

الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne Démocratique et Populaire

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Centre de Développement  
des  
Technologies Avancées



Université Saad Dahleb de BLIDA

Faculté des Sciences

Département d'informatique

**MEMOIRE**

Présenté pour l'obtention d'un diplôme de

**MASTER EN INFORMATIQUE**

**THEME**

**Etude, implémentation et tests d'un protocole de  
clustering dans les réseaux de capteurs sans fil**

**Présenté par :**

BOUMESSAIDIA Billel.

BOUTAHRAOUI Abdelghani.

**Promoteur :**

Dr. Djamel DJENOURI.

**Encadreur :**

Mme. Nassima MERABTINE.

# Remerciements

*Nous tenons à remercier Dieu le tout puissant qui nous a donné le courage et la patience et qui a éclairé notre chemin pour achever ce travail.*

*Nous tenons à remercier en premier lieu le Dr. Djamel DJENOURI pour nous avoir proposé le thème de notre projet de fin d'étude. Nous serons toujours reconnaissants de ses interventions pour approuver les résultats de ce travail.*

*Nos vifs remerciements accompagnés de toute notre gratitude vont aussi à notre encadreur Mme. Nassima MERABTINE pour nous avoir consacré un temps précieux et accompagné avec un sérieux exemplaire tout au long de ce stage et également pour ses conseils, sa disponibilité et son aide indéfectible qui ont enrichi notre modeste expérience et permis de mener à bien ce projet.*

*Nous remercions enfin nos familles respectives (nos chers parents, frères et sœurs, cousin et cousines) pour leur soutien et encouragements très appréciés ainsi que nos amis et collègues avec qui nous avons partagé des moments inoubliables pendant notre cursus universitaire et pour lesquels nous garderons de merveilleux souvenirs gravés à jamais dans nos mémoires.*

# Dédicaces

*Premièrement je remercie DIEU de nous avoir donné le courage et la patience sans lesquels nous n'aurons jamais réussi à élaborer ce mémoire.*

*A mon père et ma chère mère, qui m'ont toujours soutenu et qui croient toujours en moi : toutes les lettres ne sauraient trouver les mots pour vous exprimer mon amour, ma gratitude et reconnaissance....*

*A la mémoire de mon cher petit frère*

*A mes frères Abdelhalim, Abderrezak, Aziz, Abdelhamid, mon petit Abdelmadjid et ma sœur Chahrazed.*

*A mon cher cousin Alilou et à toute ma famille*

*A tous mes amis.*

*A mon cher binôme, pote et frère Billel.*

*Je dédie ce modeste travail.*

*Abdelghani*

*En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés afin d'élaborer ce mémoire.*

*Que je dédie à mes parents : ma mère et mon père, qui m'ont encouragé et soutenu énormément, je le dédie également à mes deux sœurs Meriem et Zineb et mon frère Abdellah ainsi que mon neveu Louai*

*A mes amis les deux Billel et surtout Fatima.*

*A mon cher binôme qui a partagé avec moi le meilleur et le pire pour mener à terme ce modeste travail*

*Billel*

# ملخص

تشكل شبكات الاستشعارات اللاسلكية Wireless Sensors Network والتي يشار لها اختصاراً WSN، ثورة علمية في مجال الاتصالات اللاسلكية والنظم المدمجة، ذلك أنها فتحت المجال أمام ابتكار جيل جديد من التطبيقات في مجالات متنوعة مثل البيئة ورصد الأحوال الجوية، والأمن مثل اكتشاف المتطفلين وعمليات اقتحام المناطق المحظورة. ومع ذلك فإن أجهزة الاستشعار جد محدودة سواء من حيث نقل البيانات او من حيث الطاقة المخزنة و التي تشكل عائق كبير في فعالية الشبكات و لذلك تم ابتكار عدة طرق و حلول للتحسين من مستوي شبكات الاستشعارات اللاسلكية , ومن بين هذه الطرق الفعالة, التوجيه الهرمي للبيانات (Clustering) و الذي يعد من أفضل الطرق للوقوف في وجه هذه العوائق.

عملنا هو جزء من تطوير نظام فعال لنقل البيانات، مما يسمح بتمديد عمر الشبكات واسعة النطاق. لهذا، قمنا بدراسة بروتوكول تجميع وتوجيه يدعا TPSO-CR بعد التحليل، وجدنا أن هذا البروتوكول لديه بعض العيوب. ولذلك، اقترحنا نسخة محسنة تدعا E-TPSO-CR بمقارنة نتائج المحاكاة تظهر لنا انا النسخة المحسنة اكثر كفاءة في مجال استخدام الطاقة وطول عمر الشبكة.

الكلمات المفتاحية: شبكة الإستشعارات اللاسلكية, التوجيه الهرمي للبيانات, قابلية التوسع, تكوين مجموعات

كفاءة الطاقة, حياة الشبكة.

# Résumé

Les réseaux de capteurs sans fil (RCSFs) représentent une technologie émergente qui vise à offrir des solutions intéressantes pour la surveillance à distance des environnements complexes et distribués. Cependant, la limitation de ressources des nœuds-capteurs constitue une contrainte importante, principalement, en matière d'autonomie d'énergie. Le routage hiérarchique des données, basé sur le clustering, dans les RCSFs se révèle comme une bonne réponse à toutes ces exigences.

Notre travail s'inscrit dans le cadre de développement d'un protocole de transmission de données efficace, qui permet la prolongation de la durée de vie dans les réseaux à grande échelle. Pour cela, nous avons étudié un protocole de clustering et du routage qui s'appelle TPSO-CR « Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network ». Après l'analyse faite, nous avons constaté que ce protocole présente quelques inconvénients. Conséquemment, nous avons proposé une version améliorée nommée ETPSO-CR « Enhanced TPSO-CR ». Les résultats de comparaison par simulation ont montré que notre protocole (ETPSO-CR) surpasse le protocole TPSO-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, routage hiérarchique, clustering, passage à l'échelle, efficacité énergétique, durée de vie, NS3, MATLAB.

# *Abstract*

Wireless sensor networks (WSNs) are an emerging technology that aims to offer interesting solutions for remote monitoring of complex and distributed environments. However, the sensor node resource limitation is an important constraint, mainly for autonomy of energy. The hierarchical data routing, based on clustering, in WSNs is revealed as a good answer to all these requirements.

Our work is part of developing an effective data transmission protocol, which allows the extension of the lifetime in large scale networks. For this, we studied a clustering and routing protocol called TPSO-CR "Two-tier particle swarm optimization and clustering protocol for routing in wireless sensor network." After the analysis, we found that this protocol has some disadvantages. Therefore, we proposed an improved version called E-CR-TPSO "Enhanced TPSO-CR". Simulation results show that our protocol (ETPSO-CR) outperforms the protocol TPSO-CR in terms of energy efficiency and network longevity, and that for homogeneous and heterogeneous networks.

**Keywords:** wireless sensor networks, hierarchical routing, clustering, scalability, energy efficiency, lifetime, NS3, MATLAB.

# Table des matières

<i>Table de matières</i> .....	<i>VII</i>
<i>Liste des figures</i> .....	<i>X</i>
<i>Liste des tableaux</i> .....	<i>XI</i>
<i>Liste des abréviations</i> .....	<i>XII</i>
<i>Introduction générale</i> .....	<i>1</i>
<i>Partie I : Etat de l'art</i>	
Chapitre I : Généralités sur les réseaux de capteurs sans fil.....	4
1. Introduction .....	4
2. Les nœud-capteurs sans fil .....	4
2.1. Qu'est-ce qu'un nœud-capteur ?.....	4
2.2. Architecture et caractéristiques d'un nœud-capteur .....	5
2.2.1. Architecture d'un nœud-capteur .....	5
2.2.2. Caractéristiques d'un nœud-capteur .....	6
3. Architecture des réseaux de capteurs sans fil.....	8
3.1. Définition d'un RCSF .....	8
3.2. Caractéristiques des RCSFs .....	9
3.2.1. Caractéristiques Matérielles.....	9
3.2.2. Caractéristiques conceptuels .....	9
3.3. Pile protocolaire .....	11
3.4. Paradigme de communication .....	13
4. Domaines d'applications.....	13
5. Conclusion .....	16
Chapitre II : Clustering dans les réseaux de capteurs sans fil .....	18
1.Introduction.....	18
2.Concepts de base de clustering .....	18
2.1. Définition .....	18
2.2. Classification des algorithmes de clustering .....	19

2.3. Défis de clustering.....	20
2.4. Objectifs de clustering.....	21
2.5. Avantages de clustering.....	22
3. Routage basé sur le clustering .....	22
4. Protocoles de clustering dans les RCSFs .....	23
4.1. LEACH « Low Energy Adaptive Clustering Hierarchy ».....	23
4.2. LEACH-C (LEACH Centralisé) .....	26
4.3. LEACH-C Optimized.....	27
4.4. A Dynamic Round-Time Based Fixed LEACH .....	29
4.5. Two-tier particle swarm optimization protocol .....	33
5. Conclusion .....	34

**Partie II : Propositions et évaluations**

Chapitre III : Etude, amélioration et conception d'un protocole de clustering .....	36
1. Introduction.....	36
2. Présentation et développement du protocole <i>TPSO – CR</i> .....	36
2.1. Vue générale sur PSO.....	36
2.2. Hypothèses.....	38
2.3. Principe de fonctionnement de <i>TPSO – CR</i> .....	38
2.3.1. Clustering :.....	39
2.3.2. Routage.....	41
2.4. Algorithme du protocole <i>TPSO – CR</i> .....	43
3. Analyse du protocole <i>TPSO – CR</i> et discussion.....	45
4. Conception de la solution proposée .....	46
4.1. Elimination du coût de re-clustering .....	47
4.1.1. Modèle énergétique .....	47
4.1.2. Module d'estimation d'énergie .....	48
4.2. Adaptation de la durée des rounds aux énergies résiduelles .....	49
4.3. Distribution équitable des CHs et équilibrage de charge .....	50
4.4. Minimisation de la taille des routes générées .....	51
4.5. Organigramme de la solution proposée .....	51
5. Conclusion .....	52
Chapitre 04 : Implémentation, tests et comparaisons.....	55
1. Introduction.....	55

<b>2. Environnement expérimental .....</b>	<b>55</b>
<b>3. Implémentation .....</b>	<b>56</b>
<b>4. Tests .....</b>	<b>60</b>
<b>4.1. Configuration et scénario de tests .....</b>	<b>60</b>
<b>4.2. Résultats, comparaison et discussion .....</b>	<b>62</b>
<b>4.2.1. Réseau homogène .....</b>	<b>62</b>
<b>4.2.2. Réseau hétérogène .....</b>	<b>71</b>
<b>5. Conclusion .....</b>	<b>76</b>
<i>Conclusion générale et perspectives .....</i>	<i>77</i>
<i>Références bibliographiques .....</i>	<i>79</i>

# Liste des figures

<i>Figure 1.1 : Architecture physique d'un capteur.....</i>	<i>5</i>
<i>Figure 1.2 : Les différents modèles de capteurs.....</i>	<i>7</i>
<i>Figure 1.3 : Réseau de capteurs sans fil.....</i>	<i>8</i>
<i>Figure 1.4 : Pile protocolaire dans les RCSFs.....</i>	<i>11</i>
<i>Figure 2.1 : Structure d'un réseau clustérisé.....</i>	<i>19</i>
<i>Figure 3.1 : Organigramme de la solution proposée.....</i>	<i>52</i>
<i>Figure 4.1. Notre architecture d'implémentation.....</i>	<i>57</i>
<i>Figure 4.2. Résultat de clustering et du routage du protocole TPSO-CR pour un réseau de 100 nœuds. ....</i>	<i>63</i>
<i>Figure 4.3. Résultat de clustering et du routage du protocole ETPSO-CR pour un réseau de 100 nœuds. ....</i>	<i>63</i>
<i>Figure 4.4. L'énergie consommée dans un réseau de 100 nœuds. ....</i>	<i>65</i>
<i>Figure 4.5. L'énergie consommée dans un réseau de 200 nœuds. ....</i>	<i>66</i>
<i>Figure 4.6. L'énergie consommée dans un réseau de 300 nœuds. ....</i>	<i>66</i>
<i>Figure 4.7. Le nombre de nœuds vivants dans un réseau de 100 nœuds.....</i>	<i>67</i>
<i>Figure 4.8. Le nombre de nœuds vivants dans un réseau de 200 nœuds.....</i>	<i>68</i>
<i>Figure 4.9. Le nombre de nœuds vivants dans un réseau de 300 nœuds.....</i>	<i>68</i>
<i>Figure 4.10. La durée de vie dans un réseau de 100,200 et 300 nœuds.....</i>	<i>69</i>
<i>Figure 4.11. Le taux de paquets reçus au niveau de la BS pour les réseaux de 100,200 et 300 nœuds. ....</i>	<i>70</i>
<i>Figure 4.12. L'énergie consommée pour un réseau hétérogène de 100 nœuds. ....</i>	<i>71</i>
<i>Figure 4.13. L'énergie consommée pour un réseau hétérogène de 200 nœuds. ....</i>	<i>72</i>
<i>Figure 4.14. L'énergie consommée pour un réseau hétérogène de 300 nœuds. ....</i>	<i>72</i>
<i>Figure 4.15. Les nœuds vivants dans un réseau hétérogène de 100 nœuds. ....</i>	<i>73</i>
<i>Figure 4.16. Les nœuds vivants dans un réseau hétérogène de 200 nœuds. ....</i>	<i>74</i>
<i>Figure 4.17. Les nœuds vivants dans un réseau hétérogène de 300 nœuds. ....</i>	<i>74</i>
<i>Figure 4.18. La durée de vie dans un réseau hétérogène de 100, 200 et 300 nœuds. ....</i>	<i>75</i>
<i>Figure 4.19. Le taux de paquets reçus au niveau de la BS dans un réseau hétérogène de 100, 200 et 300 nœuds. ....</i>	<i>75</i>
<i>Figure 4.20. Le pourcentage d'amélioration entre le protocole TPSO-CR et ETPSO-CR dans les réseaux hétérogènes. ....</i>	<i>76</i>

# Liste des tableaux

<i>Table 4.1. Paramètres de simulation .....</i>	<i>60</i>
<i>Table 4.2. Paramètres des fonctions objectives.....</i>	<i>61</i>
<i>Table 4.3. Pourcentage d'amélioration entre le protocole TPSO-CR et ETPSO-CR par rapports au nombre de sauts.....</i>	<i>64</i>
<i>Table 4.4. Le pourcentage d'amélioration entre le protocole TPSO-CR et ETPSO-CR dans les réseaux homogènes.....</i>	<i>71</i>

# *Liste des acronymes*

**APTEEN:** Adaptive Threshold sensitive Energy Efficient sensor Network protocol.

**ASIC:** Application Specific Integrated Circuit.

**BS:** Base Station.

**CAN:** Convertisseur Analogique Numérique.

**CH:** Cluster Head.

**CQ:** Cluster Quality.

**DSP:** Digital Signal Processors.

**EE:** Energy Efficiency.

**EEHC:** Energy Efficient Heterogeneous Clustered.

**ETPSO-CR:** Enhanced Two-tier Particle Swarm Optimization for Clustering and Routing in wireless sensor network.

**FitnessC:** Fitness Clustering.

**FitnessR:** Fitness Routing.

**FND:** First Node Dies.

**FNP:** First Network Partitioning.

**FPGA:** Field Programmable Gate Array.

**FS:** Free Space.

**GPS:** Global Positioning System.

**HND:** Half Node Dies.

**LEACH:** Low Energy Adaptive Clustering Hierarchy.

**LEACH-C:** Centralized Low Energy Adaptive Clustering Hierarchy.

**LEACH-F:** Fixed Low Energy Adaptive Clustering Hierarchy.

**LNP:** Last Network Partitioning.

**LQ:** Link Quality.

**LRWPAN:** Low-Rate Wireless Personnel Area Network.

**MATLAB:** Matrix Laboratory.

**MP:** Multipath Fading.

**MTE:** Minimum Transmission Energy.

**NQ:** Network Coverage.

**NS3:** Network Simulator 3.

**OEP:** Optimisation par Essaim Particulaire.

**OSI:** Open Systems Interconnection.

**PRR:** Packet Received Rate.

**PSO:** Particle Swarm Optimization.

**PSO-C:** Particle Swarm Optimization Clustering.

**PSO-R:** Particle Swarm Optimisation Routing.

**RCSFs:** Réseaux de Capteurs Sans Fil.

**RSSI:** Received Signal Strength Indicator.

**TDMA:** Time division multiple access.

**TPSO-CR:** Two-tier Particle Swarm Optimization for Clustering and Routing in wireless sensor network.

## *Introduction générale*

De nos jours, la technologie est devenue indispensable dans la vie humaine, particulièrement les réseaux sans fil qui permettent une communication continue entre les gens, en éliminant les distances qui les séparent. Au cours de ces dernières années, la technologie des réseaux sans fil n'a cessé de croître grâce aux développements technologiques dans divers domaines liés à la micro-électronique. Ces développements ont permis la construction à faible coût de minuscules nœud-capteurs consommant peu d'énergie, et qui l'aptitude de mesurer ou détecter un évènement réel tel que : l'humidité, la luminosité, la température, la chaleur, etc. Ces nœud-capteurs sont tous réunis pour construire un Réseau de Capteurs Sans Fils (RCSF ou WSN : Wireless Sensor Network en anglais), ayant pour rôles de collecter les données d'un environnement et les diffuser au sein d'un réseau. Dans la conception d'un RCSF, le facteur primordial à prendre en considération est l'énergie, car la source principale d'énergie d'un nœud est une batterie à durée de vie limitée et souvent irremplaçable. En plus de ça, ces réseaux imposent beaucoup de contraintes dues à la limitation matérielle des nœud-capteurs en termes de stockage et de puissance de calcul.

A côté de la contrainte énergétique des RCSFs, la durée de vie du réseau est souvent considérée comme le principal critère de performance. C'est pour cela qu'un usage prudent et efficace de la batterie est primordial afin de prolonger la durée de vie de réseau. En outre, une large gamme d'application exige un déploiement à très grande échelle de RCSFs pour achever leurs objectifs, ce qui introduit la problématique de scalabilité. Cette dernière peut être définie comme l'aptitude de faire fonctionner un grand nombre de nœuds tout en maintenant une bonne performance.

Le clustering s'avère comme une bonne réponse à toutes ces contraintes. Son principe consiste à diviser le réseau en un certain nombre de groupes appelés clusters. Chaque cluster possède un leader qui est un nœud particulier désigné comme chef du groupe. Le clustering permet de localiser la recherche et la maintenance des routes au sein du cluster donc de réduire la taille des tables du routage. De plus, les protocoles hiérarchiques (protocoles basés sur le clustering) permettent d'optimiser la consommation énergétique menant à une meilleure durée de vie du réseau.

Notre sujet de fin d'étude s'inscrit dans le cadre d'un projet au CDTA qui consiste au développement des nœud-capteurs à très faible consommation énergétique. Pour assurer la

communication entre ces nœuds et la station de base, les membres de projet ont proposé la catégorie du routage hiérarchique. L'objectif principal de notre travail est le développement d'un protocole de transmission de données efficace qui possède les caractéristiques suivantes :

1. Le protocole doit prolonger la durée de vie du réseau au maximum possible.
2. Le protocole 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 : homogènes et hétérogènes.

Ce mémoire s'articule autour de quatre chapitres. Le premier chapitre s'inscrit dans le contexte d'une étude globale sur les réseaux de capteurs sans fil. Pour cela, nous présenterons leurs éléments de base, les contraintes et les caractéristiques qui influencent sur leur conception, leurs domaines d'applications, leurs catégories de communication ainsi que leurs défis et objectifs.

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 présenter quelques approches de clustering proposées dans la littérature afin d'appréhender leur principe de fonctionnement. Le troisième chapitre est consacré à l'étude détaillée d'un protocole de clustering et du routage nommé *TPSO – CR*, en mettant en relief ses avantages et ses inconvénients. Nous allons par la suite proposer et concevoir une version améliorée de ce protocole pour pallier aux inconvénients recensés.

Le quatrième chapitre est dédié à l'implémentation des deux protocoles *TPSO – CR* et sa version améliorée que nous avons proposée. Par la suite, nous allons effectuer une série d'expérience et de tests afin de comparer les performances des deux protocoles implémentés, et ce, par rapport à la consommation d'énergie, la durée de vie du réseau, le nombre de nœuds vivants et le taux de paquets reçu au niveau de la station de base.

Finalement, une conclusion clôture ce manuscrit et rappelle les principales contributions élaborées tout au long de ce travail. Elle présente également les perspectives et implications relatives aux résultats obtenus.

*Partie I :*

*Etat de l'art*

*Chapitre 01 :*

*Généralités sur les  
réseaux de capteurs sans  
fil*

# Chapitre 01 : Généralités sur les réseaux de capteurs sans fil.

## 1. Introduction

De nombreux systèmes nécessitent la surveillance et le suivi de l'environnement pour mesurer des phénomènes physiques afin de prendre les décisions nécessaires. Les progrès réalisés ces dernières années en micro-électronique et micromécanique ont permis de concevoir des capteurs de plus en plus petits, de plus en plus performants, autonomes et dont les capacités énergétiques ont évolué avec le temps. D'autre part, les techniques de réseaux mobiles permettent d'éliminer les fils et donc de déployer facilement, même dans les endroits hostiles, des réseaux de capteurs sans fil [1].

Dans ce premier chapitre, nous allons présenter les éléments constitutifs des RCSFs, leurs caractéristiques ainsi que leur architecture de communication. Nous discuterons également les principaux facteurs qui influencent la conception des RCSFs. A la fin nous présentons les différents domaines d'application où nous pourrions tirer profit de déploiement d'un RCSF.

## 2. Les nœud-capteurs sans fil

La miniaturisation du matériel et l'augmentation des capacités de calcul et de mémoire en informatique, ainsi que les progrès réalisés ces dernières décennies dans les domaines de la microélectronique et des technologies de communication sans fil, ont permis aux réseaux de capteurs de voir la lumière, et cela à une échelle très large au point d'accomplir les tâches les plus complexes pour l'humain.

### 2.1. Qu'est-ce qu'un nœud-capteur ?

Un nœud-capteur est un dispositif électronique doté de fonctionnalités de sensation avancées. Il a l'aptitude de mesurer ou détecter un événement réel tel que : l'humidité, la luminosité, la température, la chaleur etc. Les nœud-capteurs sont aussi capables de traiter les

informations collectées, les analyser puis en fin les transmettre, via un lien de communication sans fil, à un nœud distant appelé la station de base.

## 2.2. Architecture et caractéristiques d'un nœud-capteur

### 2.2.1. Architecture d'un nœud-capteur

Selon le domaine d'application, il existe plusieurs modèles de capteurs sur le marché : les capteurs de vibration, de pression, d'humidité, etc. Cependant, malgré cette disparité apparente, les nœud-capteurs sont munis d'une architecture matérielle similaire. Ils sont composés essentiellement d'une unité de : captage, traitement, stockage, communication, et énergie. Des composants optionnels peuvent être ajoutés par rapport au domaine d'application [2], comme un système de localisation GPS (Global Positioning System) ou un générateur d'énergie. Ces éléments sont présentés sur la figure suivante (les traits discontinus représentent les éléments optionnels).

