédier à ce problème.

Dans ce protocole, chaque nœud envoie à la *SB* des informations sur son emplacement actuel et le niveau d'énergie résiduelle lors du déploiement du réseau. La *SB* divise le réseau en clusters optimaux et choisit les *CHs* et les prochains *CHs*, comme montré dans la figure 2.3.

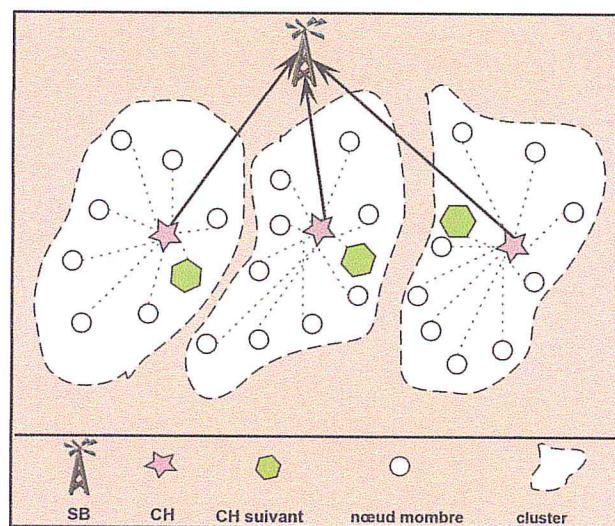
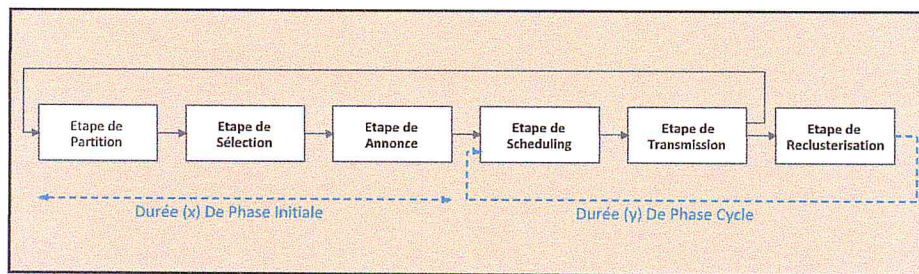


FIGURE 2.3 – Architecture de protocole *ADRP*

L'activité de ce protocole est organisée en rounds, où chaque round est divisé en deux phases : phase initiale et phase cycle [6], comme il est montré dans la figure 2.4, la durée de la phase de cycle est plus longue que la durée de la phase initiale.

FIGURE 2.4 – Fonctionnement de protocole *ADRP***(a) La phase initiale**

Cette phase est divisée en 3 étapes : partition, sélection et annonce.

- **Partition et sélection :**

La station de base reçoit les informations sur le niveau d'énergie et la position de chaque nœud, et les utilise pour diviser le réseau en clusters. Ensuite, elle détermine les *CHs* de chaque cluster, et les prochains *CHs* en calculant la moyenne d'énergie de chaque cluster dans le réseau. Les nœuds ayant une énergie résiduelle en dessous de cette moyenne ne peuvent pas être élus comme prochains *CHs*.

- **Annonce :**

Pendant cette étape, la station de base diffuse un message contenant l'ID des *CHs* et les *CHs* à tous les nœuds.

(b) Phase cycle

Cette phase est divisée aussi en 3 étapes : scheduling, transmission et reclustering.

- **Scheduling :**

Une fois les clusters formés, les nœuds capteurs doivent envoyer leurs données à leurs *CHs* en utilisant *TDMA*. Ce dernier permet aux nœuds de suivre un programme pour passer en mode veille lorsqu'ils ne transmettent pas de données et d'éviter les collisions intra-cluster.

- **Transmission :**

L'étape de transmission des données se compose de trois grandes activités :

- Collecte et envoi des données.
- Agrégation de données.
- Envoi de données (*CSMA*)

- **Reclusterisation :**

Dans la phase initiale, la station de base envoie des messages contenant l'ID de *CH* et les *CHs* suivantes pour chaque nœud de capteur du réseau, donc les nœuds peuvent passer directement aux *CHs* suivants sans communiquer avec la station de base.

Avantages et inconvénients

Dans ce qui suit, nous citerons quelques avantages et inconvénients du protocole Enhanced Adaptive Re-clustering Protocol.

- **Avantages :**

Tout comme *LEACH-F*, ce protocole évite le coût du re-clustering au début de chaque round, ce qui réduit la consommation d'énergie au niveau des nœuds-capteurs. De plus, le protocole *ADRP* utilise des critères liés à l'énergie résiduelle de chaque nœud, l'énergie de transmission, et la moyenne d'énergie dans chaque cluster pour élire les prochains *CHs*, contrairement à *LEACH-F* qui utilise un simple ordonnancement.

- **Inconvénients :**

Ce protocole partage le même inconvénient que celui de dans le fait les clusters sont créés une seule fois, lors de déploiement du réseau. Par la suite, seul le rôle de *CH* est balancé entre les membres du même cluster. De ce fait, ce protocole assure un équilibrage de charge intra-cluster et non pas entre tous les nœuds du réseau. A titre d'exemple, si le nombre de membres par cluster n'est pas équitable, les clusters qui contiennent moins de membres seront plus surchargés au cours de la durée de vie du réseau.

2.3.5 Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network (TPSO-CR)

Le protocole *TPSO-CR* [26] est un algorithme centralisé basé sur la méta-heuristique *PSO* (optimisation par essaim de particules). Il résout deux problèmes essentiels des *RCSFs* qui sont : le clustering et le routage.

Fonctionnement de TPSO-CR

Dans *TPSO-CR*, le temps de fonctionnement est divisé en rounds. Chaque round se compose de deux phases, la phase de configuration et la phase de transmission qui est identique à celle du protocole *LEACH-C*. Dans la première phase, le réseau est configuré. La station de base choisit les meilleurs clusters et *CHs*, et elle construit l'arbre de routage qui relie ces *CHs*. Pour en faire, chaque nœud dans le réseau diffuse un paquet HELLO qui inclut son ID. Le nœud qui reçoit ce paquet HELLO met à jour sa table voisinage par l'ID et l'indicateur de puissance de signal reçu (*RSSI*) dans le paquet reçu. Par la suite, chaque nœud diffuse un message qui contient : son ID, l'énergie résiduelle et la table de voisinage en utilisant la méthode d'inondation (flooding) jusqu'à atteindre la station de base.

Sur la base des informations reçues par le *SB*, ce dernier calcule le niveau d'énergie moyen de tous les nœuds. Seuls les nœuds ayant un niveau d'énergie supérieur au moyen sont éligibles pour être candidats de *CH* pour ce round. Pour s'assurer que seuls les nœuds avec une énergie suffisante sont choisis comme *CHs*, la *SB* exécute le premier niveau (l'algorithme de *TPSO-CR* pour trouver les meilleurs *K CHs*).

(a) Initialisation des particules :

La dimension de la particule est la même que le nombre de *CH* (c'est-à-dire *K*) dans le réseau avec $P_i = [X_{i,1}, X_{i,2}, X_{i,3}, \dots, X_{i,k}]$ être la *i*ème particule de la population où chaque composante, X_{id} est le *ID* de *d*ème *CH* tel que $1 \leq d \leq K$. Chaque Composante est initialisée aléatoirement avec numéro entre 1, taille de réseau -1.

(b) Evaluation des particules :

Après l'initialisation des particules ; on les évalue à partir de fonctions de fitness. Cela contribue à mettre à jour périodiquement *Pbest_i* et *Gbest* des particules.

Pour évaluer les particules, nous utilisons la fonction de fitness globale définie en suite :

$$C_p = WC_1 \times EE_p + WC_2 \times CQ_p + WC_3 \times NC_p \quad (2.10)$$

Cette fonction de fitness est divisée par trois sous fonction de fitness dont le but d'optimiser trois critères suivants :

- **Efficacité énergétique :**

$$EE_p = \sum_{k=1}^k \frac{InitialE(CH_{p,k})}{E(CH_{p,k})} \quad (2.11)$$

D'après l'équation, on déduit que si l'énergie résiduelle augmente, la valeur retournée par l'équation diminue et puisque on a un problème de minimisation, seulement les nœuds qui disposent d'une meilleure énergie résiduelle vont être choisis.

- **Qualité de cluster :**

$$CQ_p = \max_{k=1,2,\dots,k} \frac{\sum \forall n_i \in C_{p,K} LQ_{(n_i, CH_{p,k})}}{|C_{(p,k)}|} \quad (2.12)$$

Le but de cette fonction est de maximiser la qualité du lien entre les membres des clusters et leurs *CHs* respectifs.

tel que :

$$LQ_{(n_i, CH_{p,k})} = \frac{RSSI_{n_i, CH_{p,k}}}{\min RSSI} \quad (2.13)$$

représente la qualité du lien entre un membre de cluster n_i et un *CH* numéro k dans une particule p .

- **Couverture du réseau :**

Afin d'assurer que le protocole choisi garantisse le passage à l'échelle, on doit alors augmenter la couverture du réseau selon la fonction suivante.

$$NC_p = \frac{N - \sum_{k=1}^k |C_{p,k}|}{\sum_{k=1}^k |C_{p,k}|} \quad (2.14)$$

Après avoir trouver l'ensemble optimal de *CHs* dans le premier niveau, le deuxième niveau du protocole est responsable de la construction de l'arbre de routage. La *SB* exécute *PSO* à nouveau pour réaliser cette tâche.

Configuration de la diffusion : après que la *SB* termine la configuration du réseau, elle diffuse un paquet contenant cette configuration. Chaque nœud qui reçoit ce paquet va modifier son statut soit en tant que *CH*, un membre ou un nœud de relais.

Avantages et Inconvénients

Dans ce qui suit, nous citerons quelques avantages et inconvénients du protocole Enhanced Adaptive Re-clustering Protocol.

- **Avantages :**

- Bonne qualité de clustering en ce qui concerne l'élection des *CHs* grâce à la méta-heuristique *PSO*.

- **Inconvénients :**

- Coût du re-clustering au début de chaque round.
- Durée de round statique.

2.3.6 Enhanced Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network (ETPSO-CR)

Pour pallier aux inconvénients du protocole *TPSO-CR* à savoir la re-clusterisation au début de chaque round et les durées statiques des rounds, une version améliorée de ce protocole a été proposée, nommée *ETPSO-CR* pour "Enhanced Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network". Les deux protocoles *TPSO-CR* et *ETPSO-CR* partagent le même principe et utilisent les mêmes algorithmes de clustering et de routage. La différence entre ces deux protocoles réside dans ce qui suit :

le cout de re-clustering périodique a été éliminé tout en maintenant ses avantages. Ceci grâce à l'introduction d'une technique d'estimation d'énergie. Autrement dit, au lieu que les noeuds envoient la valeur d'énergie résiduelle au début de chaque round, la station de base utilise un modèle d'estimation d'énergie pour prédire l'énergie résiduelle des différents noeuds du réseau. En se basant sur ces valeurs estimées, la *SB* refait le clustering et diffuse le résultat dans le réseau.

La deuxième différence est la durée des rounds qui est statique dans *TPSO-CR* et dynamique dans *ETPSO-CR*. L'idée de base proposé dans *ETPSO-CR* est que la durée de round ne doit pas être très courte au point au l'énergie du noeud ne sera pas exploitée convenablement, ni trop longue pour permettre au noeud d'acter comme membre dans les prochains rounds. Plutôt, elle doit être soigneusement ajustée de telle sorte à optimiser l'utilisation des batteries des noeuds. Pour en faire, un seuil qui se calcule à partir de l'énergie résiduelle de chaque *CH* a été défini. Après la génération de schéma du clustering et du routage au début de chaque round, de la même manière que le protocole *TPSO-CR*, la *SB* estime la quantité d'énergie nécessaire à chaque *CH* pour satisfaire ce schéma pour un seul frame. Si la valeur de l'énergie résiduelle de tous les *CHs* est inférieure au seuil, alors nous déduisons que les *CHs* disposent des énergies suffisantes pour accomplir leurs rôles dans le prochain frame. Conséquemment, la durée du round sera incrémentée par un frame. Ensuite, la *SB* refait la même procédure jusqu'à ce que l'énergie résiduelle de l'un des *CHs* atteigne le seuil. Dans ce cas, la *SB* diffuse le schéma du clustering et du routage ainsi que la durée de round calculée aux noeuds du réseau.