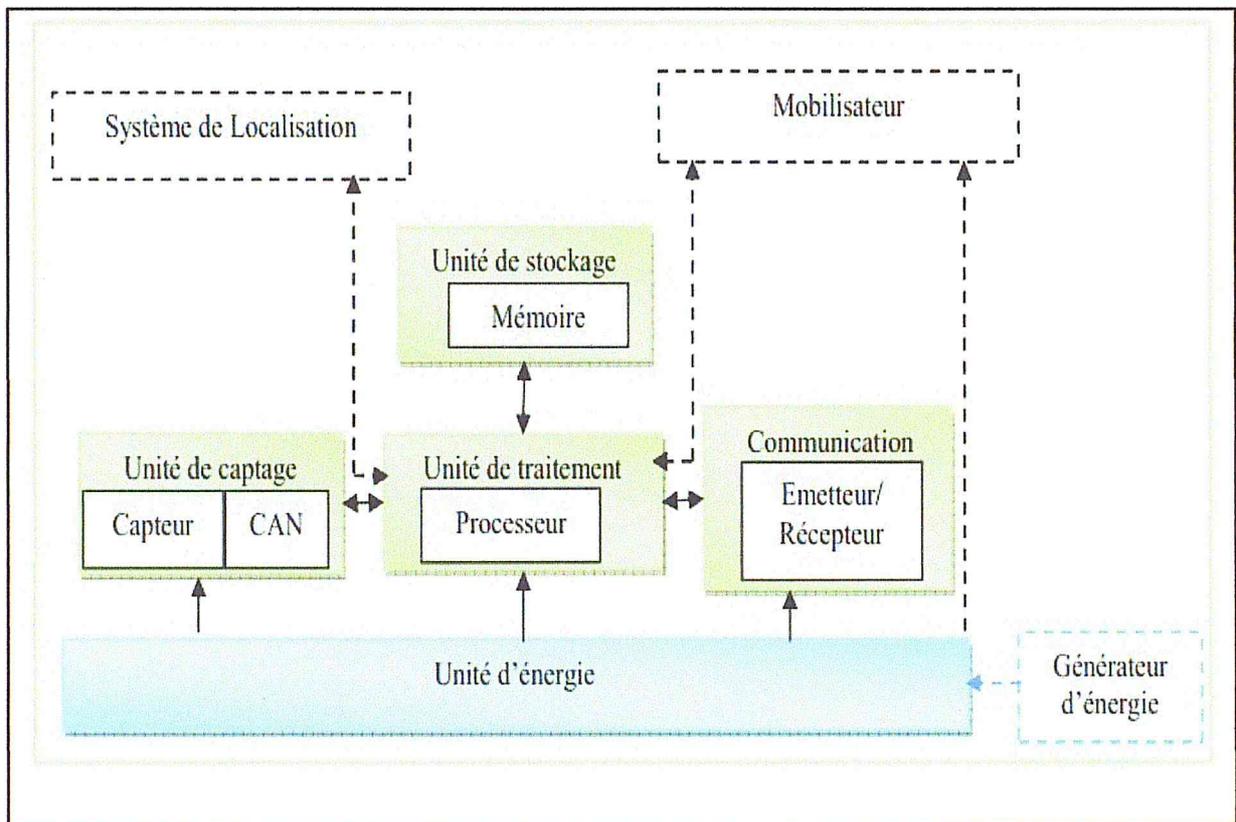


Figure 1.1. Architecture physique d'un capteur.

Nous présentons ci-dessous chacune des unités d'un nœud-capteur :

**1- Unité d'énergie :** Un nœud-capteur est équipé d'une source d'alimentation, généralement une batterie. Les batteries sont souvent non rechargeables, dans les environnements sensibles, il est difficile voire impossible de changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse car elle influe directement sur la durée de vie d'un réseau de capteurs.

**2- Unité de captage :** Unité responsable de capter différents types d'informations, qui se produisent autour d'un nœud-capteur. Ces informations peuvent être sous forme de vibrations, bruits, etc. Le capteur transforme les données analogiques captées en données numériques à l'aide d'un convertisseur analogique numérique CAN [3].

**3- Unité de traitements :** Elle contient un processeur chargé de recevoir les données de l'unité de captage ou d'autres nœud-capteurs pour effectuer les traitements nécessaires. Le processeur doit exécuter des programmes et des protocoles de communication différents. Il existe plusieurs types de processeurs utilisés comme les microcontrôleurs, les DSP (Digital Signal Processors), les FPGA (Field Programmable Gate Array) et les ASIC (Application Specific Integrated Circuit).

**4- Unité de communication :** Tous les échanges inter-réseau se font grâce à cette unité, elle se charge de tout le trafic de données que ce soit des émissions ou des réceptions par le biais d'un support de communication sans fil.

**5- Unité de stockage :** Elle inclut la mémoire de programme dont les instructions sont exécutées par le processeur et la mémoire de données (pour conserver les données fournies par l'unité de captage et d'autres données locales). La taille de cette mémoire est généralement limitée [4].

### 2.2.2. Caractéristiques d'un nœud-capteur

Un nœud-capteur possède les caractéristiques suivantes :

- **Petite taille :** La taille d'un nœud-capteur varie selon la composition entre  $1\text{cm}^3$  dans le cas d'un nœud-capteur intégré, et  $4\text{cm} \times 4\text{cm}$  dans le cas d'un nœud-capteur à carte avec des extensions possibles. Sa taille relativement petite lui permet de se loger dans beaucoup d'endroits avec discrétion (*Figure 1.2*).

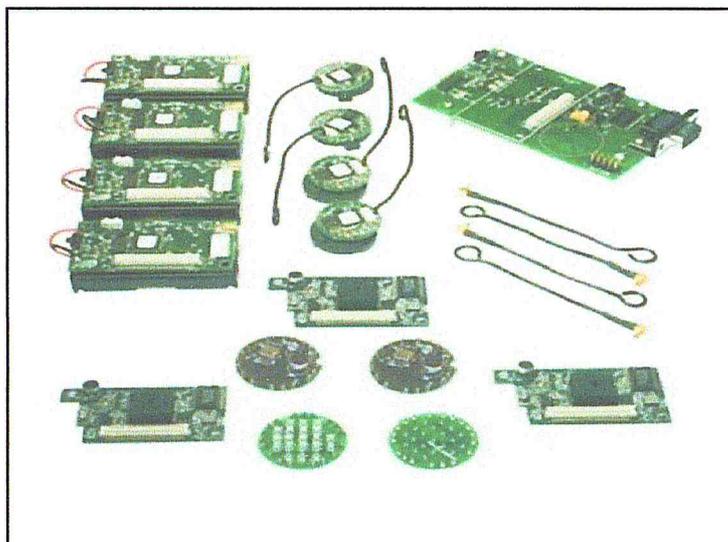


Figure 1.2. Les différents modèles de capteurs [5].

- **Source d'énergie très limitées** : L'énergie constitue le principal handicap des nœud-capteurs. La taille de la batterie détermine la taille du nœud-capteur. Même avec une batterie relativement grande (2pile AAA) l'optimisation de l'énergie surtout en communication est primordiale.
- **Taux de transfert limité** : Un faible débit de données n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes. Le taux de transfert dans les nœud-capteurs est limité à 250Kbps.
- **Faible coût de fabrication** : Le faible coût de production constitue un avantage considérable pour les nœud-capteurs, ce qui permettra de généraliser et diversifier leur utilisation, et pourquoi pas, remédier au défaut d'énergie par le faible coût de production.
- **Autonome et adaptative** : Un nœud-capteur intégré est complètement autonome, il peut effectuer toutes les tâches qui lui sont affectées (capter, communiquer, etc.) sans avoir besoin d'un composant étranger. En revanche, il est moins adaptatif qu'un nœud-capteur à carte qui est très adaptatif et peut recevoir plusieurs capteurs pour différentes grandeurs physiques.

### 3. Architecture des réseaux de capteurs sans fil

#### 3.1. Définition d'un RCSF

Un réseau de capteurs sans fils (RCSF) est un ensemble de nœud-capteurs communiquant entre eux dans le but de surveiller un phénomène quelconque et collecter des informations d'une manière autonome. Ces nœud-capteurs peuvent être aléatoirement éparpillés dans une zone géographique appelée «champ de captage » qui correspond au stade d'intérêt pour le phénomène mesuré. Les nœud-capteurs utilisent une communication sans fil pour acheminer les données mesurés par le biais d'un routage multi-sauts (une information doit passer par plusieurs nœuds pour pouvoir atteindre la station de base) vers un nœud qui est responsable de la collecte d'informations appelé station de base ou puits. La station de base fonctionne comme un point d'accès entre les nœuds et l'utilisateur. Ce dernier peut lui aussi adresser des requêtes aux nœud-capteurs en précisant l'information d'intérêt.

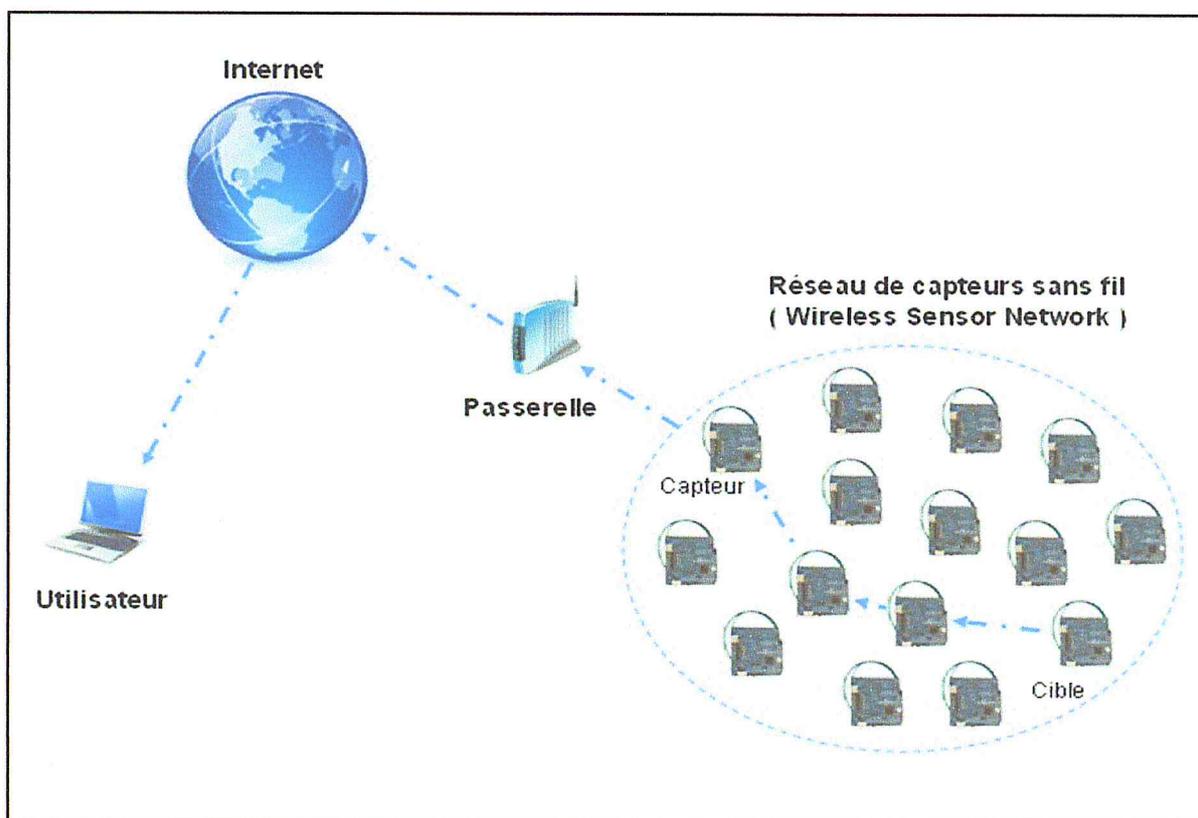


Figure 1.3. Réseau de capteurs sans fil [6].

## 3.2. Caractéristiques des RCSFs

Notons que les caractéristiques des RCSFs se dérivent des limitations qu'ils connaissent. Par ailleurs, ces caractéristiques s'avèrent comme des facteurs qui influencent directement l'architecture, en étant dédiés particulièrement à ce type de réseau.

### 3.2.1. Caractéristiques Matérielles

Les RCSFs possèdent les caractéristiques matérielles suivantes :

- **Dimension** : la taille réduite des nœuds capteurs est parmi les majeurs avantages de ces derniers. Elle permet un déploiement simple et flexible du réseau, néanmoins, la puissance des batteries utilisées est limitée par la petite taille de ces nœud-capteurs.
- **Puissance de calcul** : parmi les caractéristiques les plus critiqués dans un RCSF est la modestie de ses capacités de stockages et les puissances de calcul.

### 3.2.2. Caractéristiques conceptuels

En plus de caractéristiques matérielles décrites ci-dessus, les RCSFs possèdent les caractéristiques conceptuelles suivantes :

- **Tolérance aux pannes** : Dans un RCSF, les nœuds capteurs doivent continuer à maintenir le bon fonctionnement du réseau sans avoir la moindre interruption bien qu'ils soient sujets à des pannes dues à plusieurs causes, notamment le manque d'énergie, les interférences avec l'environnement d'observation, ou bien à cause de leur fabrication (il peut y avoir des capteurs défectueux) . Afin que les pannes n'affectent pas la tâche principale du réseau il faut minimiser l'influence de ces pannes et savoir les bien gérer.
- **Extensibilité (passage à l'échelle)** : Le nombre de nœuds déployés dans une zone de captage varie de quelques centaines à plusieurs milliers de capteurs, et dans certaines applications des millions. La densité de nœuds dépend de l'application pour laquelle le réseau a été déployé. Par exemple, dans une zone avec un diamètre moins de 10m la densité peut s'étendre aux centaines de nœuds.

- **Coût de production** : Les RCSFs se composent d'un très grand nombre de nœuds de capteurs. Le coût d'un nœud est très décisif pour pouvoir justifier le coût global d'un réseau donc il faut que le coût de fabrication de ces nœuds ou bien le coût global du réseau ne dépasse pas celui d'un réseau traditionnel sinon le réseau ne soit plus profitable.
- **Environnement** : Tout au long de la durée de vie du RCSF, les nœuds peuvent être sujets à des changements environnementaux, notamment le déplacement des nœuds à cause d'une forte vitesse de vent, la forte chaleur, l'humidité, la pluie etc. tous ces derniers peuvent perturber le fonctionnement du système. Pour cela, les nœuds capteurs doivent résister à tous ces différentes conditions.
- **Consommation énergétique** : La source principale d'énergie d'un nœud est, dans la plupart des cas, une batterie à durée de vie limitée et souvent irremplaçable. Dans la conception d'un RCSF, le facteur le plus important à prendre en considération c'est l'énergie consommée par le nœud capteur lors de la détection et de la transmission de données. Etant donné que la phase de transmission est la fonction qui consomme le plus d'énergie, qui est proportionnelle au carrée de la distance de transmission [7], et à la taille du paquet à envoyer. Par conséquent, il faut mettre en évidence que les protocoles et les algorithmes conçus doivent prendre en considération ce facteur qui est souvent négligé dans les protocoles développés dans les réseaux classiques afin de prolonger et augmenter la durée de vie du réseau.
- **Agrégation des données** : Parmi les techniques qui sont utilisées pour économiser de l'énergie ainsi que pour éviter les données redondantes, il y a eu l'agrégation ou bien la fusion de données. Elle consiste à combiner des données qui proviennent de différentes sources en leur appliquant des fonctions. Les fonctions appliquées peuvent être des fonctions de clauses comme le minimum, le maximum, etc. Par contre, dans certains réseaux, la génération des données redondantes constitue une nécessité pour bien préciser le phénomène capté.

### 3.3. Pile protocolaire

La pile de communication des RCSFs suit le modèle OSI. Indépendamment des différentes fonctions que chaque nœud du réseau doit remplir, le procédé de communication est divisé en cinq couches standardisées. Ces dernières doivent avoir la possibilité de se communiquer de telle sorte que chaque couche utilise les services de la couche précédente pour pouvoir fournir des services pour la couche suivante. La pile doit assurer en plus la coordination entre les différents nœuds même s'ils sont très nombreux.

En outre, les plans de gestion d'énergie, de mobilité et des tâches surveillent et gèrent la consommation d'énergie, les mouvements, et la répartition des tâches entre les nœuds-capteurs. Ces plans aident les nœuds-capteurs à coordonner les tâches de détection et à réduire l'ensemble de la consommation d'énergie.

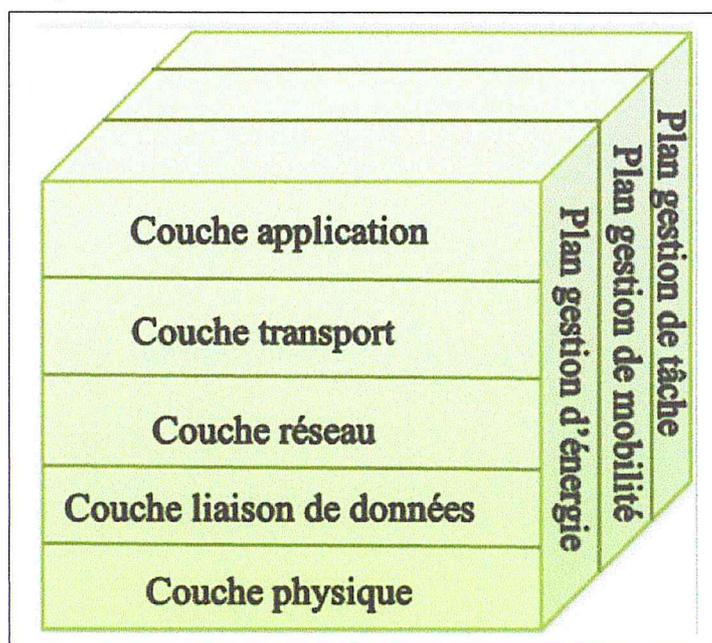


Figure 1.4. Pile protocolaire dans les RCSFs [7].

Description des couches :

**1-Couche physique** : Le niveau physique est responsable des échanges de flux de bits sur le support physique en effectuant la réception, l'émission, la modulation et la détection du signal.

**2-Couche Liaison de données :** Elle assure la communication saut-par-saut entre les nœuds, en plus elle est la responsable de multiplexage du flux de données, du contrôle des erreurs et du contrôle d'accès au média, pour gérer les collisions qui peuvent se produire à cause de la nature de medium sans fil. Tout cela doit se faire en minimisant l'énergie consommée et en répondant aux exigences imposées par les RCSFs.

**3- Couche réseau :** Cette couche est responsable d'assurer les fonctionnalités de routage et la configuration du réseau. Dans la plupart des applications des RCSFs, un nœud ne possède pas une adresse IP donc l'acheminement se déroule à base d'attributs (ID). L'envoi de données dans les RCSFs est basé sur les événements que ce soit un envoi d'informations captés vers un collecteur ou bien une diffusion de ce dernier.

**4-Couche transport :** Pour le cas des RCSFs la couche transport peut être nécessaire pour garantir une transmission totalement crédible à partir de la source vers la destination et permettre la reconfiguration des nœud-capteurs [8].

**5-Couche application :** A travers des collaborations avec les autres couches de la pile protocolaire, la couche application apporte sa contribution dans les réseaux de capteurs sans fil en satisfaisant les différentes exigences du client dans le but d'interagir avec le logiciel et rendre les couches inférieures invisible aux applications de gestion de réseaux de capteurs [9].

À l'égard des contraintes imposées par la nature des RCSF, trois niveaux de gestion sont jugés nécessaires au niveau de chaque nœud-capteur :

**1-gestion d'énergie :** La durée de vie d'un RCSF se définit par la durée de vie de ses nœud-capteurs. Pour cela, la consommation d'énergie se présente comme un facteur décisionnaire. Par conséquent une gestion d'énergie est jugé nécessaire.

**2-Gestion des tâches :** Ce niveau assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau [10].

**3-Gestion de mobilité :** Ce niveau détecte et enregistre tous les mouvements des nœuds capteurs afin de leur permettre de garder une trace ou bien une route vers l'utilisateur final et maintenir une image récente sur les nœuds voisins, tout ça a pour but d'équilibrer l'exécution des tâches et la consommation d'énergie.

### 3.4. Paradigme de communication

Les paradigmes de communication les plus utilisés dans les RCSFs sont en nombre de quatre : périodique, évènementiel, requête/réponse, hybride.

**1-Périodique :** chaque nœud-capteur d'un RCSF envoie les paquets aux destinations de manière périodique. La plupart des applications génèrent du trafic périodique.

**2-Evènementiel :** Les capteurs envoient des paquets aux destinations lorsqu'ils détectent un évènement. Par exemple, dans les applications de surveillance, un capteur n'envoie des paquets que s'il détecte une présence dans la zone qu'il surveille.

**3-Requete/Réponse :** Les capteurs envoient des paquets aux destinations lorsque le contrôleur du réseau envoie une requête pour demander une information particulière.

**4-Hybride :** c'est quand une application utilise un ou plusieurs paradigme de communication (périodique, évènementiel ou requête/réponse) à la fois.

## 4. Domaines d'applications

La miniaturisation, l'adaptabilité, le faible coût et la communication sans fil permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre le domaine des applications existantes. Parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions, on peut noter le domaine militaire, sanitaire, industriel, médical, climatique, environnemental etc. [11].

### 4.1. Applications militaires

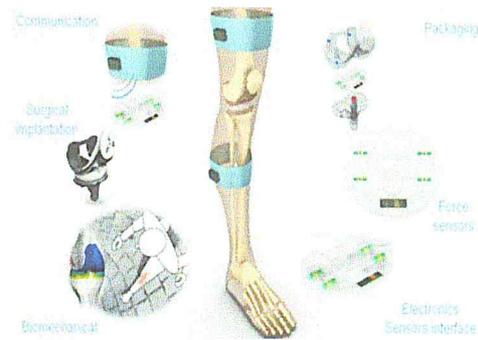
Comme dans le cas de la majorité des technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Ils peuvent être rapidement déployés et utilisés pour la surveillance des champs de bataille afin de fournir des renseignements concernant l'emplacement, le nombre, le mouvement, et l'identité des soldats et des véhicules, ou bien encore pour la détection des



agents chimiques, biologiques et nucléaires. Des tests concluants ont déjà été réalisés dans ce domaine par l'armée américaine dans le désert de Californie.

### 4.2. Applications médicales

La médecine et le système de santé peuvent aussi profiter de l'application des RSCFs, les capteurs peuvent être implantés dans le corps humain pour contrôler les problèmes médicaux comme le cancer et pour aider les patients à maintenir leur santé. Ils peuvent être utilisés aussi pour surveiller les patients et l'avancement de leurs états dans un hôpital. On



pourrait imaginer que dans le futur, la surveillance des fonctions vitales de l'être humain serait possible grâce à des nœud-capteurs qui pourront être avalés ou implantés sous la peau. Actuellement, des micro-caméras qui peuvent être avalées existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures. Les auteurs d'une récente étude, présentent des capteurs qui fonctionnent à l'intérieur du corps humain pour traiter certains types de maladies.

### 4.3. Applications environnementales

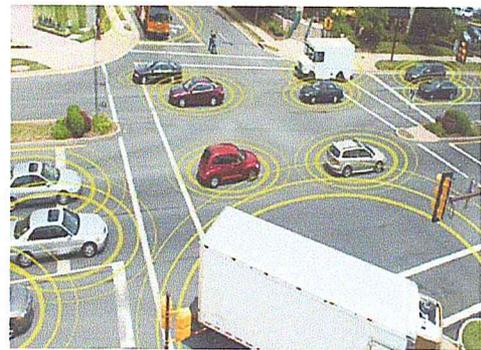
Les RSCFs peuvent être utilisés pour surveiller les changements environnementaux. [12]. Ils servent à déterminer les valeurs de certains paramètres à un endroit donné, comme par exemple, la température et la pression atmosphérique. Comme applications environnementales des RSCFs, nous pouvons penser à des thermo-capteurs dispersés à partir d'un avion sur une forêt qui peuvent signaler un éventuel début d'incendie dans le champ de capture. Nous pouvons aussi penser à des capteurs qui peuvent être distribués sur la zone à contrôler pour prédire les habitants des régions voisines d'un déclenchement d'un volcan ou d'une éruption.