2.4 Comparaison

Pour mieux comprendre la différence entre les différents protocoles étudiés précédemment, on va les comparer en suivant certains critères essentiels qui influencent sur la consommation énergétique des nœuds. La comparaison est montrée dans la table 2.4 :

- Type de protocole : Centralisé ou distribué.
- Round : Statique ou dynamique.
- Type de routage : Un seul saut ou multi-sauts.
- Reclusterisation : Une seule fois ou à chaque round.

| Protocoles | Type de protocoles | Round | Type de routage | Re-clustering |
|-----------------|--------------------|-----------|-----------------|----------------|
| <i>LEACH</i> | Distribué | Statique | Un seul saut | A chaque round |
| <i>LEACH-C</i> | Centralisé | Statique | Un seul saut | A chaque round |
| <i>ADRP</i> | Centralisé | Statique | Un seul saut | Une seule fois |
| <i>DRTPBFL</i> | Centralisé | Dynamique | Un seul saut | Une seule fois |
| <i>TPSO-CR</i> | Centralisé | Statique | Multi-sauts | A chaque round |
| <i>ETPSO-CR</i> | Centralisé | Dynamique | Multi-sauts | Une seule fois |

TABLE 2.1 – Comparaison entre quelques protocoles de clustering.

2.5 Conclusion

Le clustering est l'une des techniques efficace pour assurer la scalabilité et améliorer la durée de vie des réseaux de capteurs sans fil. Dans ce chapitre, nous avons présenté les différents critères considérés lors d'une clusterisation d'un *RCSFs*, les défis du clustering et ses objectifs. Nous avons également passé en revue quelques algorithmes de clustering tout en discutant leurs avantages et inconvénients. L'étude détaillée et la conception d'un nouveau protocole de routage hiérarchique seront le sujet de notre prochain chapitre.

Deuxième partie

Propositions et évaluations

CHAPITRE 3

PROPOSITION ET CONCEPTION D'UN PROTOCOLE DE CLUSTERING

3.1 Introduction

Depuis l'apparition des réseaux de capteurs sans fil, les recherches se sont penchées sur la proposition des nouveaux protocoles visant l'optimisation de la consommation énergétique et donc la prolongation de la durée de vie du réseau. Comme nous l'avons vu dans le chapitre précédent, la clusterisation du réseau est l'une des techniques proposées pour répondre à la problématique d'énergie dans les *RCSFs*. Néanmoins, la performance des protocoles de clustering est fortement liée à la qualité de la hiérarchie créée, qui est reflétée par le choix des *CHs*, l'affectation des membres aux clusters et la distribution des *CHs* sur le réseau. La durée attribuée à chaque round et la fréquence du re-clustering influencent également l'efficacité énergétique des protocoles de clustering.

Les chercheurs dans le domaine des *RCSFs* confirment que les protocoles de clustering centralisés offrent une meilleure qualité de clustering par rapport à ceux distribués (voir section 2.2.3). Bien évidemment, une bonne qualité de clustering dit une meilleure durée de vie du réseau. Dans cette optique, nous allons concevoir dans ce chapitre un nouveau protocole de clustering centralisé, qui essaye de pallier aux inconvénients des protocoles de clustering existants, évoqués précédemment dans la section 2.3. Pour cela, nous commençons par présenter nos hypothèses et le modèle énergétique utilisé. Nous passerons par la suite à la description de la solution proposée ainsi que sa conception.

3.2 Hypothèses

Le nouveau protocole est conçu selon les hypothèses suivantes :

- Le réseau de capteurs est constitué de N nœuds, et une seule station de base avec une source d'énergie illimitée.
- Les nœuds-capteurs et la station de base sont tous stationnaires après déploiement.
- Le type d'application considéré par ce protocole est celui de collecte périodique de données.
- Chaque nœud appartient à un et un seul cluster et chaque *CH* est le chef d'un seul cluster.
- Les messages envoyés par les nœuds ont la même taille.
- Chaque nœud connaît sa position locale.
- Initialement les nœuds peuvent atteindre directement la *SB*.

- Les nœuds ont la possibilité d'ajuster leur puissance de transmission.

3.3 Modèle de consommation d'énergie

Les principales sources de consommation d'énergie d'un nœud capteur sont le captage, le traitement, (les calculs et l'agrégation de données) et la communication radio. Les valeurs énergétiques consommées en captage et calcul seront fixées dans le prochain chapitre. Quant à la transmission radio, nous avons adopté le modèle "First Order Radio Model" [23] qui est utilisé dans la plupart des protocoles de clustering dans les *RCSFs* (comme *LEACH* et ses dérivés). Dans ce modèle, l'énergie consommée dans la transmission est proportionnelle à la fois à la taille des données et à la distance. La distance d entre l'émetteur et le récepteur détermine le modèle de canal à utiliser. Si d est inférieure à un seuil d_0 alors le modèle en espace libre fs (free space) est utilisé, sinon, le modèle de propagation d'évanouissement multivoie mp (multipath fading) est considéré. Pour transmettre l bits de données sur une distance d , l'énergie requise est :

$$\begin{cases} E_{tx}(l, d) = l \times E_{elec} + l \times E_{fs} \times d^2 & \text{Si } (d \leq d_0) \\ E_{tx}(l, d) = l \times E_{elec} + l \times E_{mp} \times d^4 & \text{Si } (d > d_0) \end{cases} \quad (3.1)$$

Tel que :

- E_{elec} (radio electronics) est l'énergie nécessaire pour allumer le circuit électronique.
- l est le nombre de bits du paquet à transmettre.
- E_{fs} est l'énergie d'amplification dans le modèle en espace libre.
- E_{mp} est l'énergie d'amplification dans le modèle d'évanouissement multivoie.
- d_0 est la distance de passage (crossover distance), elle est calculée par l'équation 3.2.

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (3.2)$$

En outre, pour recevoir l bits, l'énergie requise est :

$$E_{rx}(l) = l \times E_{elec}. \quad (3.3)$$

3.4 Description de la solution proposée

Notre travail consiste à concevoir un protocole de clustering qui a pour objectif la minimisation de la consommation d'énergie des micro-capteurs, afin de prolonger la durée de vie du réseau. Nous nous sommes basés sur le principe du protocole *LEACH-C*, où le clustering se fait d'une manière centralisée, par la *SB*. Cependant, contrairement à *LEACH-C*, dans notre protocole la procédure de clustering ne se fait qu'une seule fois, lors de déploiement du réseau. Ceci afin d'éviter le processus de re-clustering qui consiste à l'envoi des énergies résiduelles à la *SB* au début de chaque round, ce qui cause la dissipation d'une grande quantité d'énergie. Néanmoins, il est bien connu que le re-clustering joue un rôle clé dans la prolongation de la durée de vie du réseau via la distribution de rôle du *CH* entre les différents nœuds. Pour éliminer le coût de re-clustering tout en maintenant ses avantages, une nouvelle modélisation du problème de clustering, basée sur l'estimation d'énergie consommée au niveau de chaque nœud, a été proposée. L'objectif principal de cette modélisation est de répondre aux limitations des protocoles de clustering présentés dans la section 2.3 du chapitre précédent, tels que la durée statique des rounds, la distribution des *CHs* sur le réseau, etc.

Dans ce qui suit, nous présentons une nouvelle modélisation mathématique du problème de clustering. Ensuite, nous détaillons la méta-heuristique particule swarm optimization (*PSO*) (puisque c'est la méta-heuristique utilisée pour résoudre la modélisation). Après, nous donnons une description détaillée du protocole de clustering proposé.

3.4.1 Modélisation du problème de clustering dans les RCSFs

Afin de maximiser la durée de vie des réseaux de capteurs sans fil, le problème de clustering a été modélisé comme un programme non-linéaire en nombre entier (nonlinear integer programming, *NIP*). Les motivations derrière cette modélisation peuvent se résumer en :

Premièrement, la modélisation proposée permet à tous les facteurs influençant l'efficacité énergétique des protocoles de clustering d'être adaptatives. Autrement dit, aucun facteur n'est fixé au préalable. Cela permet d'attribuer les valeurs qui optimisent la durée de vie du système à tous les facteurs, i.e. nombre de round, durée de chaque round, nombre de *CHs* par round, taille des clusters par rounds, les nœuds à élire comme *CHs*, et l'affectation des membres à chaque *CH*. La deuxième motivation est la proposition d'une fonction objective qui maximise directement la durée de vie du réseau, contrairement à toutes les fonctions objectives proposées dans la littérature qui visent à optimiser un sous critères, tels que les distances entre les *CHs*, les énergies résiduelles des *CHs*, etc. Troisièmement, l'évitement du coût du re-clustering tout en

préservant ses avantages. Ceci est atteint via l'utilisation d'un modèle d'estimation d'énergie au niveau de chaque nœud.

Ci-après nous détaillons la modélisation proposée, en commençant par les constantes qui représentent les paramètres d'entrées du système, les variables qui sont les inconnus que nous voulons trouver leurs valeurs optimales, la fonction objective à optimiser et les contraintes de la modélisation.

Définition des constantes :

- N : Valeur entière, qui représente la taille du réseau (nombre de nœuds).
- K : Valeur entière, qui représente la borne supérieure de nombre de rounds final.
- $R[]$: Tableau de taille N de valeurs réelles, qui représentent les énergies initiales des nœuds. $R[i]$ représente l'énergie initiale du nœud i .
- E_p^1 : Valeur entière, qui représente l'énergie consommée par un nœud pour faire le captage dans une unité du temps.
- E_{ph}^1 : Valeur entière, qui représente l'énergie consommée par un CH pour recevoir un paquet de données et faire l'agrégation.
- $ENN[][]$: Matrice de taille $N * N$ de valeurs réelles, qui représentent les énergies de communication entre les nœuds. $ENN[i][j]$ représente l'énergie nécessaire pour transmettre un paquet de nœud i vers le nœud j .
- $ENB[]$: Tableau de taille N de valeurs réelles, qui représentent les énergies de communication entre les nœuds et la station de base. $ENB[i]$ représente l'énergie nécessaire pour transmettre un paquet de nœud i vers la station de base.

Définition des variables :

- $T = t_k \setminus 1 \leq k \leq K$: Tableau de K valeurs entières qui représentent les durées de vie des rounds. t_k représente la durée du round k .
- $C^k = C(i, j)^k \in \{0, 1\} \setminus 1 \leq i, j \leq N \wedge 1 \leq k \leq K$: K matrices binaires qui représentent les configurations de clustering dans chaque round. C^k représente la configuration de clustering du round k .

Fonction objective :

$$\max \sum_{k=1}^K t_k \quad (3.4)$$

L'objectif est de maximiser la durée de vie du réseau. Ceci dit, maximiser la somme des durées des rounds.

Contraintes :

1.

$$\sum_{j=1}^N C_{i,j}^k = 1 \dots \forall k \leq K, \forall i \in [1, N] \quad (3.5)$$

2.

$$\sum_{i=1}^N C_{i,j}^k \leq C_{i,i}^k \times \sum_{i=1}^N C_{i,j}^k \dots \forall j \in [1, N], \forall k \in [1, K] \quad (3.6)$$

3.

$$\sum_{k=1}^K t_k (E_p^1 + \sum_{j=1}^N (ENN_{i,j}^1 \times C_{i,j}^k) + ENB_i^1 \times C_{j,j}^k + E_{ph}^1 \sum_{j=1, j \neq i}^N C_{j,i}^k) \leq R_i \dots \forall i \in [1, N] \quad (3.7)$$

La première contrainte a le rôle d'imposer à chaque nœud du réseau d'être attribué à un et un seul *CH*, et cela doit être vérifié dans chaque round. L'objectif de la deuxième contrainte est d'assurer qu'il n'aura pas de membres attribués à un nœud non désigné comme *CH*. De même que la première contrainte, celle-ci doit être vérifiée dans chaque round. La troisième contrainte, étant la contrainte la plus importante, et qui reflète la contribution de cette modélisation. Elle permet de garantir la génération des configurations de clustering qui optimisent la fonction objective, tout en assurant que l'énergie des nœuds suffit pour satisfaire ces configurations pendant toute la durée de vie du réseau.

3.4.2 Particle swarm optimization (PSO)

L'optimisation par essaim particulaire (*OEP*), ou Particle Swarm Optimization (*PSO*) en anglais, est un algorithme évolutionnaire qui utilise une population de solutions candidates pour développer une solution optimale au problème. Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, tels que les bancs de poissons et les vols groupés d'oiseaux qui doivent parcourir des longues distances et optimiser leurs déplacements en termes d'énergie dépensée.

Pour appliquer la *PSO* il faut définir un espace de recherche constitué de :

- Particules (chaque particule est considérée comme une solution du problème), ces dernières sont placées aléatoirement dans l'espace de recherche.
- La fonction objectif à optimiser.

Le principe de l'algorithme est de déplacer ces particules afin qu'elles trouvent l'optimum.

Chacune de ces particules est dotée :

- D'une position.
- D'une vitesse.
- D'un voisinage.

A tout instant, chaque particule connaît :

- Sa meilleure position visitée.
- La position du meilleur voisin de l'essaim qui correspond à l'ordonnancement optimal.
- La valeur qu'elle donne à la fonction objectif car à chaque itération il faut une comparaison entre la valeur du critère donnée par la particule courante et la valeur optimale.

L'évolution d'une particule est finalement une combinaison de trois types de comportements [15] :

- Un comportement égoïste (suivre sa voie en suivant sa vitesse actuelle).
- Un comportement conservateur (revenir en arrière en prenant en compte sa meilleure performance).
- Un comportement sociale (suivre aveuglément le meilleur de tous en considérant sa performance).

Dans un espace de recherche de dimension D , la particule i de l'essaim est modélisée par son vecteur position $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ et par son vecteur vitesse $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. La qualité de sa position est déterminée par la valeur de la fonction objectif en ce point. Cette particule garde en mémoire la meilleure position par laquelle elle est déjà passée, que l'on note $Pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$. La meilleure position atteinte par les particules de l'essaim est notée $Gbest = (gbest_1, gbest_2, \dots, gbest_D)$.

Nous nous référons à la version globale de *PSO*, où toutes les particules de l'essaim sont

considérées comme voisines de la particule i , d'où la notation $Gbest$ (global best).

Au départ de l'algorithme, les particules de l'essaim sont initialisées de manière aléatoire/régulière dans l'espace de recherche du problème. Ensuite, à chaque itération, chaque particule se déplace, en combinant linéairement les trois types de comportements cités ci-dessus. En effet, à l'itération $t + 1$, le vecteur vitesse et le vecteur position sont calculés à partir de l'équation 3.8 et de l'équation 3.9, respectivement.

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1r_{i,j}^t[pbest_{i,j}^t - x_{i,j}^t] + c_2r_{i,j}^t[gbest_j^t - x_{i,j}^t], j \in \{1, 2, \dots, D\} \quad (3.8)$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}, j \in \{1, 2, \dots, D\} \quad (3.9)$$

Où w est une constante, appelée coefficient d'inertie ; c_1 et c_2 sont deux constantes, appelées coefficients d'accélération ; r_1 et r_2 sont deux nombres aléatoires tirés uniformément dans $[0, 1]$, à chaque itération t et pour chaque dimension j . Les trois composantes mentionnées ci-dessus (i.e. égoïste, conservateur et sociale) sont représentées dans l'équation 3.8 par les termes suivants :

- $wv_{i,j}$ correspond à la composante d'inertie du déplacement, où le paramètre w contrôle l'influence de la direction de déplacement sur le déplacement futur ;
- $c_1r_{i,j} [pbest_{i,j} - x_{i,j}]$ correspond à la composante cognitive du déplacement, où le paramètre c_1 contrôle le comportement cognitif de la particule ;
- $c_2r_{i,j} [gbest_j - x_{i,j}]$ correspond à la composante sociale du déplacement, où le paramètre c_2 contrôle l'aptitude sociale de la particule.

Une fois le déplacement des particules effectué, les nouvelles positions sont évaluées et les deux vecteurs $Pbest_i$ et $Gbest$ sont mis à jour, à l'itération $t + 1$, suivant les deux équations 3.10 (dans le cas d'une minimisation) et 3.11 dans une version globale de PSO , respectivement.

$$\vec{P}best_i(t+1) = \begin{cases} \vec{P}best_i(t) & \text{si } f(\vec{x}_i(t+1)) \geq \vec{P}best_i(t) \\ \vec{x}(t+1) & \text{Sinon} \end{cases} \quad (3.10)$$

$$\vec{G}best(t+1) = \arg \min_{\vec{P}best_i} f(\vec{P}best_i(t+1)), 1 \leq N \quad (3.11)$$

La procédure générale de PSO est présentée dans l'algorithme ci-dessous, où N est le nombre de particules de l'essaim.

1. Début
2. Initialiser aléatoirement N particules : position et vitesse.
3. Évaluer les positions des particules
4. Pour chaque particule $i, Pbest_i = x_i$
5. Calculer $Gbest$ selon 3.11
6. tant que le critère d'arrêt n'est pas satisfait faire
7. Déplacer les particules selon 3.8 et 3.9
8. Évaluer les positions des particules
9. Mettre à jour $Pbest_i$ et $Gbest$ selon 3.10 et 3.11
10. Fin

3.4.3 Fonctionnement général du protocole proposé

Le fonctionnement général du protocole proposé peut être divisé en deux phases principales, à savoir, la clusterisation du réseau et la transmission des données. Ces deux phases sont détaillées dans ce qui suit :

Clusterisation du réseau :

Initialement, chaque nœud envoie un paquet «INITIALISATION» qui contient sa position géographique et son énergie initiale à la SB . Lors de la réception des paquets de tous les nœuds, la SB applique un algorithme de clustering centralisé afin de déterminer les schémas de clusterisation, le nombre de rounds et leurs durées. Cet algorithme est conçu pour résoudre la modélisation présentée dans la section 3.4.1. L'algorithme est basé sur la méta-heuristique PSO . Pour le choix de cette méta-heuristique, nous nous sommes basés sur le travail de Latiff et A.L. [22] qui montre que par rapport aux problèmes de clustering, PSO surpasse les autres algorithmes tels que : l'algorithme génétique (AG) et l'algorithme $K-means$, en ce qui concerne le temps de convergence et la qualité de la solution obtenue. Comme nous l'avons déjà mentionné, l'algorithme de clustering proposé génère tous les schémas de clustering et leurs durées respectives une fois, lors de l'initialisation du réseau. Cela fournit à la SB une vision globale et lui permet d'atteindre de meilleurs schémas de clustering avec des durées de rounds adaptées.

Ainsi, l'algorithme permet d'éliminer le coût proportionnel au re-clustering périodique, en utilisant un modèle de prédiction énergétique.

Le résultat de l'algorithme est un ensemble de configurations. Chaque configuration représente un schéma de clustering qui correspond à un round et la durée du round est dynamique, cette phase sera détaillée dans la section 3.4.4.

Transmission des données :

Après avoir terminé le processus de clustering, la station de base déclenche la phase de transmission de données «steady state phase», qui est identique à celle du protocole *LEACH-C*, avec la différence que les nœuds n'auront pas à envoyer la valeur de leur énergie résiduelle dans chaque round. Par contre, au début de chaque round, la station de base diffuse un message «*HIERARCHIE*» dans le réseau. Ce message contient le schéma de clustering du round actuel ainsi que sa durée. Chaque nœud qui reçoit ce message cherche son identificateur (ID) dans ce message pour savoir s'il est élu comme *CH*, ou au quel *CH* il est rattaché.

Ensuite, chaque *CH* planifie un ordonnancement (scheduling) basé sur le protocole *TDMA* et l'envoi à ses membres. A la réception de ce scheduling, chaque nœud membre envoie la donnée qu'il a captée à son *CH* durant le temps qui lui a été alloué. Pour le reste du temps, sa radio doit être éteinte afin de préserver son énergie. A la réception de toutes les données, le *CH* les agrège et les envoie à la station de base.

Le protocole de clustering proposé construit des clusters uni-saut intra et inter-cluster. Ceci dit que chaque nœud utilisera la communication directe pour atteindre son *CH*. Idem pour les *CHs* qui enverront les données agrégées directement vers la station de base. Dans ce cas nous parlons d'une simple transmission directe plutôt qu'un routage.

L'ordonnancement des rounds de la solution proposée est illustré dans la figure 3.1

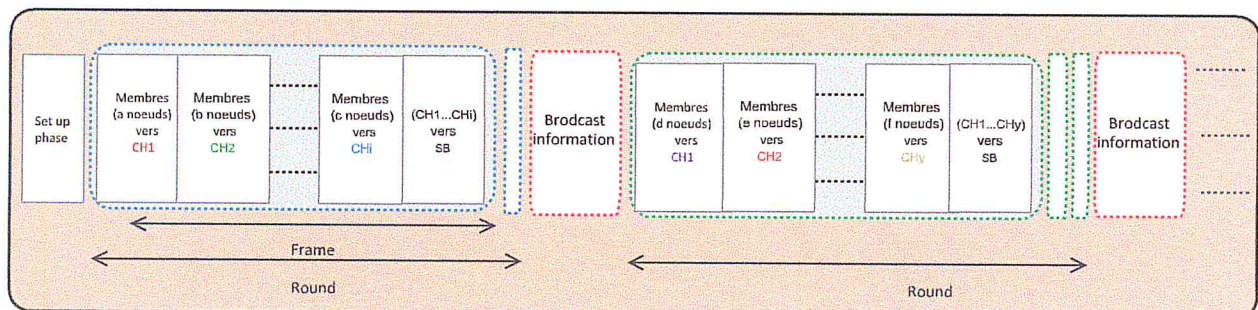


FIGURE 3.1 – L'ordonnancement des rounds

Comme il est clair dans cette figure 3.1, le nombre et la taille des clusters sont dynamiques

et les durées de rounds sont adaptatives. Ainsi, la durée de la phase de transmission de données est améliorée à cause de l'élimination de l'étape du re-clustering périodique dans le protocole *LEACH-C* et ses descendants (voir figure 2.2 du chapitre précédent).

Afin d'éclaircir le fonctionnement général de la solution proposée, notamment la phase de transmission de données, nous proposons l'organigramme de la figure 3.2.