#### 4.4. Applications industrielles

Les applications de l'industrie en termes de production s'appuient particulièrement sur l'utilisation des capteurs pour vérifier les contraintes de qualité des produits comme le poids, l'isolation galvanique, etc. Ils sont aussi applicables à la surveillance des conditions du travail des ouvriers, donc ils seront utilisés par exemple pour éviter les risques de la radioactivité. Par ailleurs, les capteurs disséminés dans l'environnement industriel peuvent relever le niveau de pollution et le relayer vers un terminal spécifié. Encore, les RCSFs permettent de suivre dans les entreprises manufacturières le procédé de production à partir des matières premières jusqu'au produit final.

#### 4.5. Application de transports

Récemment, les RCSFs sont utilisés pour valider un prototype d'un avion ou d'un satellite dans le domaine de l'aérospatiale, en effectuant les mesures qui doivent se faire en temps réel et sans perte de données. Ces mesures portent sur la température et la pression des ailes d'avions durant les vols d'essai par exemple. La taille des capteurs est un élément crucial



dans tel domaine pour éviter les altérations aux réponses mécaniques de ces structures. Autres applications de l'habitat telles que les applications du suivi et d'observation d'habitat peuvent être également réalisées.

#### 4.6. Applications dans le domaine sportif

L'évolution des réseaux de capteurs est utilisée de plus en plus dans le domaine sportif, à savoir les systèmes de surveillance, les systèmes de calcul de trajectoires (comme dans le tennis), systèmes de détection d'erreurs d'arbitrage (comme dans le football indiquent si le ballon a franchi la ligne de but) et on a pleines d'autres applications des réseaux de capteurs dans ce domaine.



## 5. Conclusion

Dans ce chapitre nous avons procédé à l'étude des RCSFs. Nous avons posé les briques de base et assemblé quelques concepts nécessaires à la compréhension de nos problématiques. Nous avons aussi représenté le nœud-capteur, ses fonctionnalités et son architecture. Ensuite nous avons opéré les caractéristiques d'un RCSF et présenté quelques applications. A travers nos lectures, nous avons remarqué que la minimisation de la consommation d'énergie d'un nœud-capteur est le cheval de bataille de toutes les solutions et protocoles proposés. En effet, lorsque ce n'est pas l'objectif principal, alors c'est sûrement un critère de performance capital. Les RCSFs possèdent des caractéristiques particulières qui les différencient des autres types de réseaux sans fil. Ces spécificités telles que la contrainte d'énergie réduite et la scalabilité sollicitent le besoin de concevoir des nouveaux protocoles qui s'adapteront aux caractéristiques des RCSFs comme par exemple ceux qui sont basés sur le clustering. Dans le chapitre suivant nous allons présenter les concepts de base de clustering en passant par ses objectifs, ses défis et ses avantages aussi. Dans ce qui suit nous allons exposer quelques protocoles afin d'éclaircir leurs impacts sur les RCSFs.

*Chapitre 02 :*

*Clustering dans les  
Réseaux de capteurs  
sans fil*

# Chapitre 02 : Clustering dans les réseaux de capteurs sans fil.

## 1. Introduction

Comme nous l'avons mentionné dans le chapitre précédent, la contrainte principale des RCSFs est leurs budgets énergétiques modérés et limités. Ceci car le remplacement des batteries des nœuds est une tâche difficile ou même irréalisable dans certains cas. Puisque la durée de vie de réseau est souvent considérée comme un critère de performance capital, un usage prudent et efficace de la batterie est primordial afin de prolonger la durée de vie de réseau. En outre, une large gamme d'application exige un déploiement à très grande échelle de RCSFs pour achever leurs objectifs, ce qui introduit la problématique de scalabilité. Cette dernière peut être définie comme l'aptitude de faire fonctionner un grand nombre de nœuds tout en maintenant une bonne performance.

Le clustering s'avère comme une bonne réponse à toutes ces contraintes [13]. Dans cette optique, ce chapitre est consacré à la présentation des concepts de base de clustering et ses avantages surtout en ce qui concerne le routage hiérarchique. Nous allons par la suite résumer le principe de fonctionnement de quelques protocoles de clustering.

## 2. Concepts de base de clustering

### 2.1. Définition

Le clustering consiste à partitionner le réseau en clusters où dans chacun d'entre eux, un seul nœud-capteur est sélectionné comme Cluster Head (CH) pour jouer le rôle spécial de point de transfert (voir figure 2.1). En effet, chaque CH crée un plan de transmission pour les nœud-capteurs dans le cluster, ce qui permet aux antennes radio de chaque nœud non-CH d'être éteintes toutes les fois, excepté pendant son temps de transmission. L'agrégation des nœud-capteurs en clusters permet de réduire la complexité des algorithmes de routage (l'acheminement de données vers la station de base (BS)), d'optimiser les données collectées en les faisant gérer localement par un CH, de faciliter l'agrégation des données, de simplifier la gestion du réseau, d'optimiser les dépenses d'énergie, et enfin de rendre le réseau plus

évolutif (scalable). Le clustering permet aux nœuds d'effectuer des communications sur de petites distances avec leurs CHs.

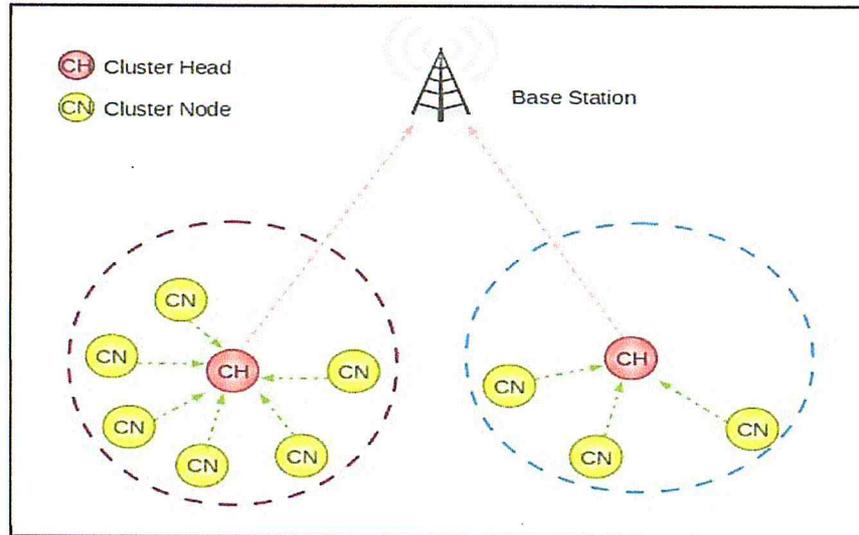


Figure 2.1. Structure d'un réseau clustérisé.

## 2.2. Classification des Algorithmes de Clustering

Les méthodes de clustering dans les RCSFs ont plusieurs classifications basées sur différents paramètres. Sur la base des paramètres utilisés pour l'élection des CHs, les algorithmes de clustering peuvent être classés comme déterministes et adaptatifs. Dans les systèmes déterministes, les attributs spéciaux des nœuds-capteurs sont considérés, comme leurs identifiants (Node\_ID), le nombre de voisins qu'ils ont au sein d'une transmission prédéfini (Node\_degré), etc. Par exemple, l'algorithme présenté en [14] favorise que les nœuds avec les plus petits IDs vont devenir CHs. Cette approche peut ne pas être appropriée pour les réseaux de capteurs qui sont limités en énergie, car elle pénalise des nœuds spécifiques dans le réseau indépendamment de leur durée de vie de la batterie. Kuhn et al. [15] et Amis et al. [16] sélectionnent les nœuds avec les plus grands degrés dans l'ordre pour créer des clusters denses et élire l'ensemble dominant minimal de CHs. Toutefois, cela pourrait entraîner un drainage rapide de la batterie des nœuds de grands degrés. Par contre, dans les schémas adaptatifs, les nœuds avec poids supérieur peuvent devenir des CHs. Le poids du nœud est utilisé pour définir sa chance pour être CH. Ce poids peut être l'énergie

résiduelle, la distance par rapport à la station de base ou une combinaison de plusieurs paramètres (par exemple, l'énergie restante et le degré).

En outre, les algorithmes de clustering peuvent être classés en centralisés et distribués. Les approches centralisées (comme LEACH-C [17], APTEEN [18]) nécessitent que des informations globales sur la topologie du réseau soient envoyées à la BS, ce qui limite leur efficacité dans les réseaux à très grande échelle. Néanmoins, les algorithmes de clustering 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. Dans de telles approches, un nœud décide de rejoindre un cluster ou devenir CH sur la base d'informations obtenues qu'à partir de ses voisins à un saut.

Selon le nombre de sauts entre les nœuds-capteurs et les CHs, en communication intra-cluster, les algorithmes de clustering peuvent être simplement classés en un saut ou multi-sauts. La même classification peut être appliquée dans la communication intra et inter-clusters (entre les CHs et la BS).

### 2.3. Défis de Clustering

Le clustering dans les RCSFs est confronté à plusieurs défis de déploiement, notamment en ce qui concerne la connectivité, la sélection de la fréquence optimale de rotation des CHs, le calcul de la taille optimale des clusters, etc. Dans ce qui suit, nous résumons quelques défis des protocoles de clustering dans les RCSFs.

- **Rotation du rôle de CH :** les nœuds-capteurs simple (non-CHs) dans un cluster ne transmettent leur données qu'à leurs CHs respectifs, tandis que le CH reçoit les données de tous les membres de son cluster, exerce des fonctions d'agrégation et de traitement du signal, et transmet les données à la BS distante. Par conséquent, l'énergie au niveau des CHs pourrait être épuisée rapidement que les autres nœuds non-CHs. Une fois que le CH draine son énergie, il ne fonctionne plus, et tous les nœuds qui appartiennent à son cluster perdent la capacité de communication. Donc il est essentiel de faire tourner le rôle de CHs parmi les nœuds. Cependant, la rotation du rôle de CH entraîne un gaspillage important de l'énergie. C'est pour cela qu'il faut bien paramétrer la fréquence de rotation du rôle de CHs.

- **La taille optimale des clusters :** La taille de cluster est un paramètre important. Si la taille de cluster diminue, la consommation d'énergie au sein de chaque cluster est plus petite. Cependant, le nombre de CHs sera alors augmenté, de sorte que le réseau de base résultant formé par ces CHs va devenir plus compliqué. Un plus petit nombre de CHs forme un réseau de base plus simple. Pourtant, ce serait exiger une plus grande taille de cluster. Il est alors important de trouver le bon compromis entre la taille de cluster et le nombre de CH.

### 2.4. Objectifs de clustering

Les algorithmes de clustering dans la littérature varient dans leurs objectifs. Souvent, l'objectif de clustering est réglé de manière à répondre aux exigences des applications. La discussion suivante surligne les objectifs les plus populaires de clustering :

- **Équilibrage de charge:** L'équilibrage de charge est une question très urgente dans les RCSFs où les CHs sont cueillies à partir des nœud-capteurs disponibles [19]. Dans ce cas, la construction des clusters équilibrés devient cruciale pour étendre la durée de vie du réseau, car ça empêche l'épuisement de l'énergie d'un sous-ensemble de CHs.
- **Tolérance aux fautes:** Tolérer l'échec des CHs est généralement nécessaire dans plusieurs applications, afin d'éviter la perte des données importantes qui proviennent des nœuds-capteurs. La façon la plus intuitive pour récupérer à partir de l'échec d'un CH est le re-clustering du réseau. La rotation du rôle des CHs par les nœuds du cluster peut également être un moyen pour la tolérance aux pannes [20].
- **Augmentation de la 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 BS [21].
- **Nombre minimal de clusters:** Cet objectif est particulièrement commun quand les CHs sont des nœuds riches en ressources spécialisées [22]. Le concepteur du réseau doit essayer d'employer le moins de ces nœuds étant donné qu'ils ont tendance à être plus chers et plus vulnérables que les autres nœuds-capteurs.
- **Longévité maximale de 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. 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 [23]. Si possible, les CHs doivent être placés à proximité de la plupart des nœuds-capteurs de ses clusters [22]. Par contre, lorsque les CHs sont des nœuds-capteurs réguliers, leur durée de vie peut être étendue en limitant leurs charges.

### **2.5. Avantages de clustering**

Le clustering des nœuds-capteurs a été largement suivi par la communauté des chercheurs dans le but de soutenir l'évolutivité du réseau, en plus de ça, le clustering présente de nombreux avantages. Il peut économiser la bande passante de communication, car il limite la portée des interactions inter-clusters au CHs (la communication entre les CHs et la BS), et ainsi il évite l'échange de messages redondants entre les nœuds-capteurs [24]. En outre, le clustering peut aussi stabiliser la topologie du réseau au niveau des nœuds-capteurs et réduire ainsi le temps de maintenance de topologie. Les nœuds-capteurs se soucieraient seulement de la connexion avec leurs CHs et ne serait pas affectée par les changements au niveau inter-clusters [25]. Le CH peut également mettre en œuvre les stratégies de gestion optimisées, agréger les données recueillies pour améliorer encore le fonctionnement du réseau, et aussi diminuer le nombre de paquets relayés et prolonger la durée de vie des batteries des nœuds-capteurs pour arriver finalement à la prolongation de la durée de vie du réseau [24].

## **3. Routage basé sur le clustering**

Le routage à base de clustering ou le routage hiérarchique est une approche promotrice dans les réseaux de capteurs sans fil. Ceci est dû aux avantages de clustering présentés précédemment (voir section 2.5).

Le clustering est la phase la plus importante dans un protocole de routage hiérarchique et a une grande influence sur le routage proprement dit. En effet, pour les algorithmes de clustering qui construisent des clusters uni-saut inter et intra-clusters, aucun algorithme de routage ne doit être mis en œuvre. Dans ce cas, chaque nœud doit enregistrer l'adresse de son CH. Ces derniers, quant à eux, enregistrent l'adresse de la BS. Nous parlons ici d'une simple transmission de données plutôt qu'un routage. Dans le cas de la construction des clusters multi-sauts, un algorithme de routage est exécuté localement dans chaque cluster pour construire les routes entre tous les membres et leurs CHs respectifs. En outre, un algorithme de routage peut être utilisé pour construire un deuxième niveau d'hierarchie qui relie les CHs et la BS.

## 4. Protocoles de Clustering dans les RCSFs

Il existe plusieurs protocoles de Clustering dans les RCSFs, nous allons exposer quelques-uns dans ce qui suit.

### 4.1. LEACH « Low Energy Adaptive Clustering Hierarchy »

#### 4.1.1. Description

Dans [26], les auteurs ont proposé un algorithme de clustering distribué appelé LEACH pour le routage hiérarchique dans les réseaux de capteurs homogènes (les nœuds-capteurs sont tous identiques physiquement). Le protocole LEACH est un protocole adaptatif et auto-organisé, qui utilise la randomisation pour répartir la charge de l'énergie de manière uniforme sur tous les nœuds-capteurs du réseau. L'opération de LEACH est divisée en rounds (cycles), où chaque round commence par une phase de mise en place *set-up phase*, dans laquelle les clusters sont organisés, suivie d'une phase *steady-state*, lorsque les transferts de données vers la BS surviennent. Ces deux phases sont résumées ci-après.

#### A) Phase d'initialisation (set-up phase)

Initialement, chaque nœud décide ou non de devenir un CH pour le round courant. Cette décision est basée sur le pourcentage suggéré des CHs pour le réseau (déterminé a priori) et le nombre de fois que le nœud était un CH dans les rounds précédents. Cette décision est prise par le nœud « n » qui choisit de manière aléatoire un nombre compris entre 0 et 1. Si le nombre est inférieur à un seuil  $T(n)$ , le nœud devient un CH pour le round courant. Le seuil est fixé comme suit:

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

Où

$P$  : le pourcentage souhaité des CHs (par exemple,  $P = 0,05$ ) ;

$r$  : le cycle actuel (round) ;

$G$  : l'ensemble des nœuds qui n'ont pas été des CHs durant les  $1/p$  rounds précédents.

Par l'utilisation de ce seuil, chaque nœud sera un CH dans les  $1/p$  rounds. Pendant le round 0 ( $r = 0$ ), chaque nœud a une probabilité  $P$  de devenir un CH. Les nœuds qui sont des CHs dans le round 0 ne peuvent pas être des CHs pour les prochains  $1/p$  rounds. Dans ce cas, nous supposons que tous les nœuds commencent avec la même quantité d'énergie et le fait d'être un CH enlève à peu près la même quantité d'énergie pour chaque nœud.

Chaque nœud qui a élu lui-même comme CH pour le round actuel diffuse un message d'annonce « CH-advertissement » au reste des nœuds. Les nœuds non-CHs doivent garder leurs récepteurs allumés au cours de cette phase pour entendre les annonces de tous les nœuds CHs. A la fin, chaque nœud non-CH décide le cluster auquel il appartiendra dans ce round et ce selon la force du signal reçu qui provient des nœuds CHs. Les nœuds doivent ensuite informer le CH choisi par l'envoi d'un message *Join request*. Au cours de cette phase, tous les nœuds du CHs doivent garder leurs récepteurs allumés.

Le nœud CH reçoit tous les messages des nœuds qui souhaitent être inclus dans le cluster. Basé sur le nombre de nœuds dans le cluster, le nœud CH crée un calendrier TDMA pour dire à chaque nœud quand est ce qu'il peut transmettre ses données. Ce calendrier est diffusé à tous les nœuds du cluster.

### B) Phase de transmission de données

Une fois que les clusters sont créés et le calendrier TDMA est fixé, la transmission de données commencera. La radio de chaque nœud non-CH doit être mise en veille jusqu'à ce que le temps de diffusion qui lui est réservé commence. Cela minimise la consommation d'énergie dans ces nœuds. Le nœud CH doit garder son récepteur allumé pour recevoir toutes les données des nœuds du cluster. Une fois que toutes les données sont reçues, le nœud CH exécute des fonctions pour agréger la taille de ces données. Ces dernières seront finalement envoyées à la BS. Après un certain temps, qui est déterminé a priori, le prochain round aura lieu et donc le même processus décrit aux sections précédentes va être ré-exécuté.

Les auteurs de LEACH l'ont comparé avec deux protocoles de routage, à savoir, une communication directe avec la BS « *Direct Communication* » et le routage multi-sauts « *Minimum Transmission Energie (MTE) routing* ». Le premier protocole et comme son nom l'indique, il se base sur l'envoi direct des données à la BS par tous les nœuds du réseau. Par conséquent, les nœuds qui sont loin de la BS vont perdre plus d'énergie que ceux qui sont

près. Dans le deuxième protocole, MTE, chaque nœud envoie ses données au nœud suivant tout en rapprochant de la BS. Il est clair que dans le routage MTE, les nœuds les plus proches de la BS sont utilisés pour acheminer un grand nombre de données à celle-ci. Donc ces nœuds vont mourir rapidement, ce qui rend le réseau inactif. En comparant ces deux protocoles avec le protocole LEACH en facteur d'énergie consommée, les chercheurs ont conclu que LEACH réduit la consommation d'énergie 4 à 5 fois que le routage MTE, et 7 à 8 fois que la communication directe. Cela est dû en fait que LEACH permet de garder la durée de vie du réseau tout en partageant la charge de manière équitable afin d'éviter l'affaiblissement de quelques nœuds seulement.

### 4.1.2. Avantages

LEACH présente plusieurs avantages :

- La configuration des clusters se fait indépendamment de la BS (algorithme distribué).
- La rotation du rôle de CH par round permet d'équilibrer l'utilisation d'énergie entre les nœuds du réseau.
- LEACH économise l'énergie, cela est dû en fait que les nœuds de cluster peuvent se mettre en veille hors leurs temps de transmission.
- L'agrégation de données au niveau des CHs permet de réduire la taille des paquets envoyés à la BS et donc de réduire l'énergie dissipée dans la transmission.

### 4.1.3. Faiblesses

Cependant LEACH présente certaines limitations :

- LEACH suppose que tous les nœuds puissent atteindre la BS pour la transmission des données, mais cela est fait avec la dissipation d'une grande puissance d'énergie.
- LEACH choisit aléatoirement la liste des CHs et il ne pose aucune contrainte sur leur niveau d'énergie.
- LEACH n'assure pas une distribution équitable des CHs dans le réseau, par conséquent les CHs peuvent se concentrer dans un même endroit
- La durée des rounds est statique, ce qui pourrait être un inconvénient dans certain cas. En effet, une durée assez longue de round peut épuiser l'énergie des CHs élus. Par

contre, si les CHs possèdent une grande quantité d'énergie, la durée de round peut être augmentée.

- Bien que la rotation du rôle de CH, qui est réalisée par le re-clustering, permette de distribuer la charge entre les différents nœuds, néanmoins, elle cause la dissipation d'une grande quantité d'énergie au début de chaque round.

## 4.2. LEACH-C (LEACH Centralisé)

### 4.2.1. Description

Contrairement au LEACH, LEACH-C [27] est un protocole qui utilise un algorithme de clustering centralisé. Il est basé sur LEACH, et la seule différence entre eux c'est le choix des CHs. Pendant la phase d'initialisation de LEACH-C (*Set-up phase*), chaque nœud envoie des informations sur son emplacement et l'énergie résiduelle à la BS. Cette dernière calcule l'énergie moyenne des nœuds, et les nœuds qui possèdent une énergie résiduelle supérieure à cette moyenne peuvent candidater pour le rôle de CH dans le round courant. La BS trouve les clusters en utilisant l'algorithme de recuit simulé pour résoudre le problème de trouver les K clusters optimaux. Cet algorithme vise à minimiser la quantité d'énergie des nœuds qui ne sont pas CH pendant le transfert de leurs données au CHs. Cela en minimisant la somme des carrés des distances entre tous les nœuds non-CHs et le CH le plus proche.

Une fois les CHs sont trouvés et les clusters sont associés, la BS diffuse un message qui contient le clustering du round courant. Si un nœud reçoit un message avec un ID qui correspond à son propre ID, le nœud se déclare comme un CH, autrement le nœud détermine le temps de transmission de données qui lui est associé et il se met en veille jusqu'à l'arrivée de son rôle de transmission. La deuxième phase (*steady\_state phase*) de LEACH-C est identique à celle de LEACH.

Les auteurs de LEACH-C l'ont comparé avec LEACH, en prenant le nombre de signaux reçus au niveau de la BS comme un facteur de comparaison. D'après les tests, ils ont conclu que LEACH n'est pas efficace que LEACH-C (LEACH-C fournit environ 40% plus de données que LEACH). Cela est due à l'envoi des informations de tous les nœuds vers la BS, par conséquent, la BS va avoir une connaissance globale de l'emplacement et l'énergie résiduelle de tous les nœuds du réseau, de sorte qu'elle peut déterminer l'ensemble de clusters

optimal qui nécessite moins d'énergie pour la transmission de données. Par contre LEACH est un protocole totalement probabiliste.

### 4.2.2. Avantages

- LEACH-C garantit une meilleure distribution des CHs grâce à l'envoi de l'emplacement des nœuds vers la BS.
- Le calcul du nombre optimal de clusters peut produire une meilleure consommation d'énergie et une longue durée de vie du réseau.

### 4.2.3. Faiblesses

- Exactement comme LEACH, le protocole LEACH-C suppose que tous les nœuds peuvent atteindre la BS pour envoyer les données concernant la localisation et l'énergie résiduelle dans la phase d'initialisation, et cela peut entraîner une grande perte d'énergie à cause des messages échangés.
- LEACH-C n'est pas adopté pour les réseaux à grande échelle, cela est dû à la collecte des données par la BS pour avoir une vue globale du réseau.
- De même que LEACH, LEACH-C souffre des problèmes de re-clustering et de durée statique des rounds.