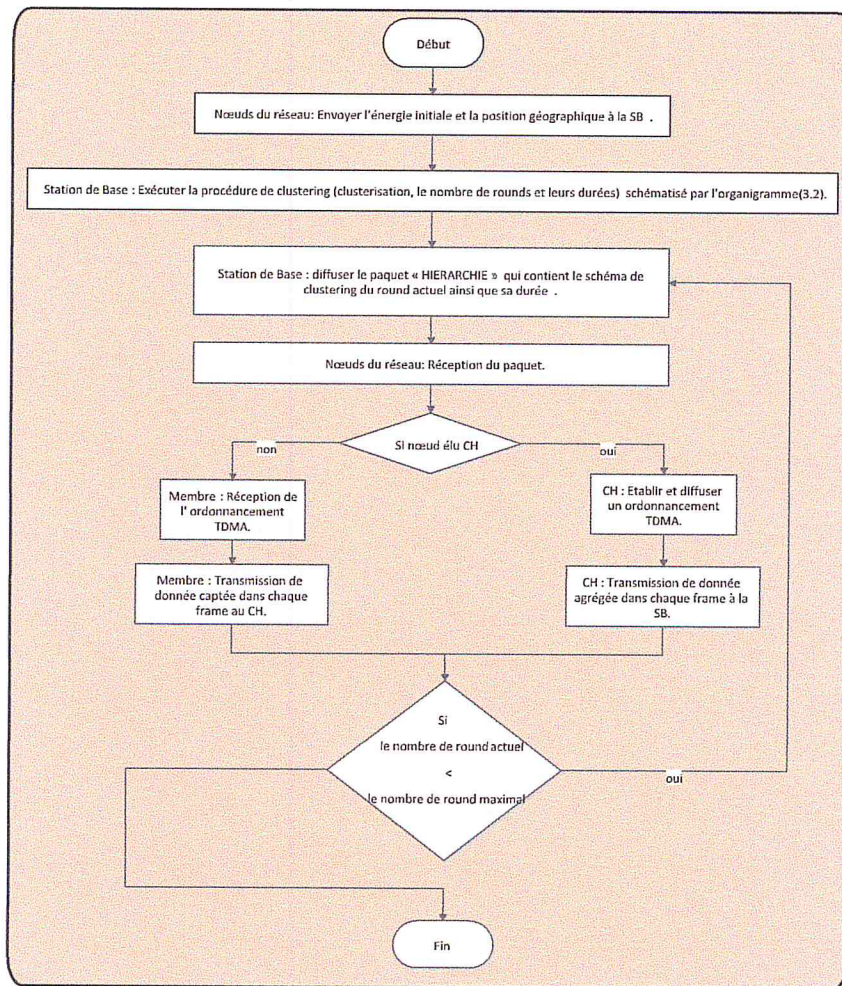


FIGURE 3.2 – Organigramme du Protocole 2OSC.

3.4.4 Algorithme de clustering centralisé basé sur PSO

Cette section a pour but la présentation de l'algorithme de clustering proposé pour résoudre la modélisation décrite dans la section 3.4.1. Cet algorithme est nommé *2OSC*, pour "Optimized One-Step clustering". Comme déjà mentionné, *2OSC* est un algorithme centralisé, basé sur la méta-heuristique *PSO*. Le fonctionnement général de l'algorithme de clustering peut être divisé en deux étapes principales :

Obtention des configurations de clustering :

Puisque la modélisation proposée contient deux variables, à savoir les configurations de clustering et les durées des rounds, donc nous avons proposé un algorithme qui fonctionne en deux étapes. La première étape consiste à obtenir les K configurations de clustering $C_{(i,j)}$, tel que K est la borne supérieure du nombre de rounds. Comme déjà mentionné, les configurations sont des matrices binaires de taille $(N * N)$. Ces configurations seront générées d'une manière aléatoire en respectant les deux contraintes 3.5 et 3.6 de la modélisation proposée. Par la suite, l'énergie nécessaire pour satisfaire ces configurations, pour un seul frame du temps, par chaque nœud, est estimée. L'estimation de l'énergie consommée par un nœud i , dans un frame f , est basée sur le modèle de consommation énergétique discuté dans la section 3.3 et du rôle de ce nœud dans ce frame. En effet, si le nœud i est un simple membre alors l'énergie sera dissipée pour capter l'information et pour l'envoyer à son CH. En revanche, si le nœud i est désigné comme CH, l'énergie sera dissipée pour recevoir les données de tous ses membres, agréger ces données et les envoyer à SB . L'énergie consommée par un nœud membre i et par un CH j est calculée par les équations 3.12 et 3.13 respectivement, tel que λ étant le nombre de membre du CH, j .

$$EnergieConsommee_i = E_p^1 + ENN[i][j] \quad (3.12)$$

$$EnergieConsommee_j = \lambda * E_{ph}^1 + ENB[j] \quad (3.13)$$

Optimisation des durées des rounds

Dans cette phase nous adaptions le *PSO* (3.4.2 Particle swarm optimization) pour calculer les durées des rounds dans le but de maximiser la durée de vie du réseau. Pour répondre fidèlement au modèle proposé, la fonction objective de notre algorithme basé sur *PSO* est la même que celle proposée dans la modélisation présentée dans la section 3.4.1, qui est :

$$\max \sum_{k=1}^K t_k : \text{Maximiser la somme des durées des rounds .}$$

Tel que t_j est la durée du round j et k est le nombre total de rounds. En outre, la position d'une particule « i » représente les durées des rounds, comme c'est montré par la formule 3.14. T_{max} étant la durée *max* que peut avoir un round.

$$\langle T_i = T_1, T_2, \dots, T_k, T_j \in [0, T_{max}] \rangle \quad (3.14)$$

La conception de notre algorithme basé sur *PSO* peut être divisée en quatre étapes principales :

(a) Initialisation :

Nous considérons un certain nombre de population de particules. Les positions de chaque particule sont initialisées par des valeurs aléatoires. Ensuite chaque particule calcule son coût à partir de sa position (via la fonction objective présentée plus haut), et définit ce coût comme étant le meilleur coût local. Le meilleur coût global est le maximum des coûts locaux de toutes les particules.

(b) Calcul des vitesses et des positions :

La vitesse de chaque particule est calculée via la formule 3.8. Puis, à partir des vitesses, on met à jour les positions via la formule 3.9.

(c) Validation d'une solution :

Une position correspond à des durées de rounds et chaque round correspond à un schéma de clustering, obtenue dans la phase précédente. Pour la validation d'une solution, la contrainte 3 (équation 3.7) doit être vérifiée. Pour cela, l'énergie à consommer par chaque nœud pour satisfaire toutes les configurations obtenues (pendant toute la durée de vie du système) sera estimée en se basant sur le type de nœud dans le round multipliées par la durée du round, en utilisant les équations 3.12 et 3.13. Si l'énergie consommée par chaque nœud durant toute sa vie est inférieure ou égale à son énergie initiale, la position est validée, sinon elle est rejetée.

La validation d'une solution relie les deux phases : Obtention des configurations de clustering et optimisation des durées des rounds afin de garantir que seuls les bons schémas de clustering vont être sélectionnés, avec les durées de rounds adaptés qui permettent de prolonger la durée de vie. Cette phase permet également d'éliminer les mauvaises configurations de clustering en convergeant leur durées de round à zéro. Il est à noter que la taille des particules, qui représente la borne maximale du nombre de rounds K , affecte directement les performances de cet algorithme. Bien évidemment, une grande valeur de K donnera plus de flexibilité à l'algorithme et augmentera la chance de génération de bons schémas de clustering dans chaque particule. La valeur de K sera fixée empiriquement dans la phase des tests 4.4.

(d) Mise à jour des coûts :

Quand la position d'une particule est validée, on calcule son coût avec la fonction objective 3.4 :

- Si ce coût actuel est meilleur que le *Pbest* coût enregistré, on met à jour le *Pbest* coût de cette particule et la nouvelle position.
- Après la mise à jour de tous les *Pbest* coûts de toutes les particules de la population, on choisit le meilleur d'entre eux et on le compare avec le *Gbest-local* enregistré. S'il est supérieur, on met à jour le *Gbest-local* avec la nouvelle position.
- A la fin de toutes les itérations, si le *Gbest-local* est meilleur que le *Gbest-global*, on met à jour le *Gbest-global* avec la nouvelle position est la configuration correspondante.

L'organigramme de la figure 3.3 résume les étapes précédentes :

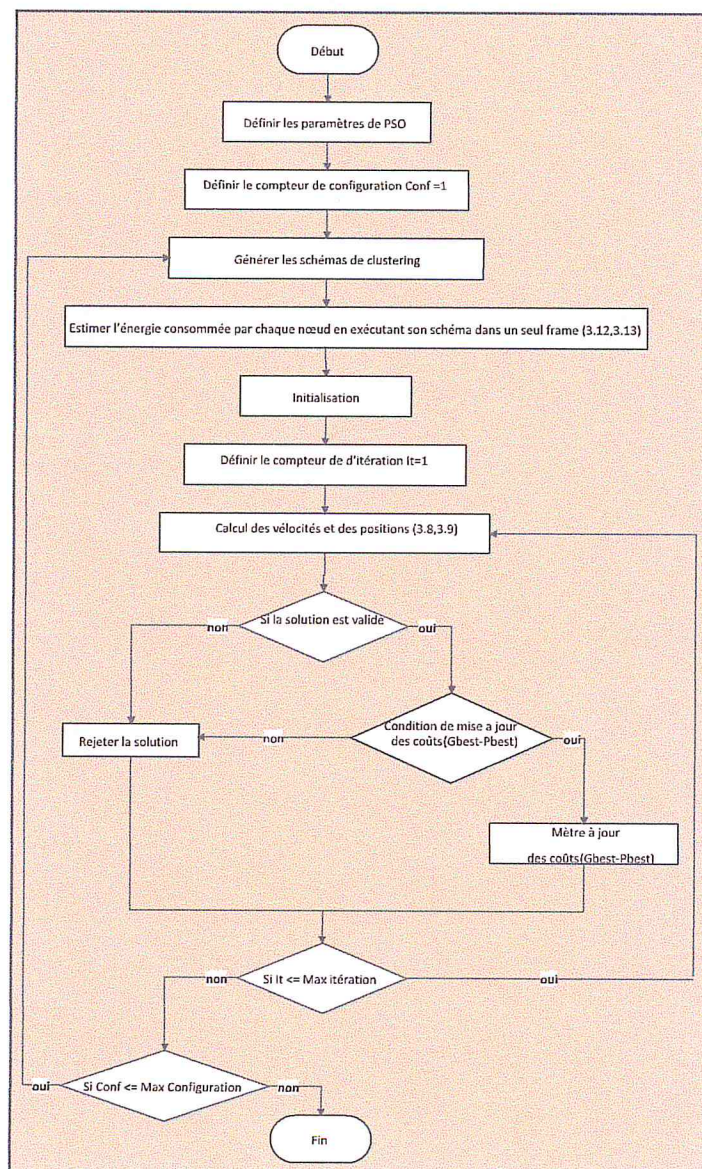


FIGURE 3.3 – Organigramme qui montre l'utilisation de *PSO* pour la Clusterisation.

3.5 Conclusion

Dans ce chapitre, une nouvelle modélisation du problème de clustering dans les réseaux de capteurs sans fil a été présentée. Ensuite, un nouvel algorithme de clustering centralisé a été proposé et conçu pour répondre fidèlement à cette modélisation. Le chapitre suivant sera consacré à la description de notre implémentation de l'algorithme proposé ainsi que sa comparaison avec d'autres protocoles d'état de l'art.

CHAPITRE 4

IMPLÉMENTATION ET TESTES

4.1 Introduction

Ce chapitre est consacré à l'évaluation des performances du protocole de clustering conçu "2OSC", ainsi que sa comparaison avec trois protocoles d'état de l'art qui sont : *LEACH-C*, *TPSO-CR* et *ETPSO-CR*. Les métriques de performances considérées dans la comparaison sont : la durée de vie de réseau, le nombre de nœuds vivants et la quantité de données délivrée à la *SB*.

Dans ce qui suit, nous allons présenter l'environnement expérimental que nous avons utilisé. Nous exposons également les étapes d'implémentation de notre protocole, et nous terminerons par la discussion des résultats de tests et des comparaisons.

4.2 Environnement expérimental

Afin de mettre en œuvre notre protocole, nous avons fait appel aux outils : *MATLAB* et *GenSeN*.

(a) MATLAB :

L'algorithme de clustering centralisé de protocole *2OSC* est basé sur la technique d'optimisation *PSO*. Comme nous l'avons déjà vu, *PSO* est un algorithme itératif, qui nécessite beaucoup de calcul et qui peut être long dans son exécution. Pour cela, nous avons besoin d'un calculateur puissant et rapide. Au début, nous avons pensé à utiliser un simulateur réseau tel que, NS3, Omnet, etc. Cependant, l'utilisation d'un simulateur réseau n'est pas vraiment significative dans notre contexte. En effet, notre objectif est d'évaluer l'efficacité d'un algorithme de clustering centralisé, et non pas d'un protocole de transmission de données. C'est pour cette raison que le choix a été porté pour *MATLAB*. Ce dernier est un système interactif et convivial de calcul numérique et de visualisation graphique, destiné aux ingénieurs, aux techniciens et aux scientifiques. C'est un outil très utilisé, dans les universités comme dans le monde industriel, qui intègre des centaines de fonctions mathématiques et d'analyse numérique [24].

D'autre part, même lors d'un déploiement réel de notre protocole, l'algorithme de clustering centralisé sera exécuté par la station de base, qui est supposée être un nœud puissant, sans contrainte d'énergie. Ceci dit qu'elle peut exécuter un code *MATLAB* aisément.

(b) "A Generator for Sensor Networks" GenSeN :

Afin de travailler sur une topologie de réseau de capteurs plus ou moins réelle, nous avons utilisé l'outil GenSeN, qui est un générateur de topologies conçu pour les réseaux de

capteurs sans fils. GenSeN prend en entrée le nombre de nœuds, la dimension de champ de captage et la stratégie de déploiement du réseau (aléatoire, en grille, etc.). Il génère en sortie la topologie sous forme d'un fichier texte qui contient les positions (X, Y) de chaque nœud capteur [25].

4.3 Implémentation

D'abord nous avons utilisé l'outil *GenSeN* pour générer une topologie réaliste d'un *RCSF*. Cette topologie suit une certaine stratégie de déploiement pour un certain nombre de nœuds. Par la suite, nous avons implémenté le procédé de clustering sous *MATLAB*, qui va jouer le rôle de la station de base pour générer les schémas de clustering et fixer les durées des rounds en se basant sur l'estimation de la consommation d'énergie. L'architecture de notre implémentation est présentée dans la figure 4.1. Comme illustré dans cette figure, la fonction "Main" est la fonction principale de notre programme, qui permet de lancer l'exécution du protocole. Cette fonction fait appel aux modules suivants :

- Module d'initialisation : dans ce module nous avons importé le fichier de configuration, calculé les distances entre les nœuds et les distances entre les nœuds et la station de base et nous avons également calculé l'énergie nécessaire pour envoyer un paquet d'un nœud à un autre et d'un nœud vers la station de base.
- Module des schémas de clustering : nous avons implémenté ce module pour générer les schémas de clustering.
- Module d'estimation d'énergie : c'est le module principal de la fonction "Main", car tout se base sur les estimations d'énergie qui sont faites à partir d'un modèle radio que nous avons implémenté.
- Module des durées de rounds : ce module est chargé de déterminer la durée de chaque round en faisant appel à la fonction objective.
- Module de vivacité du réseau : ce module détermine si le réseau est mort ou pas, selon le critère fixé (FND par exemple).

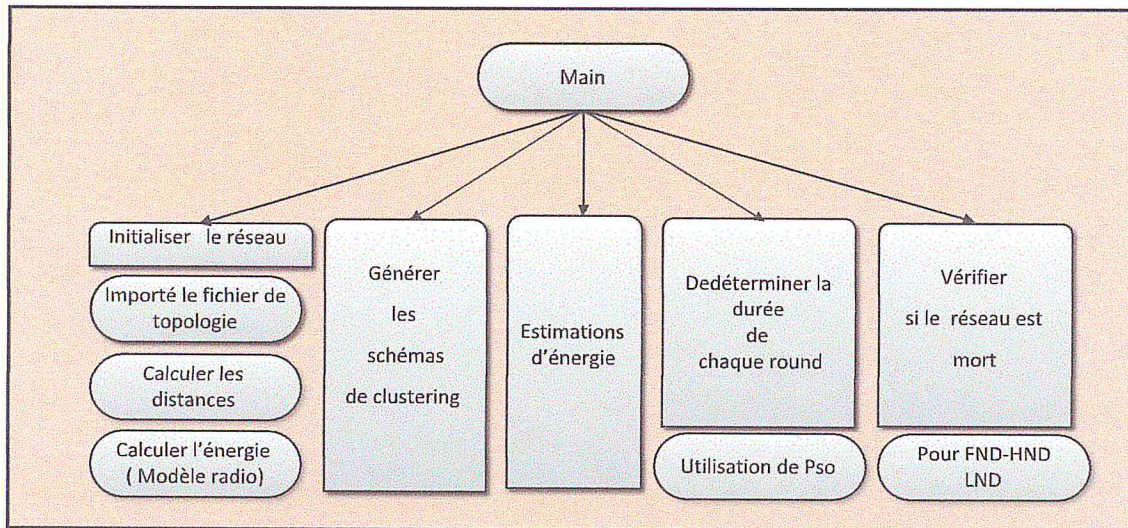


FIGURE 4.1 – Architecture d'implémentation du protocole proposé.

4.4 Tests

4.4.1 Configuration de scénario de tests.

Nous avons fixé certains paramètres du *RCSF*, afin d'évaluer notre protocole et les autres protocoles sous le même environnement. Le tableau 4.1 résume les paramètres utilisés ainsi que leurs valeurs.

| Paramètres | Valeurs |
|----------------------------------|-------------------------|
| Nombre de nœuds | 100 200 300 |
| Zone de couverture | 100 × 100 |
| Cordonnées de la station de base | (50,50) |
| Stratégie de déploiement | RANDOM |
| Énergie initiale | 0.25 Joules |
| Modèle énergétique | First Order Radio Model |

TABLE 4.1 – Paramètres du réseau de capteurs

D'autre part, nous avons implémenté le modèle énergétique "First Order Radio Model" sous *MATLAB* pour l'estimation des énergies résiduelles des micro-capteurs. Les paramètres du modèle radio implémenté sont présentés dans le tableau 4.2.

| Facteur | Valeurs |
|------------|----------------------------|
| E_{elec} | 50 nJ/bit |
| E_{fs} | 5 nJ/bit |
| E_{mp} | 10pJ/(bit.m ²) |
| l | 800 bit |
| E_{DA} | 5 nJ/bit |

TABLE 4.2 – Paramètre du modèle énergétique

En ce qui concerne les paramètres de la méta-heuristique *PSO*, qui sont : le coefficient d'inertie, les coefficients d'accélération, le nombre de population et le nombre d'itération ont été fixés comme indiqué dans la référence [27]. Ainsi, d'autres paramètres liés à notre conception de l'algorithme de clustering ont été fixés empiriquement après une série importante de tests, afin de trouver les valeurs optimales qui maximise la durée de vie. Ces paramètres sont principalement, les bornes maximales de nombre de rounds et de la durée de chaque round et le nombre de configurations. Les valeurs de tous ces paramètres sont présentées dans le tableau 4.3 :

| Paramètres | Valeurs |
|-------------------------------|---------|
| W | 1 |
| C_1 | 2 |
| C_2 | 2 |
| Population | 100 |
| Itération | 1000 |
| Configurations | 20 |
| Nombre maximale de rounds | 600 |
| Durée maximal de chaque round | 250 |

TABLE 4.3 – Paramètres de *PSO*

4.4.2 Résultats

Dans ce qui suit, nous allons évaluer les performances de notre protocole *2OSC*, et nous allons comparer ses résultats de simulation avec trois protocoles qui sont : *LEACH-C*, *TPSO-CR* et *ETPSO-CR*. Les trois protocoles sont bien présentés dans le deuxième chapitre. Tous

les protocoles choisis appartiennent à la catégorie des protocoles de clustering centralisés, tout comme la solution proposée. *LEACH-C* est un travail pionnier qui représente la référence des protocoles de clustering centralisé, d'où son choix. *TPSO-CR*, publié en 2015, utilise la même méta-heuristique que nous avons utilisé dans notre travail, à savoir *PSO*. Donc son choix est pour prouver que l'efficacité de notre solution n'est pas due seulement à la méta-heuristique *PSO*.

La comparaison se fait selon plusieurs critères qui sont :

- La durée de vie du réseau : La durée de vie d'un réseau est un critère majeur pour l'évaluation des performances de notre protocole, comme c'est l'objectif principal de notre travail. Pour examiner la durée de vie du réseau, nous avons opté à l'utilisation des trois critères suivants :
 - **First Node Dies (FND)** : c'est la période qui sépare le déploiement du réseau et la mort du premier nœud.
 - **Half Node Die (HND)** : c'est la période qui sépare le déploiement du réseau et la mort de la moitié des nœuds du réseau.
 - **Last Node Dies (LND)** : c'est la période qui sépare le déploiement du réseau et la mort du dernier nœud.
- Le nombre de nœuds vivants depuis le déploiement du réseau et jusqu'à la mort de dernier nœud.
- Le taux de délivrance des données au niveau de la station de base qui est le nombre de paquets reçus par la station de base.

Tous les tests de cette section sont évalués avec une densité de nœuds variée entre 100, 200 et 300 en gardant toujours la même surface de déploiement ($100m \times 100m$). De plus, les tests ont été réalisés dans les deux types de réseaux suivants :

- **Réseau homogènes** : Nous avons considéré l'homogénéité par rapport à l'énergie initiale. Autrement dit, un réseau est dit homogène lorsque tous les nœuds du réseau ont la même quantité d'énergie initiale.
- **Réseau hétérogènes** : Dans l'opposé, un réseau est dit hétérogène si les nœuds-capteurs ont des valeurs différentes d'énergies initiales. Dans nos tests, nous avons rangé les valeurs énergétiques initiales des micro-capteurs dans l'intervalle $[0.1, 0.25]$.

Durée de vie du réseau :

(a) Réseaux homogènes :

La figure 4.2 représente l'histogramme des durées de vie de *2OSC* et les autres protocoles dans un réseau homogène .

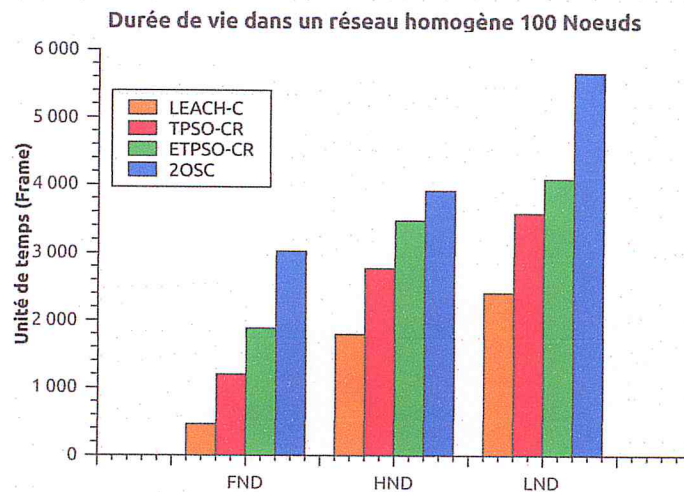


FIGURE 4.2 – Durée de vie d'un réseau homogène (100 Nœuds)

Nous remarquons d'après la figure 4.2 que notre protocole améliore largement la durée de vie du réseau par rapport aux autres protocoles. En ce qui concerne le FND, notre protocole surpasse le protocole *LEACH-C* de 84.79%, le protocole *TPSO-CR* de 60.39% et le protocole *ETPSO-CR* de 37.78% cette prolongation de la durée de vie du réseau est en fait le résultat absolu de la minimisation de la consommation d'énergie au niveau de tous les nœuds du réseau. Car nous avons éliminé le cout de re-clustering et nous avons amélioré la qualité de clustering. De plus, nous avons appliqué la durée de round dynamique, et donc la durée de vie va être prolongée encore plus. En passant à l'échelle et en augmentant le nombre de nœuds vers 200 et 300 nœuds, on remarque que notre protocole garde toujours les mêmes performances, cela est montré dans les figures 4.3 ,4.4.