## 4.3. LEACH-C Optimized

### 4.3.1. Description

Comme nous l'avons expliqué précédemment, LEACH-C sélectionne les CHs parmi les nœuds qui disposent d'une énergie au-dessus de la moyenne. Une amélioration de LEACH-C a été proposée pour optimiser encore plus l'efficacité énergétique, qui est le protocole *LEACH-C optimized* [28]. Ce protocole sélectionne un groupe de CHs en utilisant LEACH-C. Ensuite, il calcule la somme quadratique des distances de chaque CH à ses nœuds membres dans la solution optimale. La plus grande consommation d'énergie pour un CH dans

le round suivant sera estimée ( $E_{max}$ ), et tous les nœuds avec l'énergie résiduelle qui est supérieure à  $E_{max}$  seront pris à un nouveau round de recuit simulé pour trouver une meilleure solution. Finalement, la BS diffusera cette solution à tous les nœuds du réseau. Par conséquent, la perte de CH pour chaque round peut être réduite au minimum, et la durée de vie de réseau peut être prolongée en fin de compte.

LEACH-C Optimized a été comparé avec LEACH et LEACH-C. Les résultats ont montré que la répartition des CHs dans LEACH est inégale, autrement dit, la concentration des CHs est plus élevée dans certaines régions. En outre, certains nœuds sont trop éloignés de leurs CHs, ce qui conduira à une perte d'énergie dans la transmission de données. Dans LEACH-C, cette situation a été relativement améliorée. Cependant, LEACH-C *Optimized* surpasse ces deux protocoles et produit un ensemble de CHs presque optimal. Ceci va évidemment réduire la perte d'énergie en communication et donc prolonger la durée de vie du réseau.

### 4.3.2. Avantages

LEACH-C Optimized présente les avantages suivants :

- Il offre une meilleure couverture du réseau grâce à la bonne distribution des CHs, ceci est dû à l'envoi de l'emplacement des nœuds-capteurs exactement comme LEACH-C.
- Il diminue la consommation d'énergie des nœuds-capteurs car l'élection des CHs ayant une meilleure localisation peut diminuer l'énergie consommée lors de la transmission de données.

### 4.3.3. Faiblesses

- Comme les autres protocoles décrits ci-dessus, LEACH-C Optimized impose que les CHs doivent atteindre directement la BS.
- Une énorme perte d'énergie à cause des messages échangés entre les nœuds-capteurs et la BS au début de chaque round.

- Un point commun entre les protocoles qu'on avait décrit jusqu'ici, c'est que la durée du round est toujours fixe. Lorsque la durée du round est grande et la durée de vie est à moitié ou plus, les nœuds qui sont CHs pour ce round risquent de ne pas pouvoir le terminer. Par conséquent ces CHs vont mourir et la BS va perdre le moyen de récupérer les informations à partir de ces derniers.
- Même chose que les protocoles précédents, *LEACH-C Optimized* souffre des problèmes de re-clustering et de durée statique des rounds.

### 4.4. A Dynamic Round-Time Based Fixed LEACH

#### 4.4.1. Description

Ce protocole se montre comme une amélioration du protocole LEACH-F [29,30]. Ce dernier est protocole de clustering centralisé comme LEACH-C, à part que les clusters sont formés qu'une seule fois et ils seront fixés pour toute la durée de vie du réseau. L'avantage majeur de LEACH-F c'est qu'il évite la répétition de la phase de mise en place des clusters « *set-up phase* » au début de chaque round, ce qui mène à l'optimisation des dépenses énergétique et donc à la prolongation de la durée de vie. BS utilise l'algorithme de recuit simulé pour trouver les clusters efficaces, elle diffuse ensuite la liste des CHs et l'ordre de rotation des CHs au niveau de chaque cluster. Le processus de rotation se réinitialise lorsque tous les nœuds-capteurs d'un cluster ont joué le rôle de CH au moins une fois. La phase de transmission de données « *steady-state phase* » est la même que celle de LEACH et LEACH-C.

Bien que LEACH-F résout l'un de désavantages de LEACH et ses descendants, qui est l'évitement du re-clustering, néanmoins il ne traite pas le problème de la durée du round fixe. Pour remédier à ça, le protocole « *A Dynamic Round-Time Based Fixed LEACH* » [31] a été proposé, qui représente une version améliorée de LEACH-F. 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 BS. Autrement dit, la

durée de round est la même dans chaque cluster, mais elle peut différer d'un cluster à un autre.

Pour 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  $E_{non-CH/frame}$  les énergies consommées par frame pour les nœuds CHs et non-CHs respectivement. Ces énergies peuvent être calculées via les formules 2.1 et 2.2.

$$E_{CH/round} = E_{CH/frame} * X_{frame/round} \quad (2.1)$$

$$E_{non-CH/round} = E_{non-CH/frame} * X_{frame/round} \quad (2.2)$$

La méthode pour déterminer la durée de chaque round est d'assurer que l'énergie du nœud-capteur peut durer suffisamment 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 :

$$E_{CH/round} + \left(\frac{N}{K} - 1\right) E_{non-CH/round} = E_{start} \quad (2.3)$$

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

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

En suivant LEACH-F, la formule de round se calcule comme suit :

$$t_{round} = X_{frame/round} * t_{frame} \quad (2.6)$$

Pour pouvoir calculer  $t_{frame}$ , on doit calculer le temps nécessaire pour envoyer un message, cela est fait en suivant cette formule:

$$t_{msg} = \frac{l}{R_b} \text{ secondes} \quad (2.7)$$

Où  $l$  c'est la taille du paquet et  $R_b$  c'est le débit de paquets.

Dans chaque frame, les nœuds-capteurs membres envoient les données à leurs CHs respectifs en parallèle et les CHs ensuite envoient les données agrégées vers la BS. Par conséquent la durée de frame se calcule comme suit :

$$t_{frame} = \frac{l}{R_b} \frac{N}{K} \text{ secondes} \quad (2.8)$$

En remplaçant les équations (2.5), (2.8) dans (2.6), les auteurs de LEACH-F ont déduit la formule suivante :

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

Tous les paramètres précédents sont fixes, par conséquent la durée du round devient fixe. Dans LEACH-F amélioré,  $t_{round}$  est calculée à partir de l'énergie résiduelle  $E_{current}$  à la place de l'énergie initiale  $E_{start}$ .  $t_{round}$  est la même dans chaque cluster mais elle peut varier d'un cluster à un autre. Au début de chaque round, le CH envoie l'énergie résiduelle vers la BS pour qu'elle puisse calculer la durée du round courant. La durée du round est calculée en utilisant la formule suivante :

$$t_{round} = \frac{l}{R_b} \frac{N}{K} \frac{E_{current}}{\left[ \left( E_{CH/frame} \right) + \left( \frac{N}{K} - 1 \right) E_{non-CH/frame} \right]} \quad (2.10)$$

Les auteurs de ce protocole l'ont comparé avec LEACH-F. Ils ont montré que la durée de vie des nœuds a été prolongée à cause de l'application de la formule de round dynamique. Les auteurs ont fait une autre comparaison entre LEACH-F, LEACH et LEACH-C, les résultats montrent que LEACH-F surpasse LEACH et LEACH-C en ce qui concerne l'énergie consommée. Cela est dû au contournement du re-clustering au début de chaque round et donc l'élimination d'un grand nombre de messages échangés. Du coup la durée de vie du réseau va être prolongée.

### 4.4.2. Avantages

- Evitement du re-clustering au début de chaque round et donc réduire la consommation d'énergie au niveau des nœuds-capteurs.
- La durée de round dynamique se présente comme un moyen efficace pour remédier à la mort prématurée des CHs durant un certain round.

### 4.4.3. Faiblesses

- Ce protocole n'est pas flexible car il n'accepte pas les nouveaux nœuds dans le réseau puisque les clusters sont fixés en toute la durée de vie du réseau.
- Il n'est pas tolérant aux pannes car il peut y avoir des CHs dans l'ordonnancement de rotation qui sont défectueux à cause d'un événement extérieur par exemple. En conséquence ces CHs ne peuvent pas accomplir leurs tâches et le réseau devient inopérable du côté de ces CHs.
- Il présente aussi l'inconvénient commun avec les autres protocoles qu'on avait décrit car il suppose également que les CHs peuvent attendre directement la BS pour envoyer leurs données.

### **4.5. Two-tier particle swarm optimization protocol**

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 à un problème. Il s'inspire à l'origine du monde du vivant, plus précisément du comportement social des animaux évoluant en essaim, comme les vols groupés d'oiseaux. L'essaim de particules correspond à une population de solutions appelés particules. Où chaque particule possède une position et une vitesse, ces particules traversent l'espace de recherche pour trouver l'optimum global. Après l'utilisation de l'algorithme de PSO, les chercheurs ont conclu qu'il présente beaucoup d'avantages par rapport aux autres algorithmes alternatifs comme l'algorithme génétique. Ces avantages comprennent la facilité de la mise en œuvre sur le matériel ou le logiciel, des solutions de haute qualité en raison de sa capacité à atteindre la solution optimale et sa convergence rapide.

Dans cette section on introduit le protocole qui est nommé « Two-tier Particle Swarm Optimization for clustering and routing in wireless sensor network ». Depuis son titre, on déduit que ce protocole utilise l'algorithme PSO en deux niveaux, le clustering et le routage. Le protocole de clustering à base de PSO résout le problème de sélection de CHs en élisant les CHs de telle sorte que les clusters formés vont maximiser l'efficacité énergétique, la couverture du réseau et la transmission fiable des données dans le réseau. Ensuite, le protocole de routage trouve les routes optimales qui relient les CHs élus à la BS. Pour le routage, les particules sont codées pour produire une solution complète de l'arbre de routage. Les particules générées que ce soit dans le clustering ou le routage sont soumises à des fonctions de fitness afin de trouver la meilleure particule parmi toute la population. Ce protocole sera décrit et conçu avec d'amples détails dans le prochain chapitre.

*Partie II :*

*Propositions et  
évaluations*

*Chapitre 03 :*

*Etude, amélioration et  
conception d'un  
protocole de clustering*

## Chapitre 03 : Etude, amélioration et conception

### 1. Introduction

Comme nous l'avons déjà évoqué, notre sujet s'inscrit dans le cadre d'un projet au CDTA qui consiste au développement des nœud-capteurs à très faible consommation énergétique. Pour assurer la communication entre ces nœuds et la station de base, les membres de projet ont proposé la catégorie du routage hiérarchique (routage basé sur le clustering). De plus, les protocoles de clustering centralisés sont privilégiés afin de prolonger, encore plus, la durée de vie de réseau. Ceci car ces protocoles donnent une meilleure qualité de clustering en ce qui concerne le choix des CHs, l'affectation des membres aux clusters et la distribution des CHs sur le réseau. Après la synthèse faite, le choix a été porté pour le protocole *TPSO – CR* « Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network ». Ce protocole a été choisi vu qu'il surpasse plusieurs protocoles d'état de l'art comme LEACH, LEACH-C, EEHC, etc.

Tout au long de ce chapitre, nous allons étudier le protocole *TPSO – CR*, en mettant en relief ses avantages et ses inconvénients. Nous allons par la suite proposer et concevoir une version améliorée de ce protocole pour pallier aux inconvénients recensés.

### 2. Présentation et développement du protocole *TPSO – CR*

#### 2.1. Vue générale sur PSO

L'algorithme PSO comporte un essaim de solutions. Chaque solution est représentée par une particule. Ces derniers s'envolent dans l'espace de recherche en se faisant évalués par des fonctions de fitness afin de trouver la meilleure solution parmi toutes les autres.

Initialement, chaque particule «  $i$  » est générée aléatoirement. Au fur et à mesure de déroulement de l'algorithme, chaque particule essaie de trouver la meilleure position pour lui et pour tout l'essaim, en sauvegardant sa meilleure position  $P_{best_i}$  et la meilleure position globale  $G_{best}$ .

Pour générer une nouvelle solution possible (particule), l'algorithme de PSO est basé sur deux concepts : vitesse et position. Au début, la vitesse initiale de chaque particule est

calculé aléatoirement par la multiplication des valeurs de la particule par 0.1, ensuite et au cours de l'algorithme la vélocité se met à jour comme suit :

$$V_{id}(t + 1) = w * V_{id}(t) + c_1 * r_1 * (Pbest_i(t) - X_{id}(t)) + c_2 * r_2 * (Gbest_i(t) - X_{id}(t)) \quad (3.1)$$

Quant à la position, elle se calcule comme suit :

$$X_{id}(t + 1) = X_{id}(t) + V_{id}(t + 1) \quad (3.2)$$

«  $r_1$  » et «  $r_2$  » sont des valeurs aléatoires dans l'intervalle  $[0,1]$ . «  $c_1$  » et «  $c_2$  » sont les facteurs d'apprentissage local et global, respectivement. «  $w$  » est le coefficient qui contrôle la vitesse (vélocité) de la particule.

PSO est un algorithme itératif, qui se répète tant que le nombre d'itération n'a pas excédé le seuil (max\_itération) fixé par l'utilisateur. L'algorithme original de PSO est montré ci-dessous :

**Début**

**Pour** chaque particule faire

    Initialiser la particule aléatoirement ;

**Fin Pour.**

**Tant que** nombre itération < max\_itération faire

**Pour** chaque particule faire

        Calculer fitness;

**Si** fitness > Pbest

            Pbest = fitness;

**Fin de Si.**

**Fin Pour.**

    Choisir la particule avec la meilleur fitness Pbest comme Gbest;

**Pour** chaque particule

        Calculer la nouvelle vélocité via l'équation (3.1);

```
    |
    | |
    | | | Calculer la nouvelle position via l'équation (3.2);
    | | |
    | | | Fin Pour.
    | |
    | | Fin tant que.
    |
Fin.
```

## 2.2. Hypothèses

Le protocole *TPSO – CR* [32] considère un RCSF avec  $N$  nœuds,  $K$  CHs et une seule station de base (BS). Chaque nœud possède un ID unique et l'ID de la BS est égal à 0. Dans la partie de clustering, chaque nœud appartient à un et un seul cluster et chaque CH est le chef d'exactly un seul cluster. Ce protocole suppose que tous les nœuds sont stationnaires après le déploiement.

## 2.3. Principe de fonctionnement de *TPSO – CR*

Dans ce protocole, un algorithme centralisé de PSO à deux niveaux est proposé pour résoudre les problèmes de clustering et de routage dans les RCSFs. Le protocole est nommé *TPSO – CR*, à partir des initiales “Two-tier particle swarm optimization for clustering and routing in wireless sensor network”.

Le terme centralisé veut dire que l'algorithme s'exécute au niveau de la BS. Dans *TPSO – CR*, le temps de fonctionnement du réseau est divisé en cycles (rounds). Chaque round se compose de deux phases, la phase de mise en place et la phase de transmission de données. Dans la première phase, le réseau est configuré, la BS choisira le meilleur ensemble de CHs et les nœuds relais. La phase de mise en place se déroule comme suit :

**1. Découverte de voisinage :** dans cette étape, chaque nœud-capteur dans le réseau diffuse un paquet “HELLO” qui comprend son ID. Un nœud qui reçoit ce paquet met à jour sa table de voisinage avec l'ID inclut dans le paquet et la valeur RSSI correspondante à ce paquet.

**2. Diffusion des données de control :** *TPSO – CR* utilise la méthode d'inondation pour transférer les données de control à la BS. Chaque nœud diffuse ces données sur

lui-même: ID, l'énergie résiduelle et la table de voisinage. Un nœud qui reçoit ce paquet, il le rediffuse jusqu'à ce qu'il atteigne la BS.

**3. Configuration du réseau :** après un délai raisonnable et lorsque la BS reçoit tous les paquets de contrôle à partir des nœuds du réseau, elle commence la configuration. Pour cela, elle utilise le PSO en deux phases. La première est chargé de trouver l'ensemble optimal de CHs et leurs membres du cluster associés. La deuxième trouve les itinéraires optimaux de ces CHs vers la BS (voir sections 2.3.1 et 2.3.2).

**4. Diffusion de la Configuration :** lorsque la BS termine la configuration du réseau, elle utilise à nouveau les inondations pour transférer la configuration vers tous les nœuds. Chaque nœud qui reçoit ce paquet modifie son état pour être soit un CH, un membre du cluster ou un nœud de relais. Un membre du cluster met à jour son CH respectif. Un nœud de relais met à jour son prochain saut vers la BS, tant dis qu'un nœud CH établit un ordonnancement TDMA pour ses membres.

Lorsque la phase de mise en place se termine, la deuxième phase « transmission de données » prend place. Les nœuds non-CHs commencent à envoyer leurs données captées vers leurs CHs respectifs en respectant l'ordonnancement TDMA. Le CH qui reçoit ces données utilise son prochain saut pour acheminer ses données vers la BS. En outre, un nœud non-CH qui termine son transfert de données se met en veille pour économiser l'énergie.

Jusqu'ici, nous avons résumé le fonctionnement général d'un réseau de capteur sans fil exécutant le protocole *TPSO – CR*. Néanmoins, la troisième partie est la plus importante dans ce protocole, à savoir « configuration du réseau ». Cette partie consiste à faire la clusterisation du réseau et la construction des routes entre les CHs et la BS. Ces deux points seront bien détaillés dans ce qui suit.

### 2.3.1. Clustering :

Après avoir reçu tous les paquets de contrôle des nœuds-capteur, la station de base déclenche le premier algorithme, celui de clustering basé sur PSO, pour élire les meilleurs K CHs. Autrement dit, la particule de PSO dans ce cas représente les CHs élus. Ces CHs sont sélectionnés de telle sorte à manière à minimiser le coût d'une fonction objective dont le but est d'optimiser trois critères suivants :

### A) Efficacité énergétique

L'énergie résiduelle d'un nœud-capteur se voit comme un facteur principal pour la sélection des CHs. Un nœud qui possède une meilleure énergie résiduelle est prioritaire pour accomplir le rôle d'un CH. La BS utilise la sous fonction objective suivante pour calculer la fitness d'une particule p par rapport à l'énergie :

$$EE_p = \sum_{k=1}^K \frac{\text{initial } E(CH_{p,k})}{E(CH_{p,k})} \quad (3.3)$$

Tel que  $\text{initial } E(CH_{p,k})$  étant l'énergie initiale des nœuds et  $E(CH_{p,k})$  est l'énergie résiduelle. 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.

### B) Qualité de cluster

Le but de cette fonction est de maximiser la qualité du lien entre les membres des clusters et leurs CHs respectifs. On peut mesurer la qualité du lien par un indicateur qui s'appelle RSSI, cet indicateur représente la force du signal reçu par le nœud-capteur. La qualité du lien entre un membre de cluster  $n_i$  et un CH numéro k dans une particule p peut être calculée comme suit :

$$LQ_{(n_i,CH_{p,k})} = \frac{RSSI_{(n_i,CH_{p,k})}}{\text{minRSSI}} \quad (3.4)$$

minRSSI c'est la pire valeur de RSSI dans toutes les communications entre les nœuds, elle est fixée à -97. A chaque fois que la valeur de LQ augmente, la qualité du lien empire. Afin de maximiser la qualité des liens dans le cluster, la valeur de LQ doit être minimisée. Ceci est assuré par la sous fonction objective suivante :

$$CQ_p = \max_{k=1.2...K} \frac{\sum_{n_i \in C_{p,k}} LQ_{(n_i,CH_{p,k})}}{|C_{(p,k)}|} \quad (3.5)$$

$C_{(p,k)}$  représente le nombre des membres d'un cluster k dans une particule p. La fonction ci-dessus calcule la moyenne des qualités des liens entre les nœuds et leurs CHs dans tous les clusters et prend ensuite la valeur maximale, qui représente le pire des résultats qu'on peut

avoir. Puisque on est face à un problème de minimisation, les CHs qui augmentent la qualité des liens vont être élus.

### C) Couverture du réseau

Afin d'assurer que le protocole choisi garantisse le passage à l'échelle, on doit alors augmenter la couverture du réseau. Cela peut se faire en maximisant le nombre des nœuds clustérisés. Pour arriver à ça, le protocole essaie de minimiser le nombre des nœuds non-clustérisés en respectant la sous fonction objective suivante :

$$NC_p = \frac{N - \sum_{k=1}^K |C_{(p,k)}|}{\sum_{k=1}^K |C_{(p,k)}|} \quad (3.6)$$

Les trois sous fonctions objectives précédentes sont tous réunis pour former la fonction objective globale suivante :

$$C_p = wc_1 * EE_p + wc_2 * CQ_p + wc_3 * NC_p \quad (3.7)$$

$wc_1$ ,  $wc_2$  et  $wc_3$  présentent les contributions des trois sous fonctions objectives dans la fonction objective globale de clustering.

### 2.3.2. Routage

Après la phase de clustering, et pour chaque CH élu on doit générer une route depuis ce CH vers la BS. On peut l'appeler aussi un arbre de routage qui relie tous les CHs avec la BS. Cet arbre, qui est la particule de PSO, doit minimiser le cout de la fonction objective en combinant les propriétés suivantes :

#### A) Efficacité énergétique :

Afin d'assurer l'efficacité énergétique, deux sous-fonctions objectives doivent être respectées:

1. **Economiser l'énergie** : durant chaque round, les nœuds non CHs n'ont pas à être actifs tout le temps, excepté pendant leur temps de transmission ou de réception. Pour

cela, le protocole favorise que les CHs sont prioritaires pour agir comme des nœuds de relai. Pour assurer cela, la sous fonction objective suivante doit être minimisée:

$$EE(a)_p = \frac{R}{C} \quad (3.8)$$

R c'est le nombre de nœuds relais dans la particule et C étant le nombre total des CHs qui jouent le rôle des nœuds relais. D'après la fonction, on déduit qu'à chaque fois la valeur de C augmente, la valeur retournée par la fonction diminue. De cette façon, le nombre des CHs qui agissent comme des nœuds relais va être maximisé.

**2. Consommation équilibrée de l'énergie :** un nœud-capteur avec une meilleure énergie résiduelle est plus chanceux d'être inclus dans l'arbre de routage. La fonction suivante est utilisée pour équilibrer la consommation d'énergie parmi tous les nœuds du réseau.

$$EE(b)_p = \frac{\sum_{i=1}^N E(n_i)}{\sum_{r=1}^R E(RN_{p,r})} \quad (3.9)$$

## B) Qualité des liens

Pour maximiser la réception des paquets, le protocole a besoin de maximiser la qualité du lien entre les nœuds relais dans l'arbre de routage, donc il doit minimiser le pire des liens qu'on peut avoir entre ces nœuds. Cela est réalisable en appliquant la sous fonction objective suivante :

$$LQ_p = \max_{b=1,2,\dots,R} \sum_{\forall rni \in r} \frac{RSSI(rni, rni + 1)}{\min RSSI} \quad (3.10)$$

Après le calcul des sous fonctions objectives présentées ci-dessus, la fonction objective globale qui va être calculé est :

$$R_p = wr_1 * EE(a)_p + wr_2 * EE(b)_p + wr_3 * LQ_p \quad (3.11)$$

$wr_1$ ,  $wr_2$  et  $wr_3$  présentent les contributions des trois sous fonctions objectives dans la fonction objective globale de routage.

## 2.4. Algorithme du protocole *TPSO – CR*

Le protocole *TPSO – CR* fonctionne en cycle (rounds). Dans chacun, les nœuds diffusent deux types de message, le premier est le message « HELLO » pour la découverte de voisinage, tant dis que le deuxième est le message « CONTROL » qui contient l'id de l'émetteur, l'énergie résiduelle et la table de voisinage, ce message doit être rediffusé jusqu'à ce qu'il atteigne la BS. Une fois le traitement de ces deux messages est terminé et que la BS termine aussi son travail qui se résume dans le clustering et le routage, un troisième type de message « CONFIG » sera envoyé depuis la BS vers tous les nœuds du réseau. Ce message contient la configuration optimale pour le réseau, exprimée par la liste des CHs optimaux et leurs membres, ainsi que les routes optimales pour arriver à la BS depuis ces CHs. Une fois la phase de mise en place est terminée, la phase de transmission de données prend place, les nœuds-capteurs commencent à envoyer leurs données captés dans des paquets « DATA » vers leurs CHs respectifs. Les CHs agrègent ces paquets et les envoient au prochain saut jusqu'à ce qu'ils atteignent la BS. Ci-après un algorithme qui résume le fonctionnement général de ce protocole et qui modélise le comportement de chaque nœud  $N_i$ :

**Début :**

**Lors de déploiement du réseau**

Diffuser un message « HELLO » qui contient (ID\_émetteur).

FIN.

**A la réception d'un message de type « HELLO » (ID\_émetteur)**

Table\_voisinage  $\leftarrow$  ID\_émetteur, RSSI ;

FIN.

**A la réception de tous les messages « HELLO » des voisins**

Diffuser un message « CONTROL » qui contient (ID\_émetteur, énergie\_résiduelle, Table\_voisinage) ;

FIN.

**A la réception d'un message de type « CONTROL » (ID\_émetteur, énergie\_résiduelle,**

Table\_voisinage)

**Si (ID\_récepteur  $\neq$  ID\_BS)**

Rediffuser le message « CONTROL » (ID\_émetteur, énergie\_résiduelle,

table\_voisinage);

**Sinon**

**CHs\_optimaux** ← runFirstPSO (infos\_réseau);

// Exécuter la première phase du *TPSO – CR* en utilisant l’algorithme PSO

// pour trouver les meilleurs CHs pour le round courant.

**Routes\_optimales** ← runSecondPSO ( CHs\_optimaux );

// Exécuter la deuxième phase du *TPSO – CR* en utilisant l’algorithme PSO

//pour trouver les meilleures routes pour relier les CHs avec la BS.

**Diffuser un message de type «CONFIG » qui contient (CHs\_optimaux, Cluster\_members, Routes\_optimales) ;**

//diffuser un message qui contient la configuration du réseau.

**Fin Si.**

**FIN.**

**A la réception d’un message de configuration « CONFIG » (CHs\_optimaux,**

**Cluster\_members, Routes\_optimales) ;**

**Déclencher un Timer « ROUND » de durée « Durée\_Round »**

**Si (ID\_récepteur ∈ CHs\_optimaux)**

État ← CH ;

Planifier l’ordonnancement TDMA pour recevoir les données des membres.

Attendre la réception des paquets de tous les membres de cluster.

**Fin Si.**

**Si (ID\_recepteur ∈ Cluster\_members)**

État ← Cluster\_member ;

**Tant que (energie\_residuelle > 0)**

Capter les données ;

Envoyer un paquet « DATA » qui contient (ID\_émetteur, donnée\_captée)  
vers le CH respectif.

**Fin Tant que.**

```
    Fin Si.  
    Si (ID_receiver ∈ Routes_optimales)  
    |  
    | Mettre à jour le prochain saut pour atteindre la BS.  
    Fin Si.  
FIN.  
  
A la réception d'un message de type « DATA » (ID_émetteur, donnée_captée)  
|  
| Agréger les données reçus par les membres.  
| Envoyer les paquets « DATA » agrégés vers le prochain saut.  
FIN.  
  
A l'expiration de Timer « Round »  
|  
| Diffuser un message « HELLO » qui contient l'ID de nœud émetteur.  
FIN.  
FIN.
```

### 3. Analyse du protocole *TPSO – CR* et discussion

Le protocole *TPSO – CR* a été choisi car il présente beaucoup d'avantages par rapport aux autres protocoles qu'on avait présentés dans le chapitre précédent. En effet, dans la partie clustering, les CHs sont élus d'une manière à minimiser la consommation d'énergie au niveau de chaque nœud et maximiser la couverture du réseau. Quant à la partie routage, les routes sont construites de telle sorte à maximiser la réception des données au niveau de la BS. De plus, les auteurs de ce protocole l'ont évalué par simulation et ont prouvé que *TPSO – CR* surpasse six protocoles bien connus d'état de l'art, et ce, par rapport à plusieurs critères de performance.

Néanmoins, d'après l'analyse que nous avons faite autour de ce protocole, nous avons remarqué qu'à chaque round une énergie importante est dissipée par chaque nœud pour effectuer le re-clustering, i.e. pour échanger les paquets « HELLO » et diffuser les paquets « CONTROL » à la BS. Comme nous l'avons expliqué dans la section 2.3 du chapitre 02, l'objectif de re-clustering est de distribuer le rôle de CH équitablement entre les différents nœuds du réseau afin d'éviter l'épuisement des batteries de quelques nœuds. Cependant, le re-clustering cause un gaspillage énergétique remarquable, d'autant plus que les messages

« CONTROL » sont envoyés à la BS par inondation. Bien évidemment, ceci aura une influence négative sur la durée de vie du réseau.

Un autre problème de ce protocole est celui de la durée de round qui est fixe tout au long de l'exécution du protocole. Autrement dit, tous les rounds ont une même durée, fixée par simulation à 500 secondes, et ce, indépendamment de l'énergie résiduelle des CHs. Ceci peut causer la mort prématurée des CHs avant l'expiration du temps de round, ce qui engendre la perte d'information déjà récoltée de leurs membres, qui seront isolés pendant tout le reste de la durée de round.

Ainsi, d'après notre analyse des fonctions objectives d'élection des CHs et du routage proposées dans le protocole *TPSO – CR*, nous avons remarqué que ce protocole n'accorde aucune importance à la distribution des CHs dans le réseau. Autrement dit, les CHs peuvent être concentrés dans une région laissant le reste du réseau non couvert. En outre, lors de l'affectation des membres aux CHs élus, seulement le RSSI est utilisé pour évaluer la qualité des liens entre tous les nœuds et les CHs élus. Lorsque les CHs sont très proches les uns des autres, cela peut résulter en des CHs très surchargés et d'autres presque sans membres. Donc il va y avoir une distribution et une charge non équitables entre les CHs. En effet, le nombre de membres par cluster a une grande répercussion sur la charge du CH en question. Nous avons remarqué également qu'aucune contrainte n'est imposée sur la taille des routes générées entre les CHs et la BS. En effet, le protocole *TPSO-CR* se concentre seulement sur l'énergie résiduelle et la qualité des liens entre les nœuds du relai. Cela peut mener à la génération des routes qui sont très longues. Un CH plus chargé et une route plus longue vont systématiquement épuiser l'énergie des nœuds concernés, ce qui influence négativement sur la durée de vie du réseau.

Pour remédier à ces problèmes, nous avons proposé des techniques qui seront présentées dans la section suivante.

#### **4. Conception de la solution proposée**

En réponse aux deux problèmes surlignés dans la section précédente, nous avons proposé une version améliorée du protocole *TPSO – CR* appelé *ETPSO – CR* pour « Enhanced Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor

network ». L'objectif principal de notre solution est la prolongation de la durée de vie du réseau. Ceci via l'élimination du coût de re-clustering tout en maintenant ses avantages, et l'adaptation de la durée de chaque round aux énergies résiduelles des nœuds.

## 4.1. Elimination du coût de re-clustering

Afin d'éliminer le coût de re-clustering tout en maintenant ses avantages, nous avons proposé une technique basée sur un modèle d'estimation d'énergie dans la station de base. Dans ce qui suit, nous détaillons le modèle d'énergie utilisé et le module d'estimation d'énergie proposé.

### 4.1.1. Modèle Energétique

Nous avons considéré les sources principales de consommation d'énergie dans un nœud-captteur qui sont: le captage, le traitement (calcul et agrégation de données) et la communication radio. Les valeurs énergétiques consommées en captage et calcul seront fixées dans le prochain chapitre. Quant au modèle de dissipation d'énergie de la radio, nous avons adopté le modèle discuté dans [26]. Dans ce modèle, l'énergie consommée dans la transmission  $E_{TX}$  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<sub>0</sub>" alors le modèle « free space » (fs) est utilisé, sinon le modèle « multipath fading » (mp) est considéré. Les équations (3.12) et (3.13) représentent l'énergie consommée dans chaque cas:

$$E_{TX}(l, d) = lE_{elec} + l\varepsilon_{mp}d^4, \text{ if } d > d_0 \quad (3.12)$$

$$E_{TX}(l, d) = lE_{elec} + l\varepsilon_{fs}d^2, \text{ if } d \leq d_0 \quad (3.13)$$

Où  $E_{elec}$  est l'énergie nécessaire pour faire fonctionner le circuit électronique de l'émetteur. "l" est le nombre de bits du paquet transmis.  $\varepsilon_{fs}$  et  $\varepsilon_{mp}$  représentent l'énergie d'amplification dans les modèles fs et mp respectivement, et "d<sub>0</sub>" est la distance de croisement. Celle-ci est calculée par l'équation (3.14) comme suit:

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \quad (3.14)$$

En outre, l'énergie consommée dans la réception  $E_{RX}$  est relative seulement à la taille du paquet reçu. Pour recevoir un paquet de « l » bits, le nœud consomme l'énergie suivante :

$$E_{RX} = lE_{elec} \quad (3.15)$$

#### 4.1.2. Module d'estimation d'énergie

En général un nœud consomme son énergie dans la réception, la transmission, l'agrégation et le captage. L'énergie consommée par un nœud est fortement liée au rôle de ce nœud dans ce round. Dans le protocole *TPSO – CR*, un nœud peut être un CH, un membre d'un cluster, un CH/relai (un CH et en même temps il joue le rôle d'un nœud relai pour assurer le routage), ou membre/relai (un membre d'un cluster et un relai). Donc l'énergie consommée par un nœud « n » dans un round « r » est égale à l'énergie dissipée par ce nœud dans un seul frame multipliée par la durée de ce round. Dans un seul frame, l'énergie consommée par un nœud selon son rôle est estimée comme suit :

1. Un nœud CH consomme l'énergie lors de la réception des paquets de ses membres ( $E_{RX(l)}$ ), l'agrégation de ces données ( $E_{agr}$ ), l'envoi des paquets agrégés vers le prochain saut pour atteindre la BS ( $E_{TX(l,d)}$ ), et le captage fait par lui-même ( $E_{capt}$ ). Cela se résume dans la formule suivante :

$$E_{CH}^1 = E_{RX(l)} * nbr_{mbr} + E_{agr} + E_{TX(l,d)} + E_{capt} \quad (3.16)$$

2. Un nœud CH/relai consomme l'énergie en faisant son rôle en tant que CH, en plus il joue le rôle d'un nœud de relai c'est-à-dire il reçoit des données et il les envoie au prochain saut vers la BS. Dans ce cas, l'énergie consommée est calculée comme suit :

$$E_{CH/relai}^1 = E_{CH} + (E_{RX} + E_{TX(l,d)}) * nbr_{messages\_relayés} \quad (3.17)$$

3. Un nœud membre d'un cluster fait seulement le captage et la transmission des données captées vers son CH respectif, cela est calculé comme suit :

$$E_{membre}^1 = E_{capt} + E_{TX(l,d)} \quad (3.18)$$

4. Un nœud relai consomme de l'énergie pour effectuer ses tâches en tant que membre, en plus, il joue le rôle d'un nœud de relai dans le routage des données vers la BS. L'énergie consommée par ce type de nœud est donnée par la formule suivante :

$$E_{membre/relai}^1 = E_{membre} + (E_{RX} + E_{TX(l,d)}) * nbr_{messages\_relayés} \quad (3.19)$$

De cette façon, la station de base pourra prédire l'énergie consommée ( $E_{consommée}$ ), et donc l'énergie résiduelle au niveau de chaque nœud après l'expiration de la durée d'un round « r ».

$$E_{consommée}(n, r) = E_{consommée}^1(n) * durée\ de\ round \quad (3.20)$$

$$E_{résiduelle}(n, r) = E_{résiduelle}(n, r - 1) - E_{consommée}(n, r - 1) \quad (3.21)$$

Puis, en se basant sur ces informations, un re-clustering du réseau aura lieu au début de chaque round.

#### 4.2. Adaptation de la durée des rounds aux énergies résiduelles

Pour pallier au problème de la durée fixe des rounds, nous avons proposé de l'adapter aux énergies résiduelles des CHs élus. L'idée de base est que la durée de round ne doit pas être très courte au point où l'énergie du nœud ne sera pas exploitée convenablement, ni trop longue pour permettre au nœud d'acter comme membre dans les prochains rounds. Plutôt, elle doit être soigneusement ajustée de telle sorte à optimiser l'utilisation des batteries des nœuds. Pour en faire, nous avons défini un seuil qui se calcule à partir de l'énergie résiduelle de chaque CH multiplié par un pourcentage. La valeur de ce pourcentage sera fixée empiriquement dans la phase des tests. Donc, après la génération de schéma du clustering et du routage au début de chaque round, la BS estime la quantité d'énergie nécessaire à chaque CH pour satisfaire ce schéma pour un seul frame, et ce, en utilisant la formule (3.16). 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 BS refait la même procédure jusqu'à ce que l'énergie résiduelle de l'un des CHs atteigne le seuil. Dans ce cas, la BS diffuse le schéma du clustering et du routage ainsi que la durée de round calculée aux nœuds du réseau. Ci-après un algorithme qui résume cette procédure.

Début :

Pour chaque  $CH_i$  et au début de chaque round

Nombre\_frame = 0 ;

Energie\_Résiduelle\_Test = 0 ;

Seuil = Energie\_Résiduelle ( $CH_i$ ) \* Pourcentage ;

$E_{CH}^1 = E_{RX(l)} * nbr_{mbr} + E_{agr} + E_{TX(l,d)} + E_{capt}$  ;

Energie\_Résiduelle\_Test( $CH_i$ ) = Energie\_Résiduelle( $CH_i$ ) -  $E_{CH}^1$  ;

Tant que (Energie\_Résiduelle\_Test ( $CH_i$ )  $\geq$  Seuil)

Nombre\_frame = Nombre\_frame + 1 ;

Energie\_Résiduelle\_Test ( $CH_i$ ) = Energie\_Résiduelle ( $CH_i$ ) -  $E_{CH}^1$  ;

Fin Tant que

Fin Pour.

Choisir la valeur minimale parmi les Nombre\_frame générées.

Durée\_Round = Nombre\_frame ;

Fin.

### 4.3. Distribution équitable des CHs et équilibrage de charge

Pour faire face au problème de la mauvaise distribution des CHs recensé dans le protocole *TPSO – CR*, nous avons opté pour une modification de la fonction objective du clustering, en ajoutant un quatrième facteur. Ce facteur se montre comme une somme des valeurs de RSSI entre tous les CHs élus. Etant donné que notre problème dans la fonction objective est un problème de minimisation et que la valeur de RSSI retournée soit toujours inférieure à 0, alors, les CHs choisis auront une distance maximale entre eux. D'autres termes, les CHs choisis seront ceux ayant une valeur de RSSI très petite en valeur négative. Cela veut dire que ces CHs ne sont pas proches et ils occupent des régions différentes. Conséquemment, leur charge sera plus ou moins équitable, ce qui permet de prolonger la durée de vie du réseau. La nouvelle formule de fitness (*Enhanced – C<sub>p</sub>*) est montrée dans ce qui suit :

$$C_p = wc_1 * EE_p + wc_2 * CQ_p + wc_3 * NC_p + wc_4 * \text{sumRssiCH} \quad (3.22)$$

Où  $sumRssiCH$  est la somme des valeurs de RSSI entre tous les CHs.

#### 4.4. Minimisation de la taille des routes générées

D'autre part, afin de répondre au problème de *TPSO – CR* concernant la taille des routes générées, nous avons ajouté un quatrième facteur à la fonction de fitness de routage, qui consiste à choisir parmi toutes les routes générées celles qui ayant un nombre minimal des nœuds de relais. Donc on cherche à minimiser la taille des routes pour aller directement à la BS et donc minimiser la quantité d'énergie dissipée. La nouvelle formule de la fonction de fitness (*Enhanced –  $R_p$* ) est montrée ci-dessous :

$$R_p = wr_1 * EE(a)_p + wr_2 * EE(b)_p + wr_3 * LQ_p + wr_4 * sumNbrRN \quad (3.23)$$

Où  $sumNbrRN$  est le nombre de nœuds de relais dans les routes générées depuis une particule.

#### 4.5. Organigramme de la solution proposée

L'algorithme de la solution proposée est le même que celui présenté dans la section 2.4, mis appart que les messages HELLO et CONTROL seront diffusés une seule fois, lors de l'initialisation du réseau. En outre, les durées « *Durée\_Round* » seront calculées via la formule 3.21 et non pas fixée par simulation comme proposé par le protocole *TPSO – CR*. Le fonctionnement général du protocole proposé « *E – TPSO – CR* » est schématisé dans l'organigramme suivant.

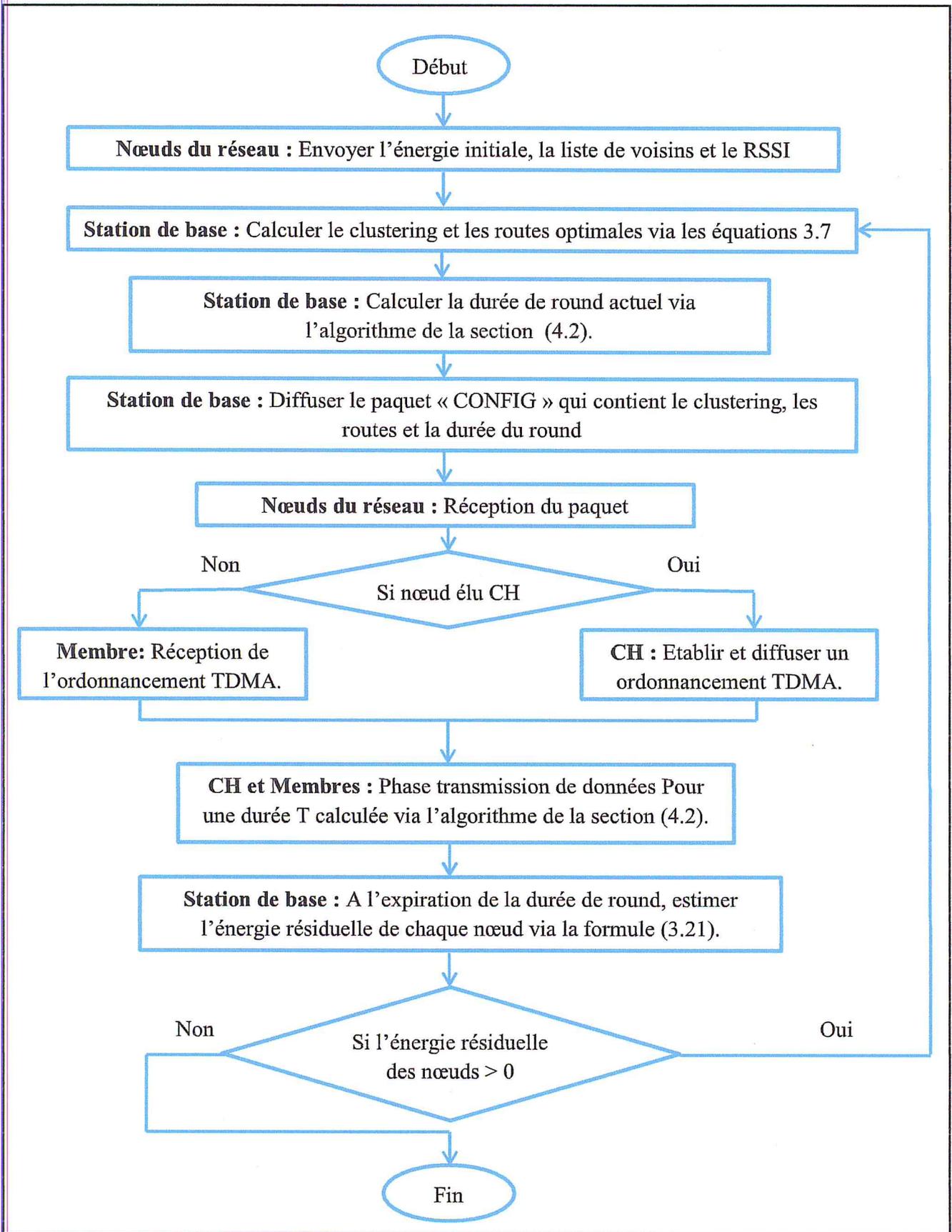
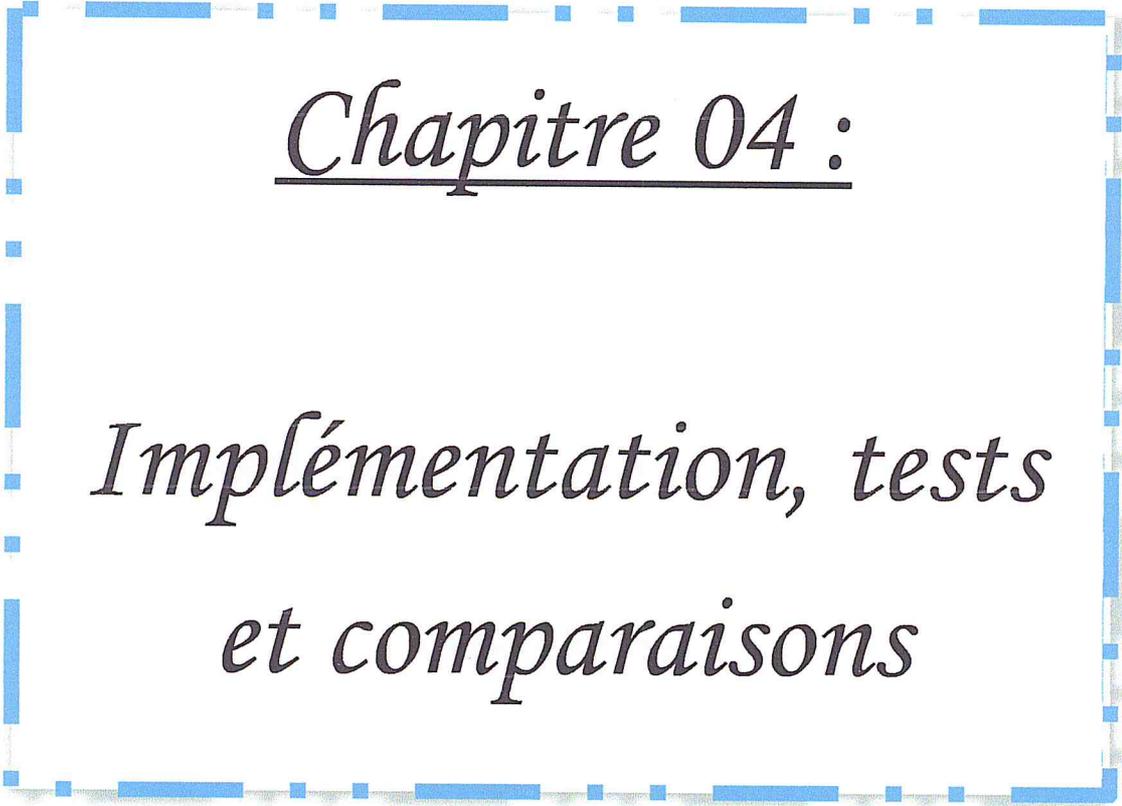


Figure 3.1 : Organigramme de la solution proposée.