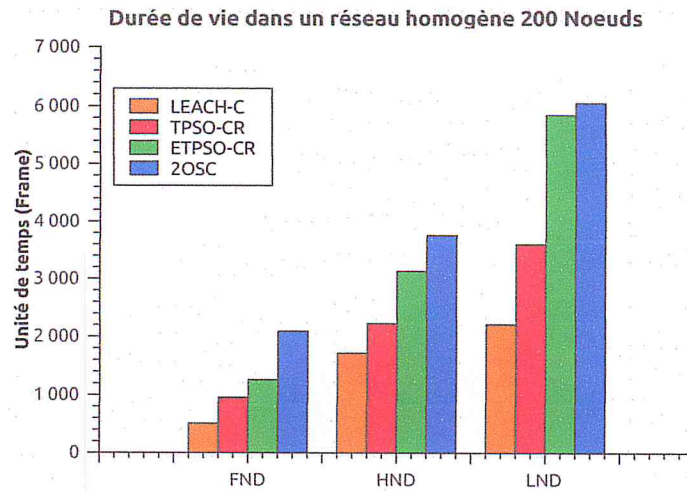


FIGURE 4.3 – Durée de vie d’un réseau homogène de 200 Nœuds.

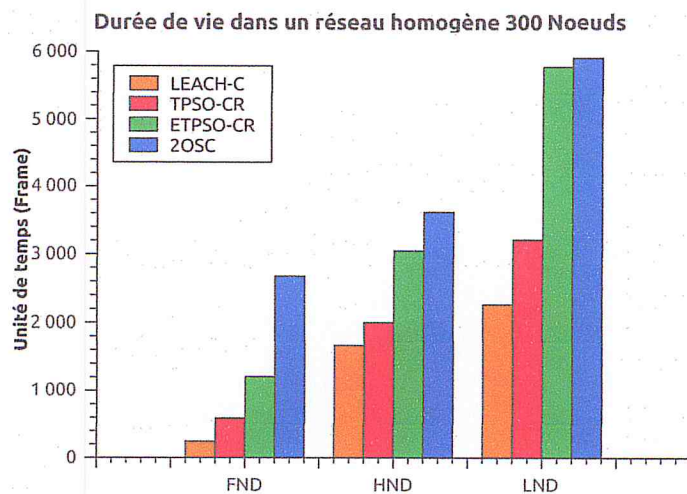


FIGURE 4.4 – Durée de vie d’un réseau homogène de 300 Nœuds.

(b) Réseaux hétérogènes :

On constate que notre protocole supporte aussi l’hétérogénéité, et il donne toujours de meilleurs résultats par rapport aux autres protocoles.

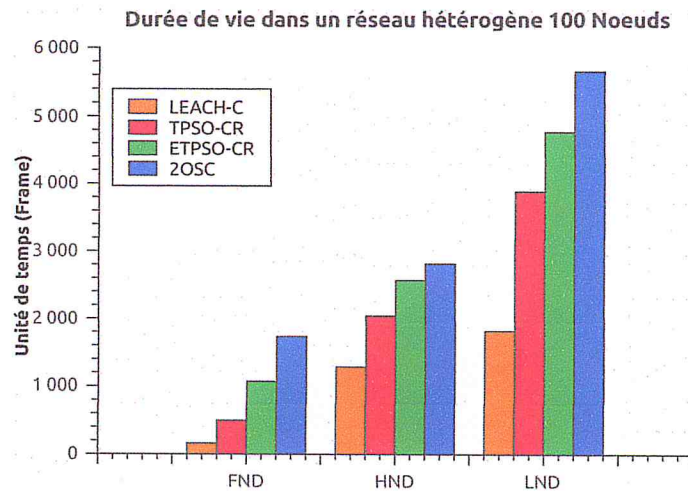


FIGURE 4.5 – Durée de vie d'un réseau hétérogène de 100 Nœuds.

Pour les réseaux hétérogènes qui disposent d'une densité de nœuds élevée (200, 300 nœuds), notre protocole a également assuré une longue durée de vie, comme montré dans les figures 4.6 ,4.7.

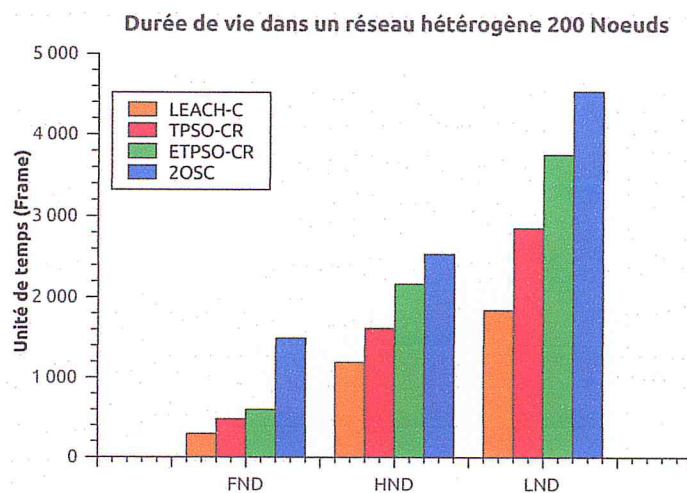


FIGURE 4.6 – Durée de vie d'un réseau hétérogène de 200 Nœuds.

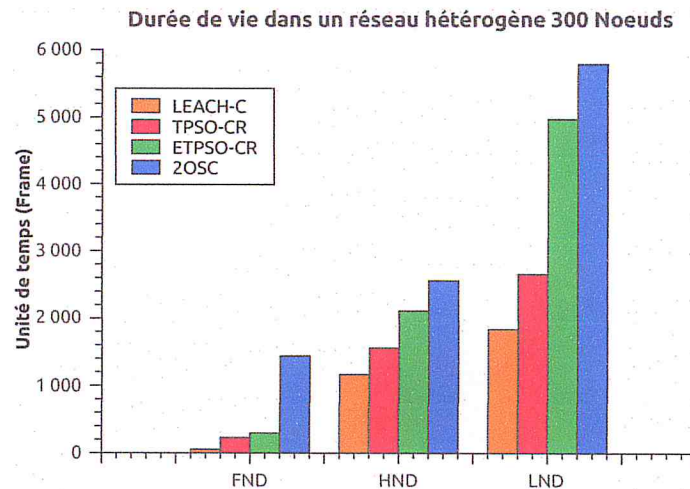


FIGURE 4.7 – Durée de vie d'un réseau hétérogène de 300 Nœuds.

Nombre de nœuds vivants :

- (a) **Réseaux homogènes** : La figure 4.8 représente le nombre de nœuds vivants dans un réseau homogène à travers le temps (frames).

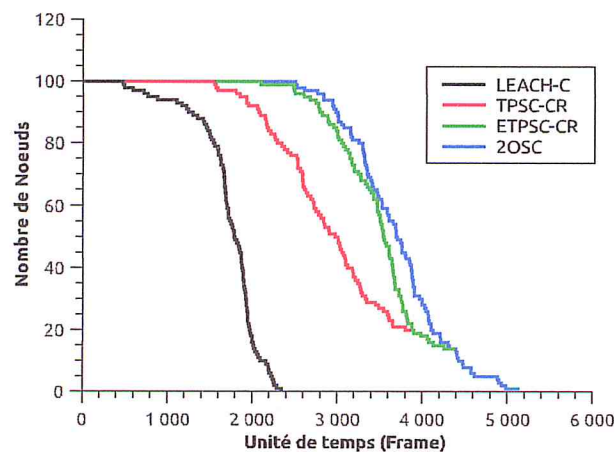


FIGURE 4.8 – Les nœuds vivants dans un réseau homogène de 100 nœuds.

Nous remarquons que les nœuds meurent vite dans le protocole *LEACH-C*, le premier nœud meurt à la 508ème frame et le dernier nœud au 2385ème frame, tandis que dans notre protocole le premier nœud meurt à la 2844ème frame après que le dernier nœud du protocole *LEACH-C* meurt, et le dernier au 5180ème frame. Ceci est toujours grâce à l'élimination du procédé de re-clustering, et aux durées dynamiques des rounds.

Le nombre de nœuds vivants dans notre protocole reste toujours plus grand par rapport aux autres protocoles (*LEACH-CH*, *TPSO-CR* et *ETPSO-CR*) même en augmentant le

nombre de nœuds dans le réseau à 200 et 300 nœuds, comme le montre les figures 4.9 et 4.10, respectivement.

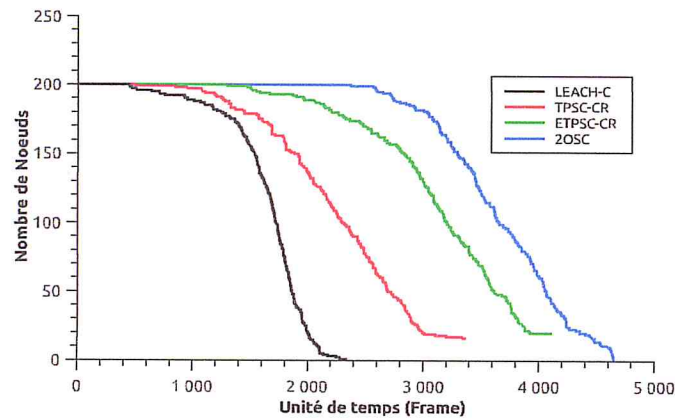


FIGURE 4.9 – Les nœuds vivants dans un réseau homogène de 200 nœuds.

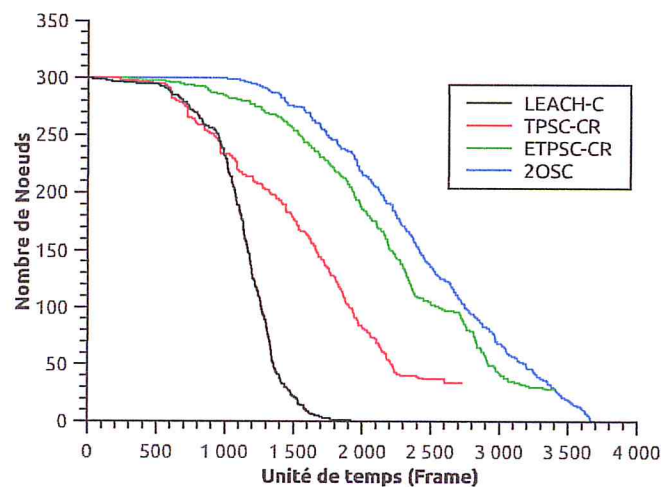


FIGURE 4.10 – Les nœuds vivants dans un réseau homogène de 300 nœuds.

(b) **Réseaux hétérogènes :**

En ce qui concerne le nombre de nœuds vivants, nous remarquons que les résultats de tous les protocoles dans des réseaux hétérogènes se sont dégradés comparant aux résultats obtenus dans des réseaux homogènes. Ceci car les valeurs énergétiques des micro-capteurs ont été générées aléatoirement dans l'intervalle $[0.1, 0.25]$, donc on peut avoir des micro-capteurs avec une très faible énergie initiale. Cependant, notre protocole surpasse toujours les autres protocoles quel que soit la densité du réseau, (voir la figure 4.11).