## 5. Conclusion

Le protocole *TPSO – CR* représente une bonne solution pour le clustering et le routage hiérarchique dans les RCSFs, vu ses caractéristiques, à savoir sa convergence rapide pour trouver la meilleure solution, son efficacité en terme d'énergie, et sa meilleure couverture du réseau. Au cours de ce chapitre, nous avons développé et analysé ce protocole. D'après l'analyse faite, nous avons pu cerner quatre inconvénients de ce protocole, à savoir, le re-clustering dans chaque round, la durée de round fixe, la mauvaise distribution des CHs dans le réseau et la taille des routes. Nous avons essayé de surmonter ces inconvénients par la proposition de quelques techniques afin d'arriver à une meilleure durée de vie du réseau, par rapport à la version originale de ce protocole. Ceci sera prouvé dans le chapitre suivant, qui présentera les détails de notre implémentation des deux protocoles *TPSO – CR* et sa version améliorée(*ETPSO – CR*), ainsi que leur évaluation et comparaison via le simulateur NS3.



*Chapitre 04 :*

*Implémentation, tests  
et comparaisons*

# Chapitre 04 : Implémentation, tests et comparaisons

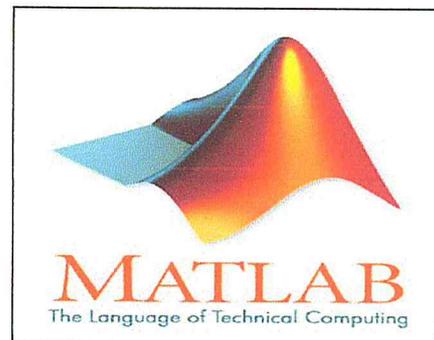
## 1. Introduction

Dans ce chapitre, nous allons présenter l'environnement logiciel utilisé pour la mise en œuvre et les tests des deux protocoles *TPSO – CR* et sa version améliorée (*ETPSO – CR*), ainsi que leurs détails d'implémentation. Après, nous discuterons les résultats obtenus suite à plusieurs scénarios de tests. Enfin, nous finaliserons le tout par une conclusion.

## 2. Environnement expérimental

### 2.1. MATLAB

MATLAB [33] est un système interactif de calcul numérique employable comme une calculatrice et qui dispose d'un grand nombre de fonctions, d'un langage de programmation et d'outils de visualisation graphique. Il est basé sur le calcul matriciel qui permet de traiter les données sans aucune limitation de taille et de réaliser des calculs numériques de façon très rapide.



#### 2.1.1. Pour quelle raison utilisons-nous MATLAB ?

Dans le cadre de notre projet, nous avons à implémenter deux protocoles de clustering et du routage centralisés. Ceci dit que les schémas de clusterisation et même les routes seront générés par la station de base. Les deux protocoles à implémenter sont basés sur la technique d'optimisation PSO. Cette dernière est un algorithme itératif, qui nécessite beaucoup de calcul et qui peut être long dans son exécution. Pour ces raisons, nous avons opté pour le choix du MATLAB pour implémenter le travail de la station de base. En effet, MATLAB est un outil rapide, comparant aux simulateurs réseaux, et il permet d'afficher les résultats en temps réel. Ainsi, MATLAB dispose d'une bibliothèque mathématique très riche et il permet également de faire des visualisations graphiques. C'est pour cela que nous avons jugé l'utilisation de MATLAB un choix judicieux.

## 2.2. NS3 (Network Simulator 3)

La simulation est une étape importante dans le cycle de développement des systèmes embarqués et distribués. Elle met en œuvre un environnement d'expérimentation qui permet de tester à moindre coût les différents protocoles et scénarios. En effet, un simulateur réseau permet de modéliser le comportement d'un réseau, dans notre cas un RCSF, lors d'un déploiement réel. Afin de comparer les performances des protocoles *TPSO – CR* et *ETPSO – CR*, nous avons choisis le simulateur NS-3 [34].



NS3 est un simulateur réseau à événements discrets. Son projet a été créé en 2008 par « Thomas R, Henderson et Sumit Roy », donc il est parmi les plus récents simulateurs. C'est un projet open source programmé en C++ et Python.

Ce simulateur dispose des caractéristiques suivantes :

- ✓ Peut être utilisé sur les plateformes Linux/Unix, OS X(Mac), et Windows (via Cygwin ou une machine virtuelle).
- ✓ Il propose deux langages de programmation: C++ et Python.
- ✓ Très rapide en termes d'exécution (tout est préalablement compilé).
- ✓ Performant par rapport à d'autres simulateurs en termes de gestion de mémoire.

## 3. Implémentation

Notre implémentation a été basée sur plusieurs modules qui permettent d'utiliser et de fournir un ensemble de fonctions. Sachant que, nous avons implémenté deux protocoles : *TPSO – CR* et *ETPSO – CR* (notre amélioration), qui se diffèrent entre eux dans les modules FitnessR, FitnessC (chaque protocole a ses propres modules de calcul de la fonction de fitness pour le routage et le clustering) et Estimation\_Energie, qui est utilisé seulement par le protocole ETPSO-CR.

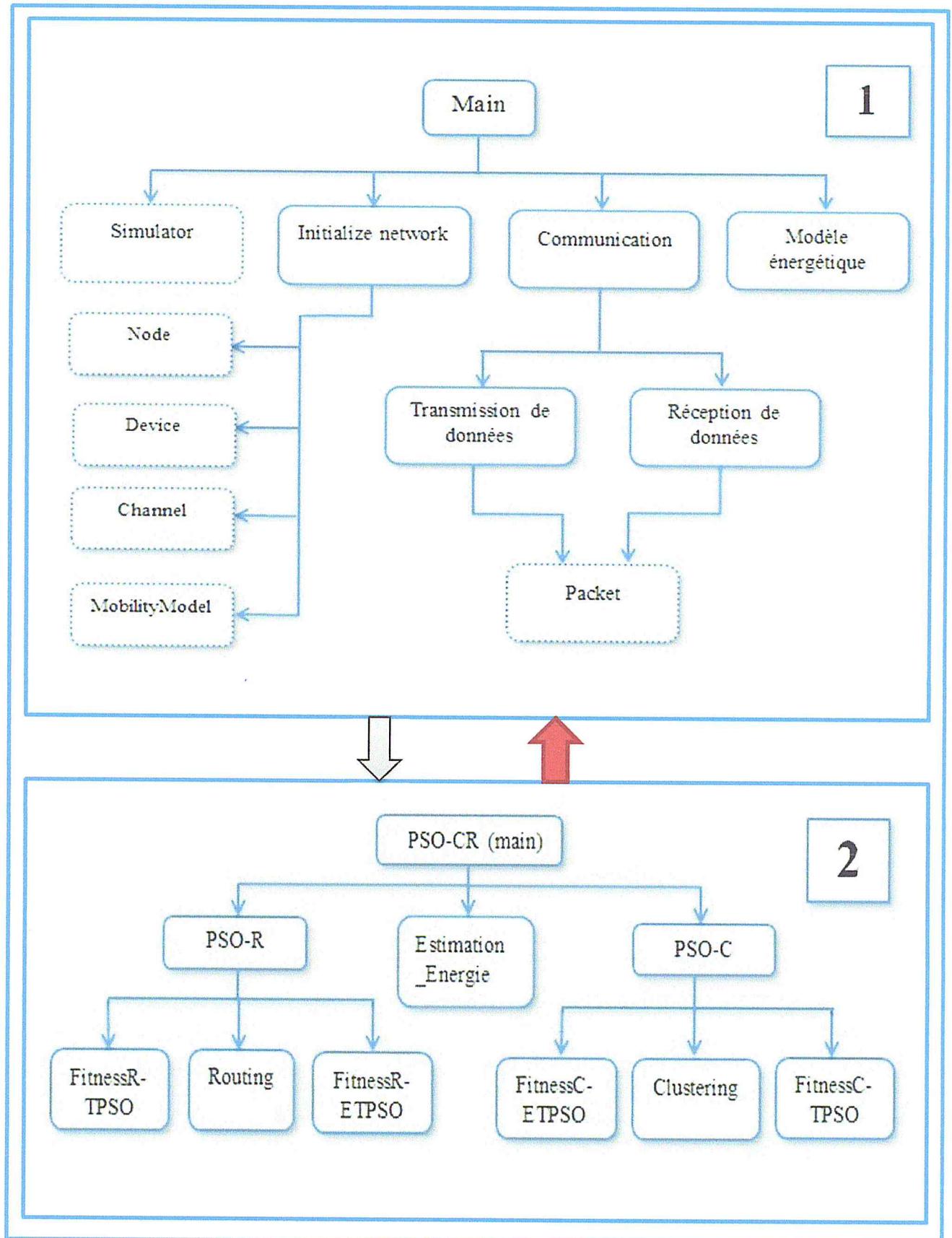
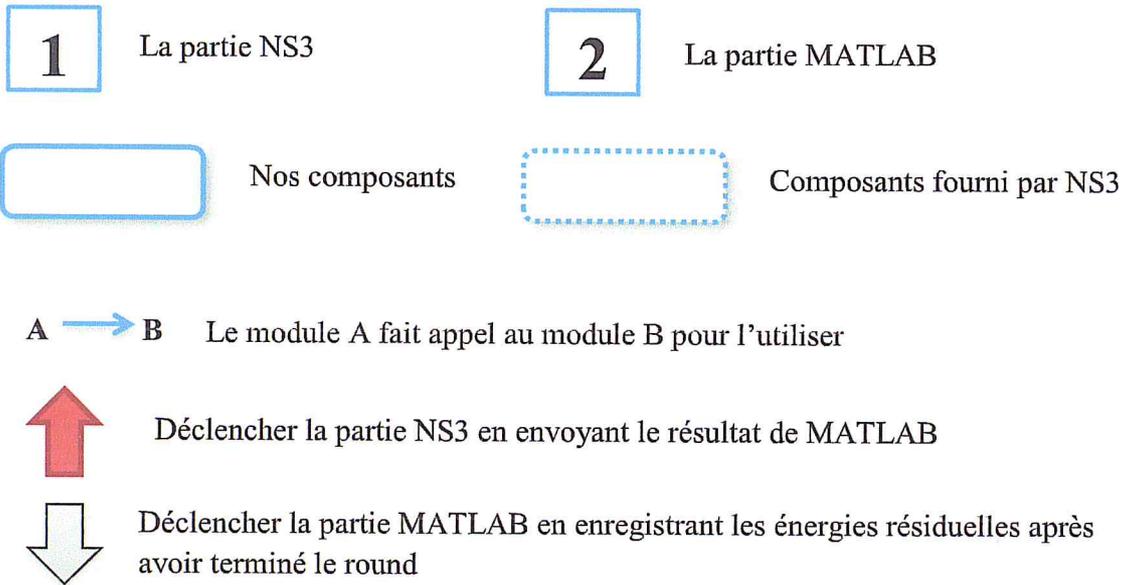


Figure 4.1. Notre architecture d'implémentation

**Légende :**



Dans ce qui suit, une description des différents modules sera présentée:

**1. La partie NS3 :**

- **Main** : Notre fonction principale, elle permet de lancer l'exécution de notre algorithme et tous les autres modules situés au niveau de NS3.
- **Simulator** : C'est le corps de simulation, ce module fournis par NS3 permet de lancer la simulation des évènements que ce soit des transmissions ou des réceptions.
- **InitializeNetwork** : Représente le module principal qui permet d'initier le réseau. Il fait appel aux différents modules qui sont montré dans la figure précédente, à savoir :
  1. **Node** : Ce module permet de créer les nœuds du réseau.
  2. **Device** : Il permet d'installer les cartes réseaux sur les nœuds. De ce fait, nous avons opté pour les cartes qui supporte le protocole de communication LRWPAN (Low Rate Wireless Personnel Area Network) [35], ce protocole a été conçu spécialement pour les réseaux à basse consommation d'énergie, donc il est préférable pour lesRCSFs.
  3. **Channel** : Il représente le module responsable pour initialiser le canal de communication. Ce canal est un moyen pour établir une liaison

entre les nœuds, et il doit être compatible avec le protocole installé sur les nœuds.

4. **MobilityModel** : Ce module est appelé pour enregistrer les positions constantes des nœuds. Ces positions sont générées avec un outil de génération des topologies réalistes qui s'appelle GenSen [36]. Cet outil prend en entrée le nombre de nœuds et la surface et en sortie il génère des positions aléatoires dans un fichier. Les positions enregistrées sont utilisées ultérieurement pour calculer la distance entre les nœuds.
- **Communication** : Ce module comporte les deux phases qui concernent la transmission et la réception de données. Ces deux dernières font appel à un module fournis par NS3 qui s'appelle Packet. En utilisant ce module, nous avons créé cinq types de paquets, à savoir, le paquet « HELLO » pour découvrir le voisinage, le paquet « CONTROL » pour envoyer les énergies résiduelles et la table des voisins vers la BS, le paquet « CONFIG » diffusé par la BS vers tous les nœuds qui contient la configuration de réseau, le paquet « DATA » pour envoyer les données captées depuis les membres vers leurs CHs respectifs, et le paquet « DATA agrégé » celui qui est envoyé depuis les CHs vers la BS.
  - **Modèle énergétique** : C'est le module qui fait le calcul de l'énergie consommée lors de l'envoi ou la réception de données. En effet, puisque NS3 ne dispose pas d'un outil d'analyse d'énergie, nous avons implémenté le modèle énergétique discuté dans la section 4.1.1 du chapitre 3. Ceci pour pouvoir comparer l'efficacité énergétique des deux protocoles implémentés.

## 2. La partie MATLAB :

Cette partie implémente le travail de la BS, i.e. le calcul de clustering/routage et la diffusion de résultat dans le réseau. Elle se déclenche lors de la réception des paquets « CONFIG » qui proviennent des nœuds-capteurs, donc au déploiement du réseau et au début de chaque round. Notre architecture d'implémentation au niveau du MATLAB se résume dans ce qui suit :

- **PSO – CR (main)** : c'est notre module principal qui nous permet d'exécuter l'algorithme de la BS en faisant appel au deux modules essentiels à savoir, PSO-C et PSO-R. Le premier est appelé pour faire la phase de clustering afin de

trouver les meilleurs CHs. Ceci en générant des particules qui sont évalués par le module FitnessC-TPSO dans le cas du protocole TPSO-CR ou FitnessC-ETPSO pour le protocole ETPSO-CR, afin de trouver la meilleure particule. Le deuxième module est appelé pour construire les routes optimales depuis les CHs élus vers la BS. Ces routes sont trouvées lors de la génération des particules. Ces dernières vont être soumises à des fonctions objectives situées dans le module FitnessR-TPSO ou FitnessR-ETPSO (pour les protocoles TPSO-CR et ETPSO-CR, respectivement) pour trouver les routes optimales qui assurent l'acheminement des données vers la BS.

## 4. Tests

### 4.1. Configuration et scénario de tests

Pour pouvoir évaluer les performances du protocole *TPSO – CR* et le comparer avec celui qu'on avait amélioré, nous avons effectué une série de tests. Les résultats de ces tests sont montrés dans la section suivante.

Le simulateur NS3 jusqu'à présent ne dispose pas d'un modèle d'énergie officiel qui est spécifié pour le protocole de communication utilisé, donc on a opté pour l'implémentation du modèle d'énergie discuté dans la section 4.1.1 du chapitre 3. Les différentes valeurs de configuration qui concernent la partie d'estimation d'énergie sont fixées comme c'est montré dans [35]. Les autres paramètres sont définis par rapport à notre implémentation. Par exemple, le simulateur NS3 limite la taille des paquets seulement à 800 bits (100 octets), c'est pour ça qu'on l'a utilisé dans les paramètres à la place de 2000 bits qui a été fixé dans [37]. Ceci car NS3 n'aura pas supporté cette valeur.

Facteur	Valeur
$E_{elec}$	50 nJ/bit
$\epsilon_{fs}$	5 nJ/bit
$\epsilon_{mp}$	10 pJ/(bit. m <sup>2</sup> )
$E_0$	0.25 J
$l$	800 bit
Surface de déploiement	100m * 100m
Densité des nœuds	100, 200, 300

Table 4.1. Paramètres de simulation

Pour la partie MATLAB, celle qui concerne la station de base, c'est la partie dans laquelle on a implémenté l'algorithme PSO pour trouver les CHs, les membres de clusters et les routes optimales. Cet algorithme prend en entrée un réseau de 100, 200 et 300 nœuds, avec un pourcentage de CHs de 5%. Cette valeur a été étudiée et prouvée pour qu'elle soit la meilleure valeur qui nous mène vers de meilleurs résultats [38]. Pour les deux parties clustering et routage, on a considéré une population initiale de 50 particules, et on l'a fait tourner pendant 200 itérations. En ce qui concerne le protocole TPSO-CR, on a fixé les paramètres de l'algorithme PSO comme c'est montré dans l'article du protocole [32]. Cependant, pour notre protocole amélioré, nous avons modifié les valeurs des facteurs des fonctions objectives après avoir fait plusieurs tests pour préciser les meilleures valeurs. Par exemple, pour  $w_{r_1}$  on la mit à 0 car le protocole *TPSO - CR* de base favorise les CHs pour jouer le rôle d'un point de relai. Ceci peut être vu comme un inconvénient car de cette façon l'algorithme peut sélectionner CH situé loin de la BS comme un nœud relai, ce qui augmente la taille de la route générée. Donc, nous avons décidé d'enlever cette valeur et de prendre en considération les autres facteurs seulement. En d'autres termes, on va se soucier seulement du fait que la route nous mène directement vers la station de base en respectant principalement les trois autres facteurs. Les valeurs sont montrées dans le tableau suivant.

	$w_{c_1}$	$w_{c_2}$	$w_{c_3}$	$w_{c_4}$	$w_{r_1}$	$w_{r_2}$	$w_{r_3}$	$w_{r_4}$
<b>TPSO-CR</b>	0.33	0.33	0.33	///	0.33	0.33	0.33	///
<b>ETPSO-CR</b>	0.3	0.2	0.3	0.2	///	0.25	0.25	0.5

*Table 4.2. Paramètres des fonctions objectives.*

Pour la partie NS3, comme nous l'avions déjà dit, nous nous intéressons aux applications de collecte périodique des données. Nous avons procédé par plusieurs tests et scénarios afin de comparer le protocole original avec celui qu'on avait amélioré. Les tests ont été faits sur plusieurs topologies. Les nœuds sont déployés aléatoirement sur une surface de taille (100m\*100m). La densité du réseau est variée entre 100, 200 et 300 nœuds. La BS est fixée avec les coordonnées (50,50), comme le montre le tableau 4.1.

En termes de transmission de données, le nombre de frames est fixé à 100 frames par round pour le protocole original. Quant à notre protocole, au début le nombre de frames est fixé à 100 aussi, car on est sûr que les CHs élus disposent d'une bonne quantité d'énergie, mais au court de l'exécution du protocole et quand la durée de vie du réseau est presque à

moitié, on commence à calculer la durée du round dynamiquement comme le montre l'algorithme discuté dans la section 4.2 du chapitre 3. La valeur du pourcentage déclaré dans l'algorithme est fixée expérimentalement à 0.7, après avoir fait plusieurs tests comparatifs.

Les performances des protocoles TPSO-CR et ETPSO-CR ont été comparées selon plusieurs métriques, à savoir, la qualité de clustering et du routage, l'énergie consommée, le nombre de nœuds vivants, les paquets reçus au niveau de la BS et la durée de vie du réseau. Tous ces tests ont été réalisés sur des nœuds homogènes qui disposent de la même capacité énergétique, et même sur des réseaux hétérogènes où les énergies initiales se diffèrent d'un nœud à un autre. Les résultats obtenus sont présentés et discutés dans ce qui suit.

## 4.2. Résultats, comparaison et discussion

Dans cette section, nous allons évaluer les deux versions implémentées, en exposant les résultats avec leur interprétation. Elle est organisée en deux sous-sections principales, une pour les tests des protocoles dans les réseaux homogènes et l'autre pour ceux hétérogènes. Nous allons conclure cette section par deux tableaux récapitulatifs, un pour les réseaux homogènes et l'autre pour les réseaux hétérogènes. Ces deux tableaux vont nous montrer le pourcentage d'amélioration entre le protocole TPSO-CR et ETPSO-CR par rapport à tous les tests réalisés.

### 4.2.1. Réseau homogène

Tous les tests de cette section sont évalués avec une densité de nœuds varié entre 100,200 et 300 en gardant toujours la même surface (100m\*100m).

#### 4.2.1.1. Qualité de clustering et de routage

Dans cette section, nous allons voir la différence entre les deux protocoles en termes de qualité de clustering et de routage. La *figure 4.2* nous permet de constater que l'algorithme du protocole *TPSO – CR* présente quelques inconvénients à savoir, la génération des CHs qui sont très proches géographiquement ce qui va mener à un déséquilibre sur leur charge, la formation des routes qui sont un peu longues ce qui conduira à une grande perte d'énergie, et la mauvaise couverture du réseau en laissant des nœuds non clustérisés. Cette situation a été améliorée dans notre solution *ETPSO – CR* grâce aux modifications proposées au niveau des fonctions objectives de clustering et du routage. Nous rappelons que ces modifications se

résumé en l'élimination du facteur qui favorise les CHs comme nœuds relais dans la fonction objective du routage, et l'ajout de deux autres facteurs qui assurent l'équilibrage de charge dans les fonctions objectives de clustering et du routage. D'après la *figure 4.3*, nous remarquons que les CHs sont bien éparpillés dans le réseau, une meilleure couverture du réseau qui ne laisse pas des nœuds isolés et une génération des routes optimales sans avoir recours à faire plusieurs sauts pour arriver à la BS. Les deux figures sont montrées ci-dessous pour un réseau de 100 nœuds.

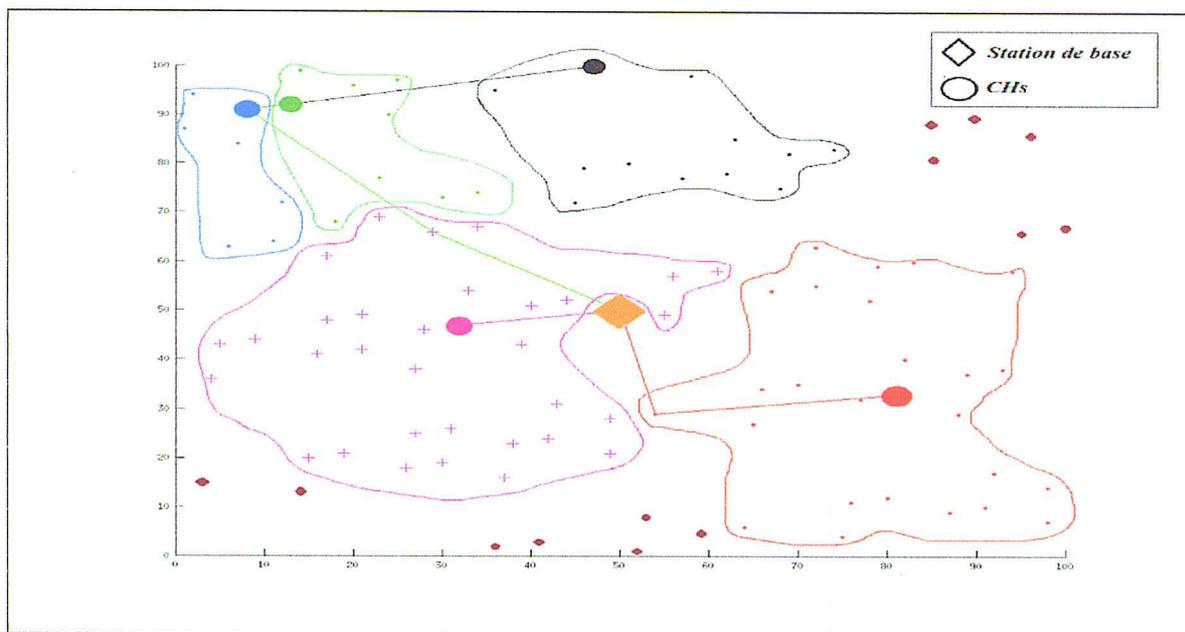


Figure 4.2. Résultat de clustering et du routage du protocole TPSO-CR pour un réseau de 100 nœuds.