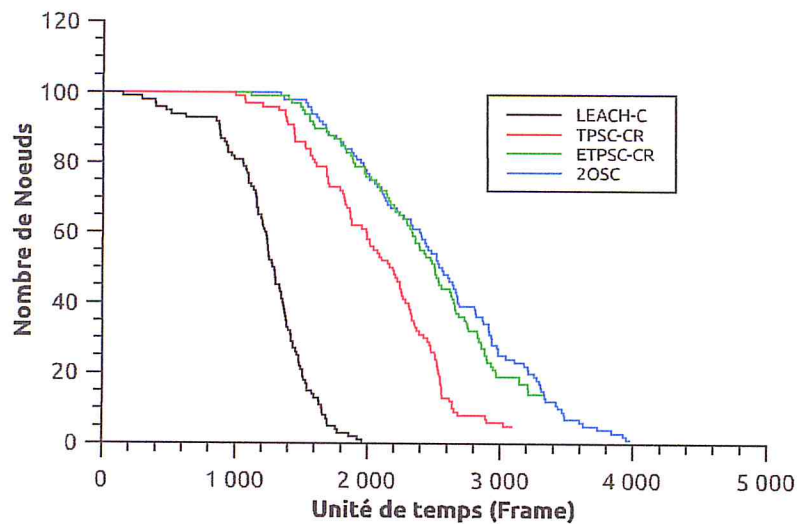


FIGURE 4.11 – Les nœuds vivants dans un réseau hétérogène de 100 nœuds.

Même commentaires est valable pour un passage à l'échelle avec 200 et 300 nœuds 4.12 ,4.13.

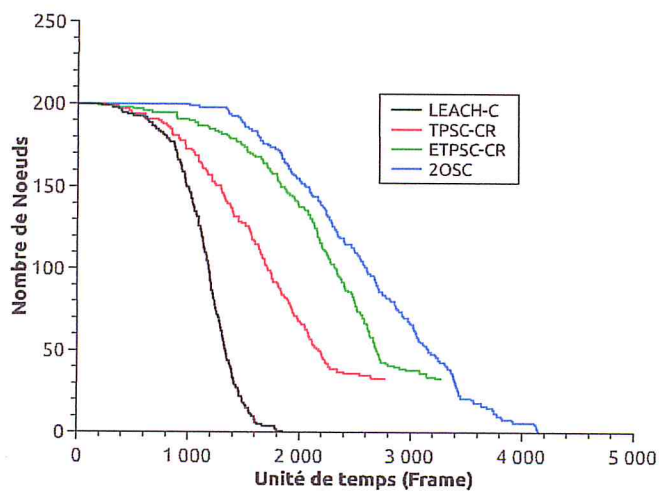


FIGURE 4.12 – Les nœuds vivants dans un réseau hétérogène de 200 nœuds.

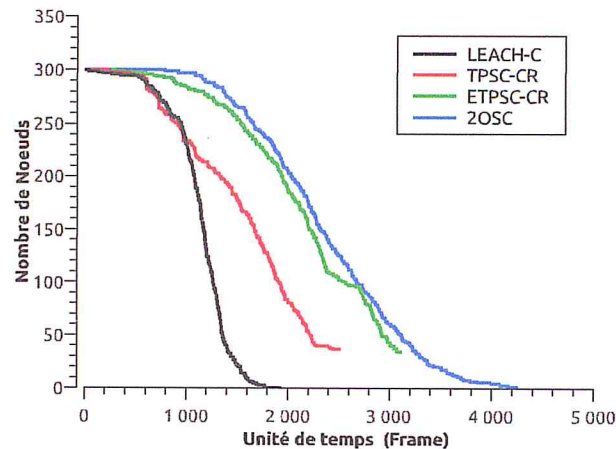


FIGURE 4.13 – Les nœuds vivants dans un réseau hétérogène de 300 nœuds.

Taux de délivrance de données :

(a) Réseaux homogènes :

La figure 4.14 représente le nombre de paquets de données reçus par la station de base tout au long de la durée de vie du réseau dans les quatre protocoles, et ceci pour des réseaux de 100, 200 et 300 nœuds.

Nous remarquons que le taux de délivrance dans notre protocole est meilleur que celui des protocoles *LEACH-C*, *TPSO-CR* et *ETPSO-CR*. Notre protocole arrive en moyenne à délivrer plus de 88.30% de données à la *SB* par rapport aux autres protocoles.

Ces performances sont obtenus grâce à la prolongation de la durée de vie du réseau d'une part et à la dynamique du nombre de *CHs* élus dans le réseau d'autre part. En effet, le nombre de *CHs* élus dans chaque round dans les protocoles *LEACH-C*, *TPSO-CR* et *ETPSO-CR* est fixé à 5% de nombre total des nœuds dans le réseau, alors que ce nombre est totalement dynamique et adaptatif aux besoins du réseau dans notre protocole. Bien évidemment, plus le nombre de *CHs* est élevé dans le réseau, plus le taux de délivrance de données sera amélioré, vu que les *CHs* sont les seuls nœuds qui se chargent de la transmission de données vers la station de base.

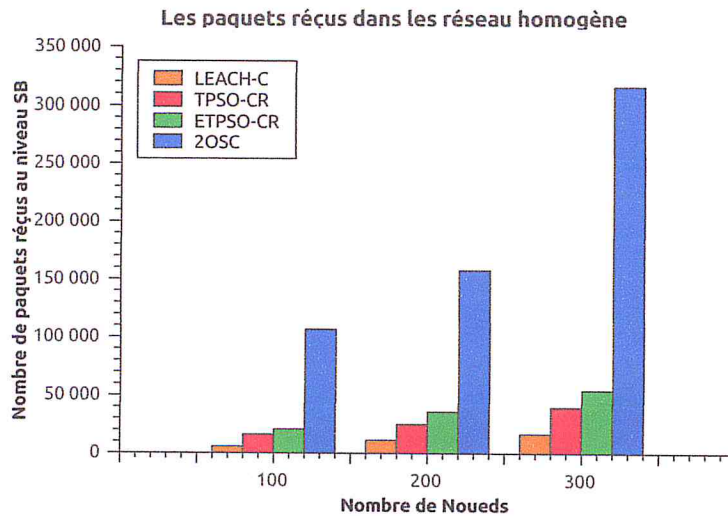


FIGURE 4.14 – Le taux de paquets reçus au niveau de la *SB* dans des réseaux homogènes.

(b) Réseaux hétérogènes :

La figure 4.15 confirme l'efficacité de notre protocole en ce qui concerne le taux de délivrance de données même dans des réseaux hétérogènes.

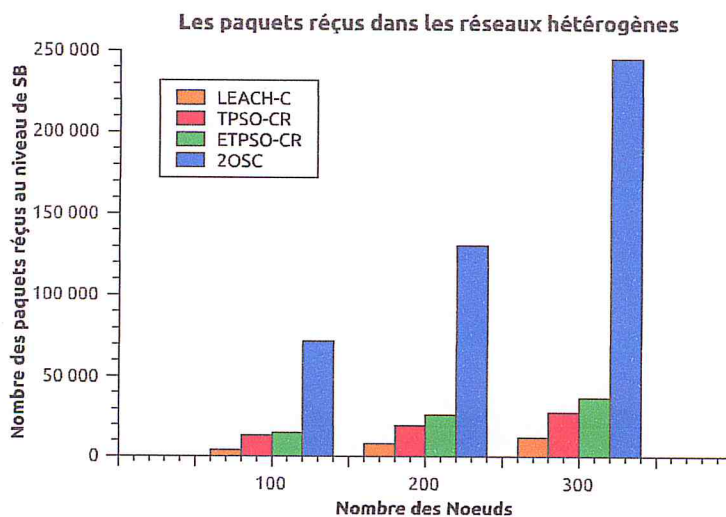


FIGURE 4.15 – Le taux de paquets reçus au niveau de la *SB* dans des réseaux hétérogènes.

Discussion :

Nous pouvons conclure que pour les trois métriques d'évaluation présentées précédemment, notre protocole a de meilleures performances par rapport aux protocoles étudiés, et cela pour les deux types de réseaux considérés : homogène et hétérogène. Les deux tableaux 4.4 et 4.5 présentent les pourcentages d'amélioration de *2OSC* par rapport aux protocoles *LEACH-C*, *TPSO-CR* et *ETPSO-CR* dans le cas des réseaux homogènes et hétérogènes, respectivement.

| | Protocole | <i>FND</i> | <i>HND</i> | <i>LND</i> | DATA |
|-----------|-----------------|------------|------------|------------|--------|
| 100 nœuds | <i>LEACH-C</i> | 84.79% | 69.50% | 57.45% | 94.72% |
| | <i>TPSO-CR</i> | 60.39% | 29.16% | 36.70% | 84.86% |
| | <i>ETPSO-CR</i> | 37.78% | 11.17% | 27.81% | 80.69% |
| 200 nœuds | <i>LEACH-C</i> | 82.50 | 54.31% | 66.70% | 92.87% |
| | <i>TPSO-CR</i> | 66.90% | 40.79% | 40.38% | 84.09% |
| | <i>ETPSO-CR</i> | 56.54% | 16.74% | 3.37% | 77.29% |
| 300 nœuds | <i>LEACH-C</i> | 90.94% | 53.96% | 61.76% | 94.70% |
| | <i>TPSO-CR</i> | 77.92% | 44.76% | 45.73% | 87.38% |
| | <i>ETPSO-CR</i> | 54.88% | 15.86% | 2.33% | 82.64% |

TABLE 4.4 – Tableau d'amélioration dans un réseau homogène

| | Protocole | <i>FND</i> | <i>HND</i> | <i>LND</i> | DATA |
|-----------|-----------------|------------|------------|------------|--------|
| 100 nœuds | <i>LEACH-C</i> | 90,72% | 54.36% | 67.83% | 94.38% |
| | <i>TPSO-CR</i> | 71.34% | 27.43% | 31.30% | 81.60% |
| | <i>ETPSO-CR</i> | 38.18% | 8.80% | 15.68% | 79.50% |
| 200 nœuds | <i>LEACH-C</i> | 79.78% | 53.08% | 59.47% | 93.96% |
| | <i>TPSO-CR</i> | 66.64% | 36.32% | 37.14% | 84.96% |
| | <i>ETPSO-CR</i> | 58.23% | 14.63% | 17.28% | 79.79% |
| 300 nœuds | <i>LEACH-C</i> | 93.55% | 54.56% | 68.23% | 95.17% |
| | <i>TPSO-CR</i> | 83.97% | 39.12% | 54.02% | 88.64% |
| | <i>ETPSO-CR</i> | 79.39% | 17.59% | 14.10% | 84.90% |

TABLE 4.5 – Tableau d'amélioration dans un réseau hétérogène

Cette amélioration est due principalement à :

1. L'élimination de re-clustering : en éliminant le cout de re-clustering au début de chaque round, nous avons pu diminuer la quantité d'énergie dissipée au niveau de chaque nœud. Conséquemment, la durée de vie du réseau a été prolongée.
2. La durée dynamique des rounds : La durée dynamique des rounds apporte aussi un gain énergétique considérable, et minimise la mort prématurée des *CHs*, ce qui permet de

prolonger la durée de vie du réseau. La figure 4.16 présente la variation dans la durée de round dans 20SC.