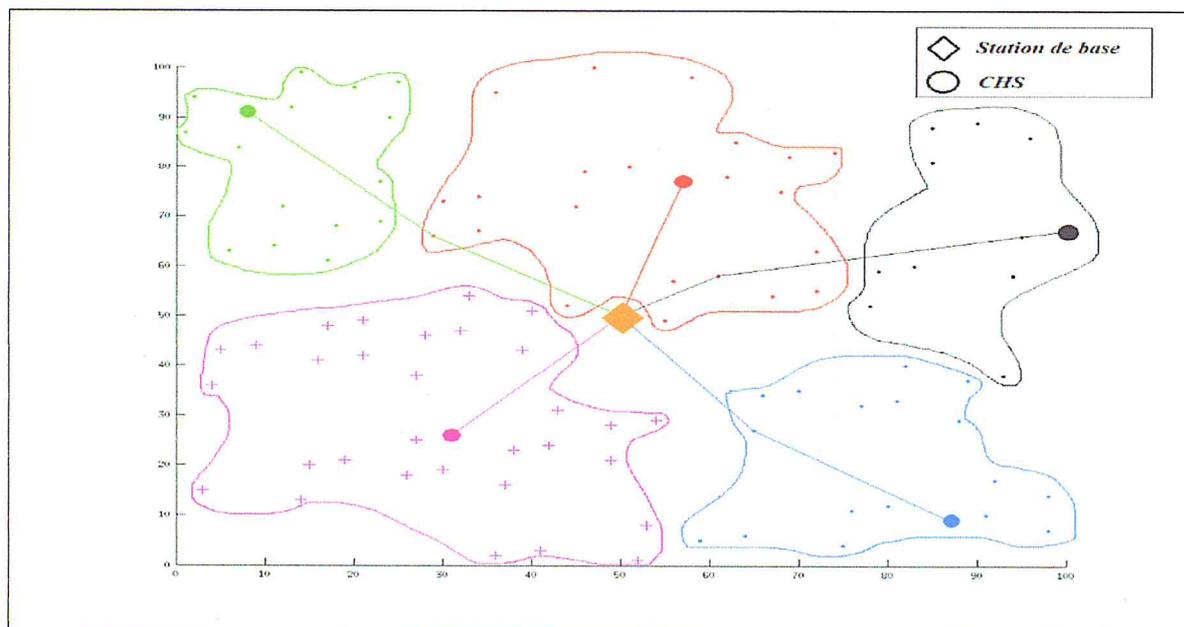


Figure 4.3. Résultat de clustering et du routage du protocole ETPSO-CR pour un réseau de 100 nœuds.

Ci-après un tableau comparatif qui nous montre le nombre de sauts dans toutes les routes générées par rapport au protocole TPSO-CR et ETPSO-CR, et ça pour plusieurs rounds pris comme exemples. Ce tableau nous a permis de constater que notre protocole choisira toujours les meilleures routes en termes d'efficacité énergétique et nombre de saut minimal.

	TPSO-CR	ETPSO-CR	Pourcentage d'amélioration
Round 8	13	5	61.53 %
Round 9	12	2	83.33 %
Round 10	9	6	33.33 %
Round 17	9	4	55.55 %
Round 18	10	6	40 %

*Table 4.3. Pourcentage d'amélioration entre le protocole TPSO-CR et ETPSO-CR par rapports au nombre de sauts.*

#### 4.2.1.2. Efficacité énergétique et durée de vie

##### A) Energie consommée

Dans cette expérience, nous voulons observer le comportement des deux protocoles en termes de l'énergie consommée. Pour cela nous avons tracé un graphe qui illustre la somme de l'énergie consommée par tous les nœuds du réseau, pendant toute la durée de vie, voir *figure 4.4*. D'après cette figure, on voit clairement que notre protocole surpasse le protocole TPSO – CR, et ceci est dû en fait à :

- ✓ L'élimination du coût de re-clustering à cause de l'élimination des messages « hello » et « control » au début de chaque round. Donc une énergie importante va être conservée.
- ✓ La modification apportée au niveau des fonctions objectives, par conséquence la qualité de clustering et du routage s'est améliorée. Car en équilibrant la charge entre les CHs, on va diminuer la quantité de paquets reçus et donc on diminue le cout de la réception, ce qui nous mène vers une consommation d'énergie équitable. Ainsi, la minimisation de la taille des routes permet d'optimiser d'avantage la dissipation énergétique des nœuds relais.
- ✓ La durée dynamique des rounds assure une utilisation optimale des batteries des nœuds.

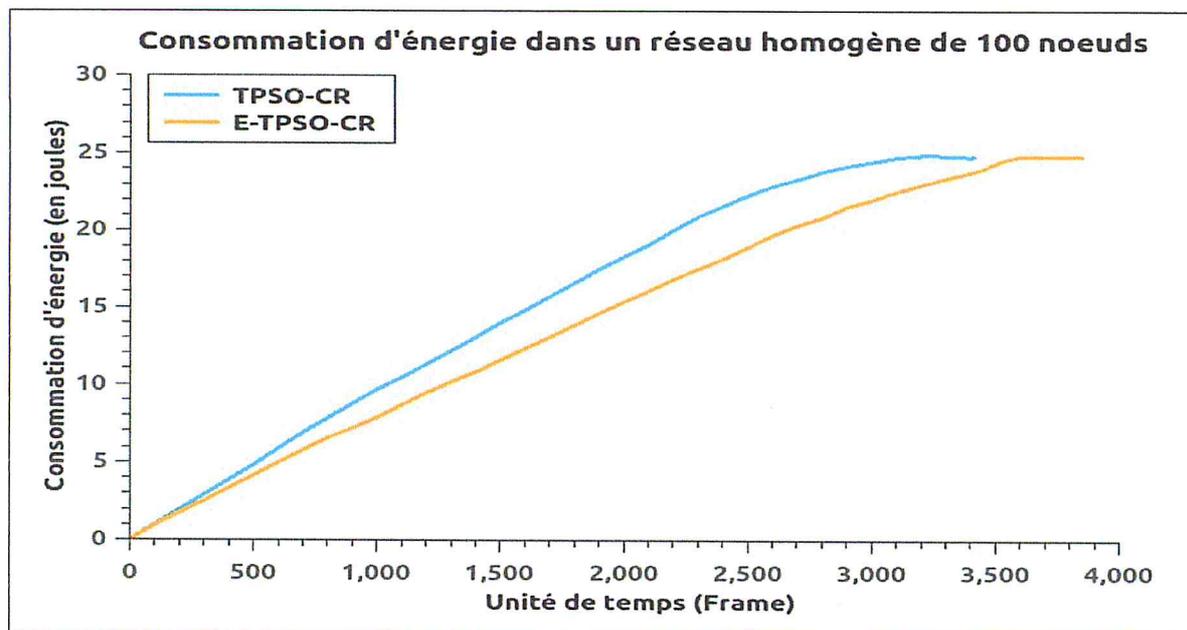


Figure 4.4. L'énergie consommée dans un réseau de 100 noeuds.

Dans ce qui suit, nous allons effectuer le passage à l'échelle (scalabilité). Ce dernier peut être défini comme l'aptitude de faire fonctionner un grand nombre de noeuds tout en maintenant une bonne performance. Pour voir le résultat nous allons interpréter l'effet de l'énergie consommée sur un nombre de noeuds qui égale à 200 et 300. D'après les *figures 4.5 et 4.6*, on remarque que notre protocole dépasse le protocole original, d'ailleurs sa qualité en termes d'énergie consommée se voit clairement quand on passe vers un nombre élevé des noeuds. Ceci est due à l'élimination du coût de re-clustering c'est-à-dire qu'on va éliminer un grand nombre de paquets de configuration puisque on dispose d'une densité très élevée de noeuds. En plus et à l'aide de l'équilibrage de charge entre les CHs, la quantité d'énergie dissipée par tous les noeuds va être diminuée. La conclusion qu'on peut tirer à partir de ces deux graphes est que notre protocole est bien scalable, et que son efficacité énergétique s'accroît proportionnellement à la taille du réseau.

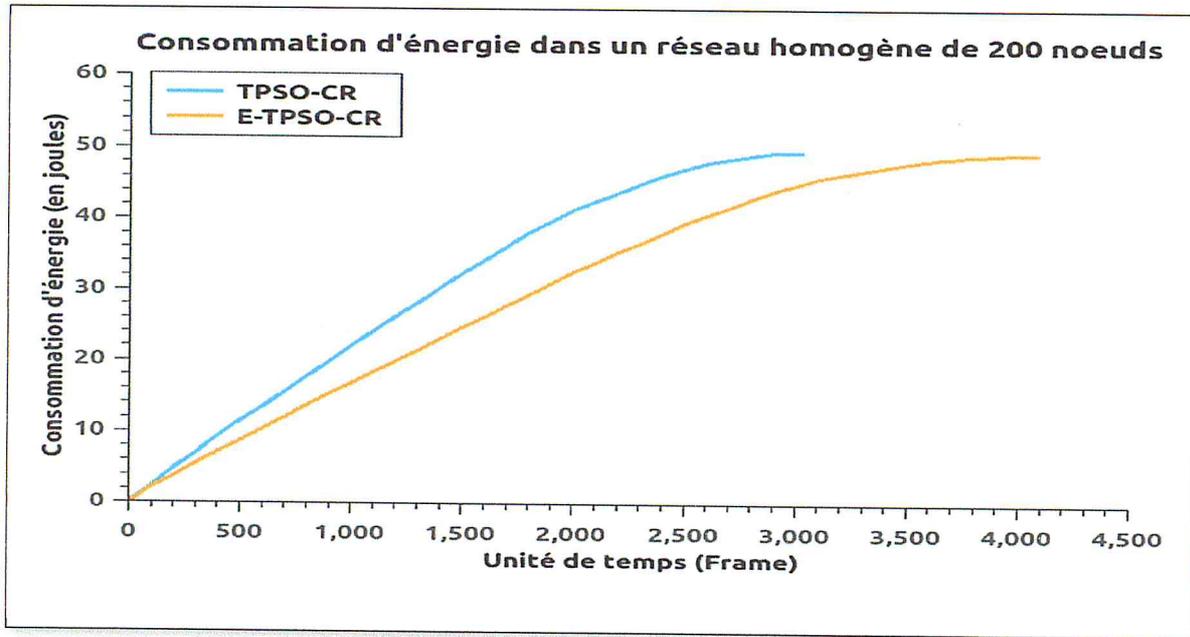


Figure 4.5. L'énergie consommée dans un réseau de 200 nœuds.

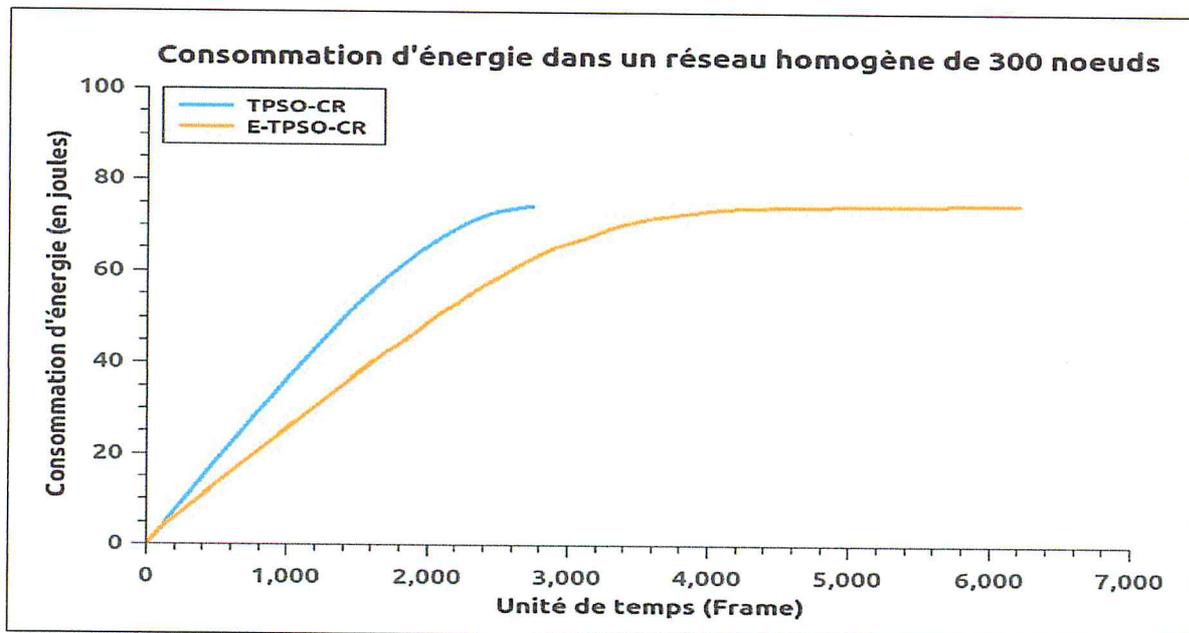


Figure 4.6. L'énergie consommée dans un réseau de 300 nœuds.

### B) Nombre de nœuds vivants

Dans cette section, nous allons présenter le graphe qui nous montre le nombre de nœuds vivants dans le réseau par rapport aux frames. Nous avons ajouté le critère FNP (First Network Partitioning en anglais) qui représente le premier nœud de relai qui meurt au cours de

l'exécution du protocole. La figure nous permette de constater que la courbe de protocole *TPSO - CR* se dégrade rapidement par rapport à notre protocole qui est stable pendant une longue période, et le critère FNP nous montre que tous les nœuds de relais restent en vie pour une période plus grande que *TPSO - CR*. Cela est dû en fait à :

- La phase de l'élimination du coût de re-clustering car en faisant ça on va pouvoir diminuer la quantité d'énergie dissipée au niveau de chaque nœud, contrairement au protocole *TPSO-CR* qui dissipe une grande énergie lors de début de chaque round. Donc les nœuds vont mourir rapidement.
- La durée de round qui se calcule dynamiquement, donc nous allons garder les nœuds en vie pour une longue période contrairement au protocole *TPSO-CR*.

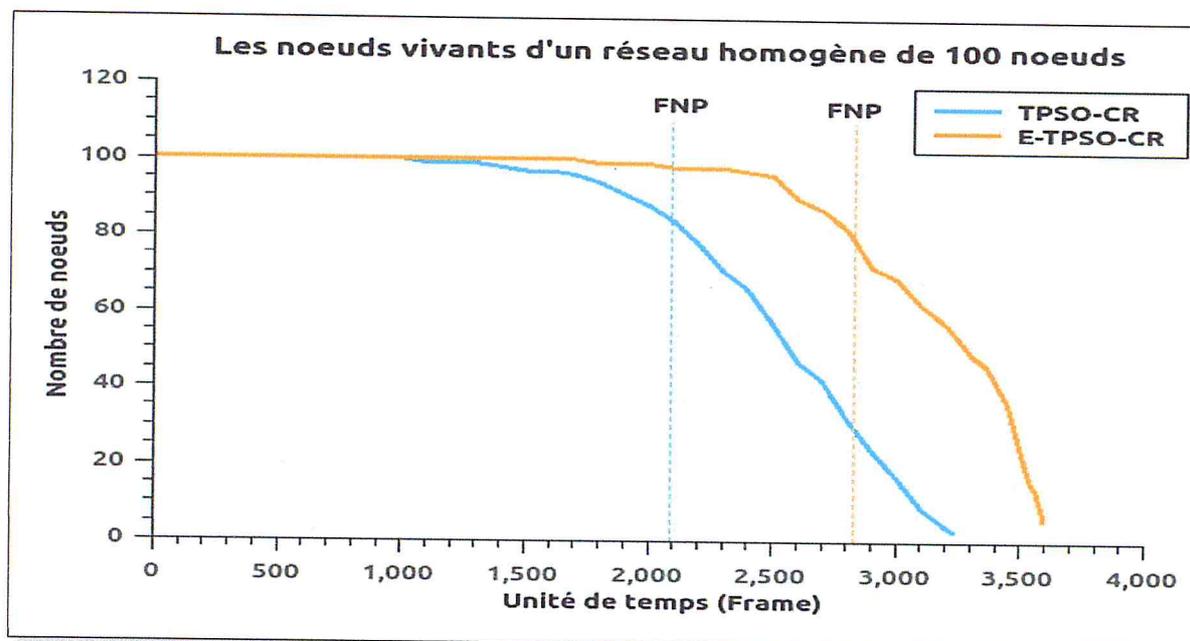


Figure 4.7. Le nombre de nœuds vivants dans un réseau de 100 nœuds.

De la même manière que les tests précédents, on va tenter de voir le résultat sous une densité supérieure (200 et 300 nœuds). En termes de nœuds vivants dans le réseau, on remarque que notre protocole domine le protocole original, la courbe de *ETPSO - CR* se dégrade lentement par rapport à celle de *TPSO-CR*. Ceci car *ETPSO-CR* adopte des techniques qui permettent de lui donner une meilleure durée de vie et une bonne consommation d'énergie.

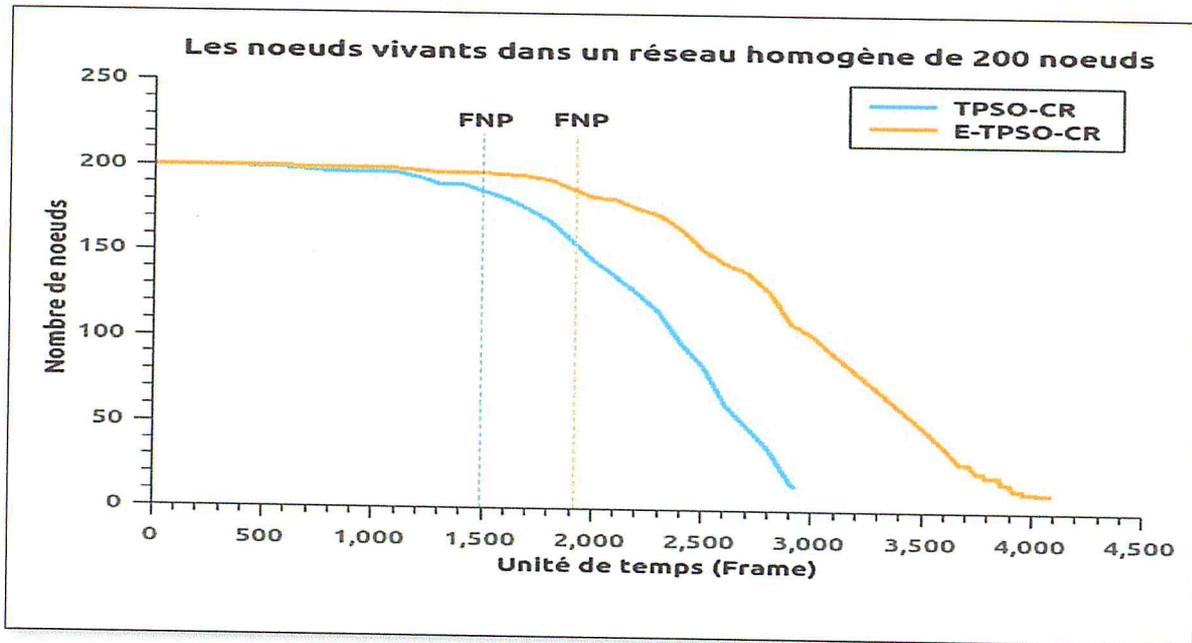


Figure 4.8. Le nombre de noeuds vivants dans un réseau de 200 noeuds.

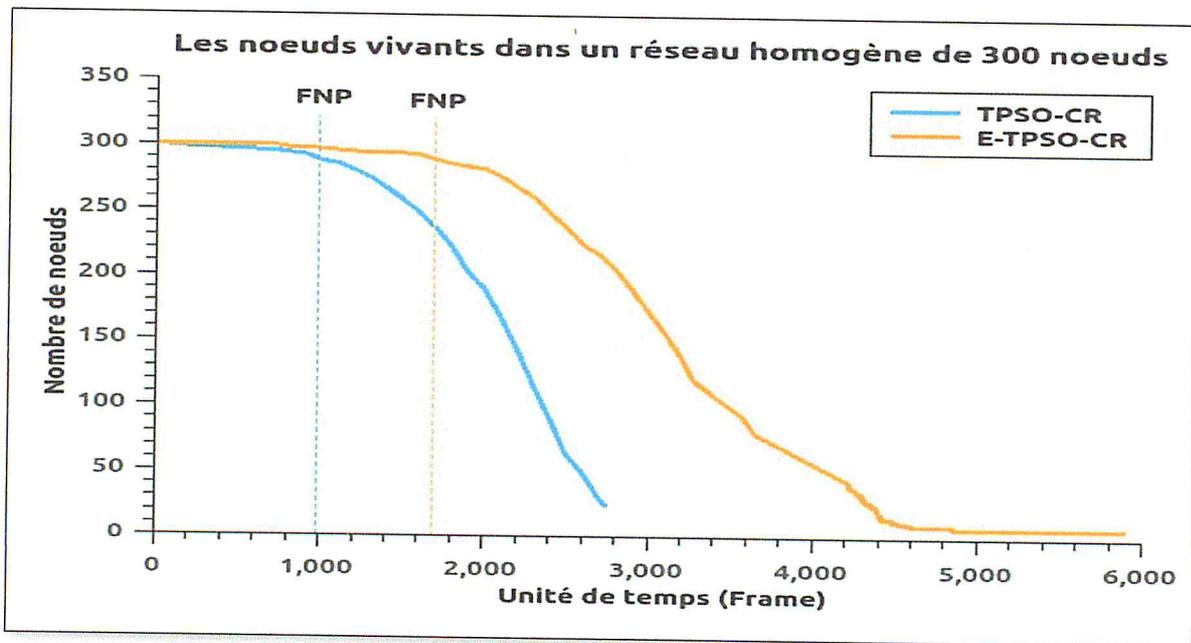


Figure 4.9. Le nombre de noeuds vivants dans un réseau de 300 noeuds.

### C) Durée de vie FND, HND et LNP

Dans cette section, nous allons présenter les graphes de durée de vie du réseau. Celle-ci est considérée comme étant le critère de performance le plus important. Pour examiner la durée de vie du réseau, nous avons opté à l'utilisation des trois critères suivants : FND, HND et LNP.

- FND (First Node Dies en anglais) c'est la période qui sépare le déploiement du réseau et la mort du premier nœud.
- HND (Half Node Dies en anglais) c'est la période qui sépare le déploiement du réseau et la mort de la moitié des nœuds du réseau.
- LNP (Last Network Partitioning en anglais) c'est la période qui sépare le déploiement du réseau et le moment où il devient inopérable.

Les résultats de ces trois critères sont montrés dans la *figure 4.10*. On peut voir à partir de cette figure que la durée de vie est prolongée dans notre protocole contrairement au protocole original. La 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. Dans la partie qui concerne la durée de vie FND, nous remarquons que notre protocole dépasse TPSO-CR, car nous avons éliminé le cout de re-clustering et nous avons amélioré la qualité de clustering et du routage. De plus, dans les autres parties HND et LNP 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 memes performances, cela est montré dans la figure suivante.