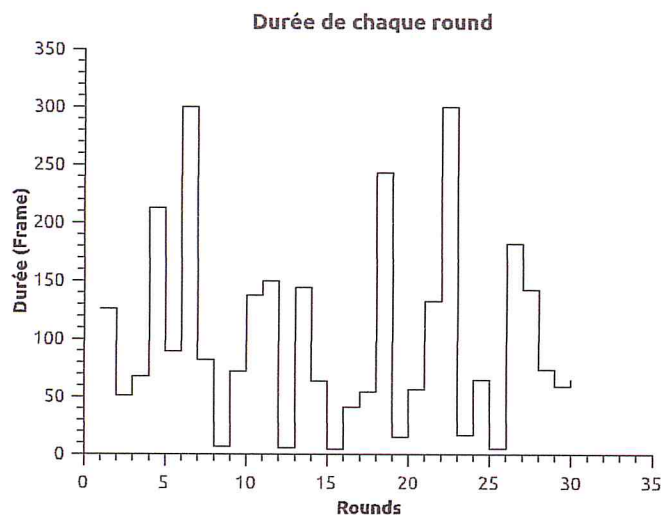


FIGURE 4.16 – Nombre des *CHs* dans chaque round

3. La dynamicité des facteurs de clustering : Les facteurs influençant l'efficacité énergétique des protocoles de clustering sont adaptatives, et aucun facteur n'est fixé au préalable. Cela permet d'attribuer les valeurs qui optimisent la durée de vie du système à tous les facteurs : nombre de round, durée de chaque round, nombre de *CHs* par round, taille des clusters par rounds, les nœuds à élire comme *CHs*. La figure 4.17 représente la dynamicité de nombre de *CHs* par round.

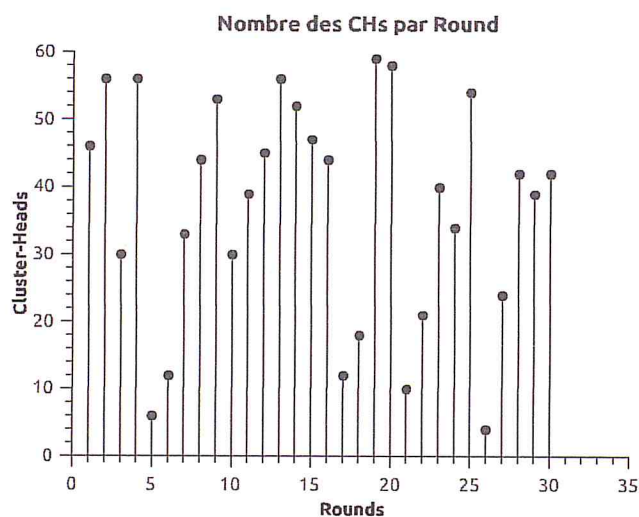


FIGURE 4.17 – Variation des durées de rounds

4.5 Conclusion :

Dans ce chapitre, nous avons présenté les outils utilisés pour la mise en œuvre de notre protocole. Par la suite, nous avons comparé les performances de notre protocole *2OSC* avec trois protocoles d'état de l'art : *LEACH-C*, *TPSO-CR* et *ETPSO-CR*, et cela selon plusieurs critères d'évaluation des *RCSFs*.

Suite à une série de tests, nous nous sommes arrivés à des résultats très satisfaisants vu que notre protocole surpasse nettement les autres protocoles étudiés, même dans les réseaux à forte densité et les réseaux hétérogènes.

CONCLUSION GÉNÉRALE

Les *RCSFs* sont caractérisés par de modestes ressources matérielles, notamment celle énergétique, ce qui constitue leur principal handicap. D'autre part, plusieurs applications des *RCSFs* nécessitent le déploiement d'un grand nombre de micro-capteurs dispersés aléatoirement autour du terrain d'intérêt du phénomène capté. Les contraintes matérielles des *RCSFs* combinées à la densité de déploiement compliquent davantage les tâches ordinaires des *RCSFs*, telles que le routage des données, et rendent la prolongation de la durée de vie du réseau un challenge réel. La clusterisation d'un *RCSFs* s'avère comme une technique efficace permettant l'optimisation de la consommation d'énergie des nœuds capteurs, et donc la prolongation de la durée de vie des réseaux de capteurs sans fil, surtout ceux déployés à grande échelle. Dans cette optique, notre projet de fin d'études consiste à la conception et l'évaluation d'un nouvel algorithme de clustering.

Pour la réalisation de notre projet, nous avons commencé par une étude sur les caractéristiques des réseaux de capteurs, leurs architectures, et surtout leurs défis de conception. Par la suite, nous avons présenté les concepts de base de clustering et nous avons exposé quelques protocoles de clustering proposés spécifiquement pour les réseaux de capteurs sans fil. D'après l'analyse faite, nous avons constaté que la qualité de la hiérarchie dans les protocoles de clustering centralisés est de loin meilleure comparant à celle produite par les protocoles distribués. Ainsi, nous avons remarqué que le re-clustering au début de chaque round et la durée fixe des rounds sont des inconvénients majeurs et communs pour la majorité des protocoles de clustering.

Pour remédier à ces problèmes, nous avons conçu un nouvel algorithme de clustering centralisé nommé *2OSC* "Optimized One-Step Clustering" qui élimine le coût du re-clustering tout en maintenant ses avantages, et qui permet de générer des durées des rounds dynamiques

et adaptatives aux énergies résiduelles des différents nœuds dans le réseau. *2OSC* est basé sur la méta-heuristique *particle swarm optimization (PSO)* qui est, selon des travaux existants, la plus adéquate au problème de clustering. A la fin, nous avons mené des simulations extensives afin de comparer les performances du protocole *2OSC* avec *LEACH-C*, *TPSO-CR* et *ETPSO-CR*. L'évaluation s'est faite selon plusieurs métriques et sur diverses densités. Dans tous les scénarios de tests, les résultats montrent clairement que le protocole proposé surpasse les autres protocoles. Dans certains scénarios, en comparant avec le protocole *ETPSO-CR* (qui a les meilleures performances par rapport aux protocoles *LEACH-C* et *TPSO-CR*), l'amélioration a atteint 79,39% par rapport à la durée de vie *FND*, 17,59% pour *HND*, 14,10% pour *LND* et 84,90% en ce qui concerne le taux de paquets reçus dans la station de base pour un réseau hétérogène avec 300 nœuds.

Le travail que nous avons effectué dans ce projet de fin d'étude nous ouvre de nombreuses perspectives que nous avons structurées comme suit :

- La perspective principale de ce travail consiste à la proposition d'une version qui assure la tolérance aux pannes des micro-capteurs. En effet, dans la solution que nous avons proposée, aucune communication n'est prévue entre les nœuds et la station base après l'initialisation du réseau. Conséquemment, les pannes des nœuds ne sont prises en considération. Autrement dit, le protocole proposé peut assigner le rôle important de CH aux nœuds qui sont devenus hors service. Une technique qui détecte les nœuds qui sont tombés en pannes dans le réseau doit être implémentée pour remédier à ce problème.
- L'utilisation d'un modèle d'estimation énergétique plus réaliste qui prend en considération l'écoute passive (*idle listening*), la retransmission en cas de collisions, etc. s'avère comme perspective intéressante.

BIBLIOGRAPHIE

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam et E. Cayirci : Wireless sensor networks : a survey. *Computer Networks : The International Journal of Computer and Telecommunications Networking*, 38(4) :393-422, 2002.
- [2] Kris Lin, Jennifer Yu, Jason Hsu, Sadaf Zahedi, David Lee, Jonathan Friedman, Aman Kansal, Vijay Raghunathan, and Mani Srivastava. Heliomote :”enabling long-lived sensor networks through solar energy harvesting”. :In *SenSys Proceedings of the 3rd international Conference on Embedded networked sensor systems*” p.309,309 New York, NY, USA
- [3] H. Karl et A. Willig : *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2007.
- [4] Mohammad Mahfuzul Islam Akramul Azim1. “A Dynamic Round-Time Based Fixed Low Energy Adaptive Clustering Hierarchy for Wireless Sensor Networks”. In : *Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications*. IEEE. 2009, p.922–926.
- [5] Audithan S Karthikeyan T. “An Enhanced Adaptive Re-Clustering Protocol in Wireless Sensor Network”. In : *2nd International Conference on Current Trends in Engineering and Technology, ICCTET’14*. IEEE. 2014, p. 418–422.
- [6] Fuad Bajaber , Irfan Awan. “Adaptive decentralized re-clustering protocol for wireless sensor networks”. In : *Journal of Computer and System Sciences*, 77 . 2011, p. 282-292.

- [7] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS-33), January 2000.
- [8] W.R. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1, no.4 :660-670, October 2002.
- [9] Ravneet Kaur, Deepika Sharma and Navdeep Kaur, Comparative Analysis Of Leach And Its Descendant Protocols In Wireless Sensor Network, *International Journal of P2P Network Trends and Technology*, Vol. 3, Issue 1, 2013, pp. 51-55.
- [10] Ali Norouzi1, Abdul Halim Zaim, An Integrative Comparison of Energy Efficient Routing Protocols in Wireless Sensor Network, *Scientific Research of Wireless Sensor Network*, Vol. 4, 2012, pp. 65-67.
- [11] W. Heinzelman, Chandrakasan, A. P., and Balakrishnan, H. (2002), "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, 1(4) pp. 660-670.
- [12] J. A.Braman and Umapathi G.R, "A Comparative Study on Advances in LEACH Routing Protocol for Wireless Sensor Networks : A survey," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, Vol. 3, Issue 2, pp .5683-5690, February 2014.
- [13] W.Heinzelman, A.Chandrakasan, and H.Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transaction on Wireless Communications*, vol. 1, no. 4, pp. 660-670, 2002.
- [14] J.Gnanambigai, Dr.N.Rengarajan, and K.Anbukkarasi, "Leach and Its Descendant Protocols : A Survey," *International Journal of Communication and Computer Technologies (IJCCT)*, vol. 01, issue 02, no.3, pp. 15-21, September 2012.
- [15] [CLERC et SIARRY 2003] Une nouvelle métaheuristique pour l'optimisation difficile : la méthode des essais particuliers, 2003
- [16] Riham.S.Y.Elhabyan and Mustapha.C.E.Yagoub. « Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network », *Journal of network and computer science*. Vol. 52, June 2015, pp 116-128.

- [17] KHENIFAR Afra CHEBBA Asmaa. "Propositions de techniques tolérantes aux pannes dans les réseaux de capteurs sans fils". Mém.de mast.
- [18] H. Balakrishnan W. Heinzelman J. Kulik. "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks". In : 5th ACM/IEEE Mobicom Conference. IEEE. 1999, p. 85–184.
- [19] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, Application specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Networking (2002).
- [20] E. Ilker Oyman, Cem Ersoy, Multiple sink network design problem in large scale wireless sensor networks, in : Proceedings of the IEEE International Conference on Communications (ICC 2004), Paris, June 2004.
- [21] M. Younis, M. Youssef, K. Arisha, Energy-aware management in cluster-based sensor networks, Computer Networks 43 (5) (2003), pp.649–668.
- [22] Bayan S Sharif NM Latiff Charalampos C Tsimenidis. "Energy-aware clustering for wireless sensor networks using particle swarm optimization". In : Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on. IEEE, 2007.
- [23]] Sudha Selvaraj Hemavathi Natarajan. "A Fuzzy Based Predictive Cluster Head Selection Scheme for Wireless Sensor Networks". In : Proceedings of the 8th International Conference on Sensing Technology. 2014, p. 560–566.
- [24] Introduction au logiciel Matlab. 2005. url : <http://www.ann.jussieu.fr/postel/matlab/node2.html> (visité le 01/06/2016).
- [25] A.Rodrigues F.Boavida T.Camilo J.Sá Silva. "GENSEN : A Topology Generator for Real Wireles Sensor Networks Deployment". In : IFIP International Federation for Information Processing, 2007, p. 436–445.
- [26] Riham.S.Y.Elhabyan and Mustapha.C.E.Yagoub. « Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network », Journal of network and computer sciencce. Vol. 52, Junc 2015, pp 116-128.

- [27] Zambrano-Bigiarini M, Clerc M, Rojas R. Standard particles swarm optimisation 2011 at cec-2013 : a baseline for future improvements. In : IEEE congress on evolutionary computation (CEC) ; June 2013. p. 2337-44.