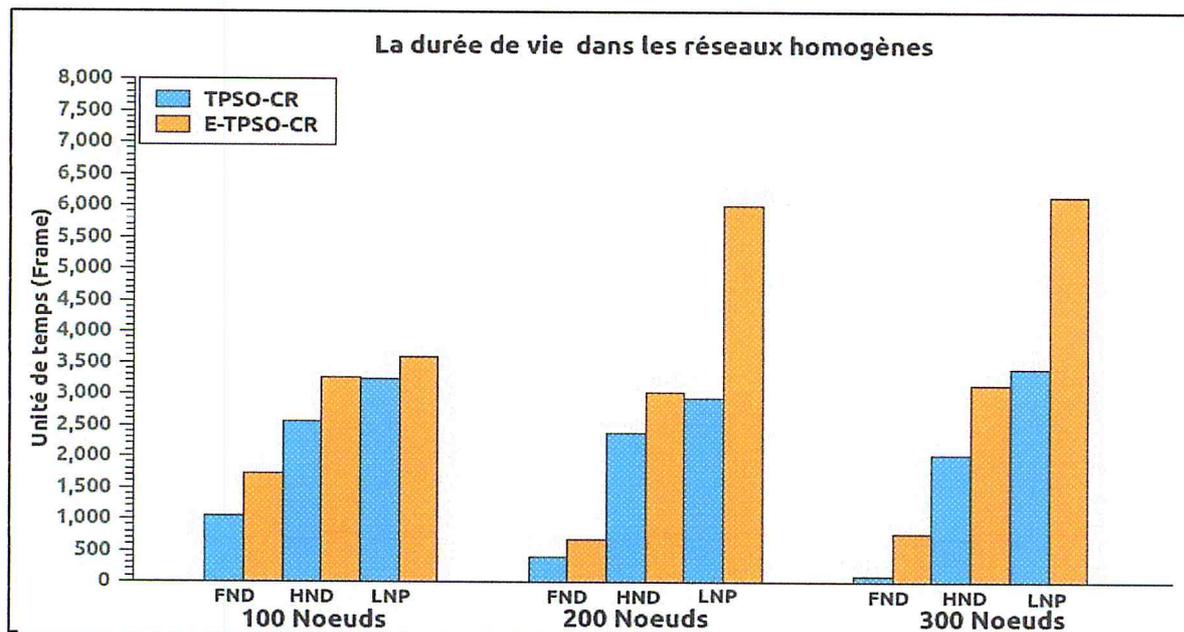


Figure 4.10. La durée de vie dans un réseau de 100,200 et 300 nœuds.

### 4.2.1.3. PRR (Taux de paquets reçus au niveau de la BS)

En arrivant à ce stade-là, on peut dire que ce test va confirmer tous les tests précédents. En fait, ce qui est le plus essentiel pour un RCSF c'est la réception des paquets agrégés au niveau de la BS à tout moment et pendant une longue période que possible. Lorsqu'on assure une bonne consommation d'énergie et une longue durée de vie, on aura par conséquence une bonne réception des paquets agrégés et ça pour une longue durée. La figure suivante représente les paquets reçus au niveau de la BS pour un réseau de 100, 200 et 300 nœuds.

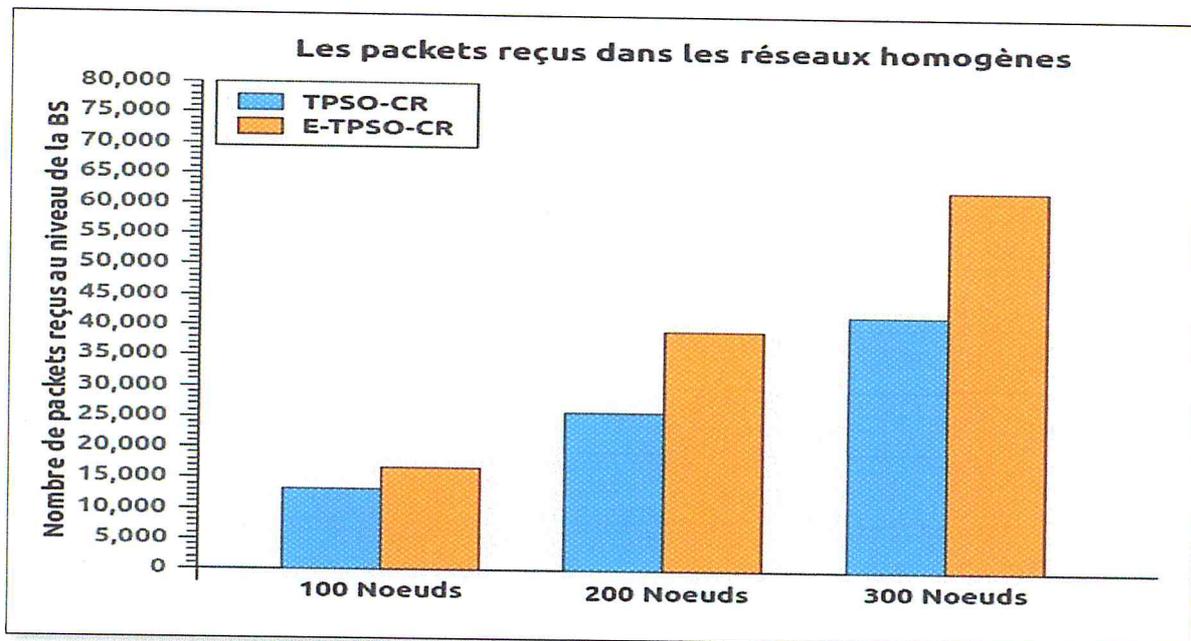


Figure 4.11. Le taux de paquets reçus au niveau de la BS pour les réseaux de 100,200 et 300 nœuds.

A ce niveau, nous pouvons conclure que pour les réseaux homogènes, les contributions qu'on avait proposées dans le chapitre 3 ont bien fait leur travail en ce qui concerne tous les facteurs de comparaisons qu'on avait abordés. Ci-après un tableau qui nous montre le pourcentage d'amélioration entre *TPSO - CR* et *ETPSO - CR* dans les réseaux homogènes par rapport aux facteurs FND, FNP, HND et LNP.

Nous allons par la suite passer notre protocole vers les réseaux hétérogènes afin d'évaluer ses performances dans ce genre de réseaux.

	FND	FNP	HND	LNP	PRR
Pourcentage d'amélioration (100 nœuds)	39,48 %	26,06 %	21,61 %	10,07 %	20,50%
Pourcentage d'amélioration (200 nœuds)	41,45 %	22,09 %	21,47 %	51,29 %	51,92%
Pourcentage d'amélioration (300 nœuds)	88,84 %	41,44 %	35,43 %	44,66 %	33,12%

Table 4.4. Le pourcentage d'amélioration entre le protocole TPSO-CR et ETPSO-CR dans les réseaux homogènes.

#### 4.2.2. Réseau hétérogène

En abordant cette section, nous allons refaire tous les tests précédents avec maintenant des réseaux hétérogènes, c'est-à-dire que les nœuds vont avoir des énergies initiale différentes, aléatoires, rangées dans l'intervalle [0.1, 0.25] et en gardant toujours les mêmes densités de nœuds et les mêmes paramètres qu'on avait fixés.

##### 4.2.2.1. Efficacité énergétique et durée de vie

###### A) Energie consommée

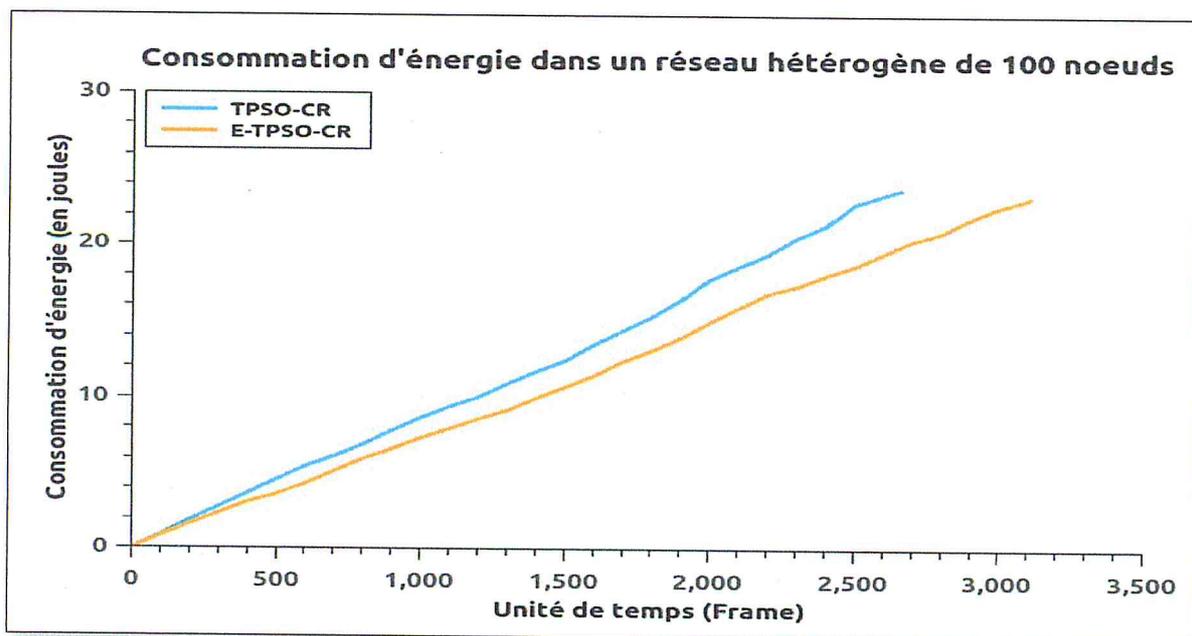


Figure 4.12. L'énergie consommée pour un réseau hétérogène de 100 nœuds.

D'après la *figure 4.12* on constate que notre protocole supporte aussi l'hétérogénéité, et il donne toujours de meilleurs résultats par rapport au protocole TPSO-CR. Pour les réseaux hétérogènes qui disposent d'une densité de nœuds élevée (200, 300 nœuds), notre protocole a également assuré son fonctionnement. Les deux figures suivantes nous montrent à quel point notre protocole est capable de garder son efficacité énergétique.

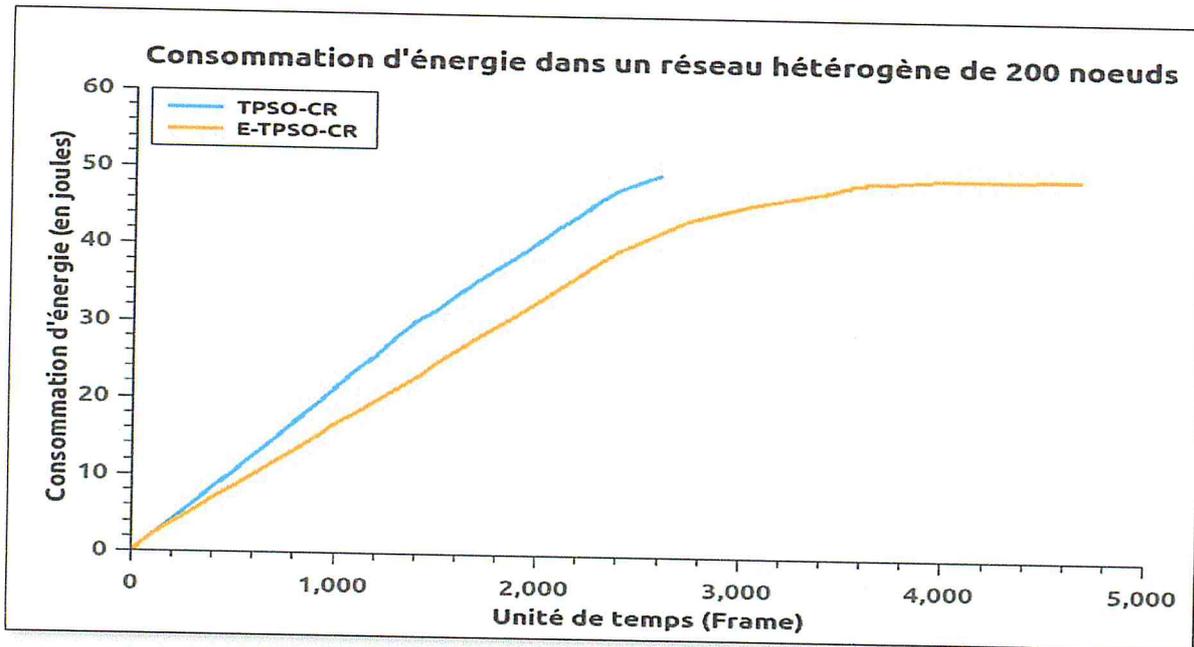


Figure 4.13. L'énergie consommée pour un réseau hétérogène de 200 nœuds.

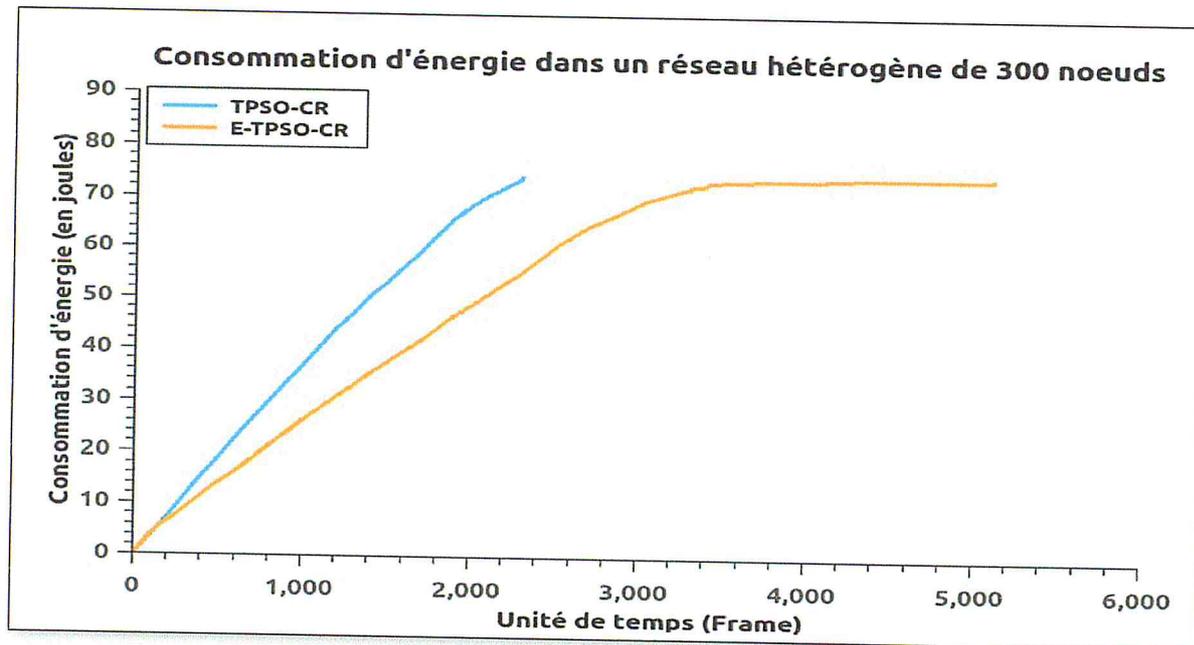


Figure 4.14. L'énergie consommée pour un réseau hétérogène de 300 nœuds.

### B) Nombre de nœuds vivants

Bien évidemment dans les réseaux hétérogènes, les courbes des deux protocoles vont se dégrader car, de base, on a utilisé des énergies aléatoires donc il peut y avoir des énergies qui sont faibles par rapport aux réseaux homogènes qui sont fixés au début à 0.25 J. Malgré cette contrainte notre protocole garde toujours son avantage, et il surpasse le protocole *TPSO – CR* dans le nombre de nœuds vivants au cours de l'exécution du protocole. La figure suivante confirme la supériorité de notre protocole par rapport au protocole *TPSO – CR*.

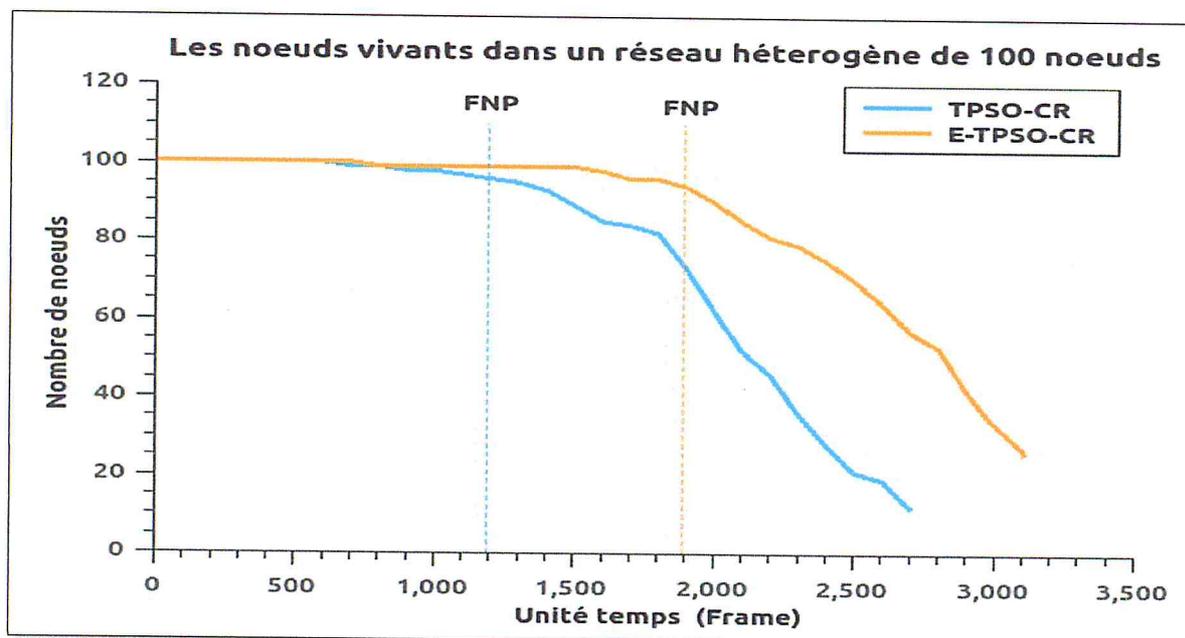


Figure 4.15. Les nœuds vivants dans un réseau hétérogène de 100 nœuds.

Par la suite, nous avons passé notre protocole *E – TPSO – CR* à l'échelle pour voir s'il résiste à l'augmentation de nombre de nœuds vers 200 et 300 nœuds. On a remarqué qu'il présente de bons résultats car il garde les nœuds en vie pour une bonne période bien supérieure que le protocole *TPSO – CR* pour les mêmes raisons déjà expliquées. Les figures suivantes présentent les nœuds vivants dans un réseau hétérogène de 200 et 300 nœuds.

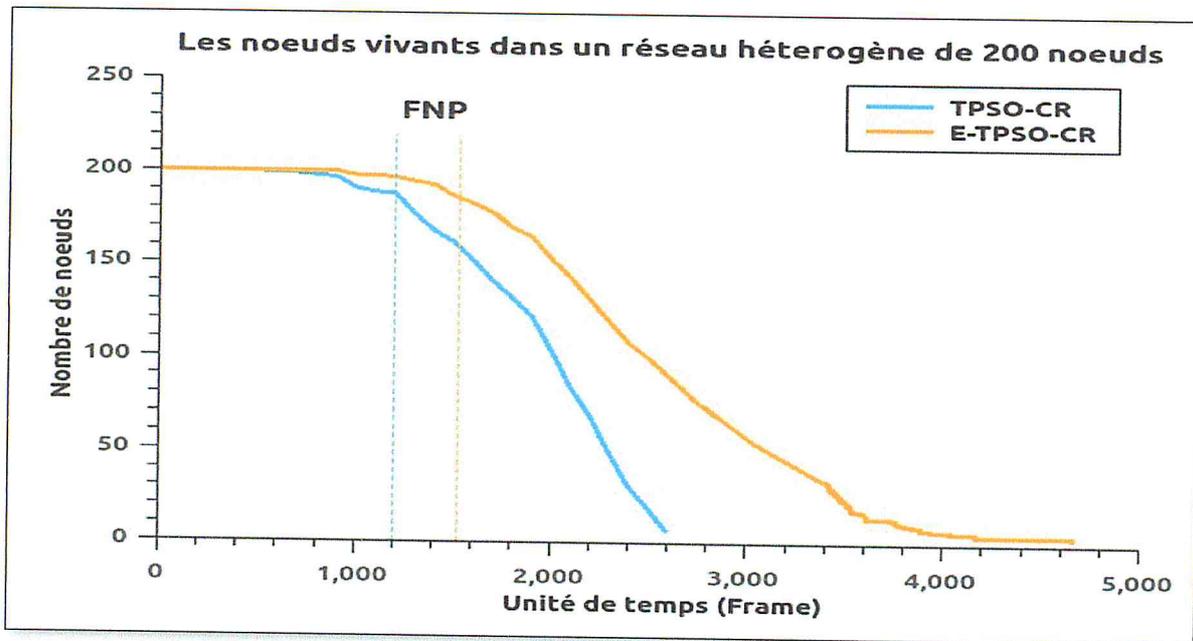


Figure 4.16. Les nœuds vivants dans un réseau hétérogène de 200 nœuds.

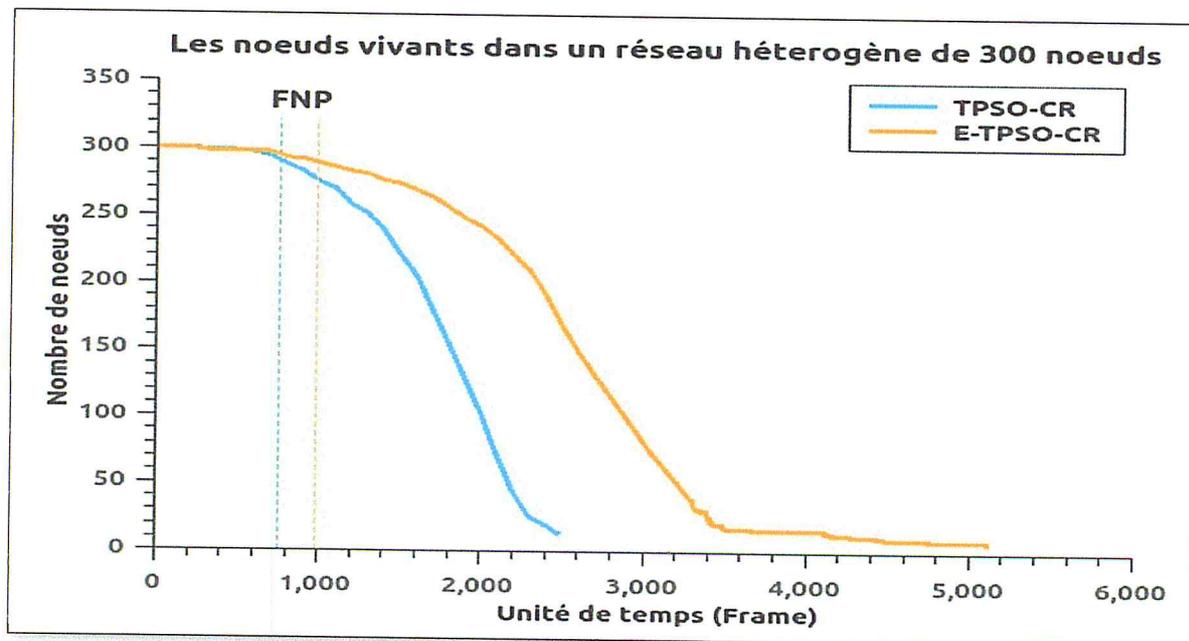


Figure 4.17. Les nœuds vivants dans un réseau hétérogène de 300 nœuds.

### C) Durée de vie

Dans cette expérience, nous avons réalisé des tests qui concernent la durée de vie pour les réseaux hétérogènes. Puisque notre protocole a surmonté le protocole *TPSO – CR* dans la consommation d'énergie et le nombre de nœuds vivants, donc il va certainement dépasser le protocole *TPSO-CR* en termes de durée de vie du réseau, ce qui est clairement observé dans la figure suivante.

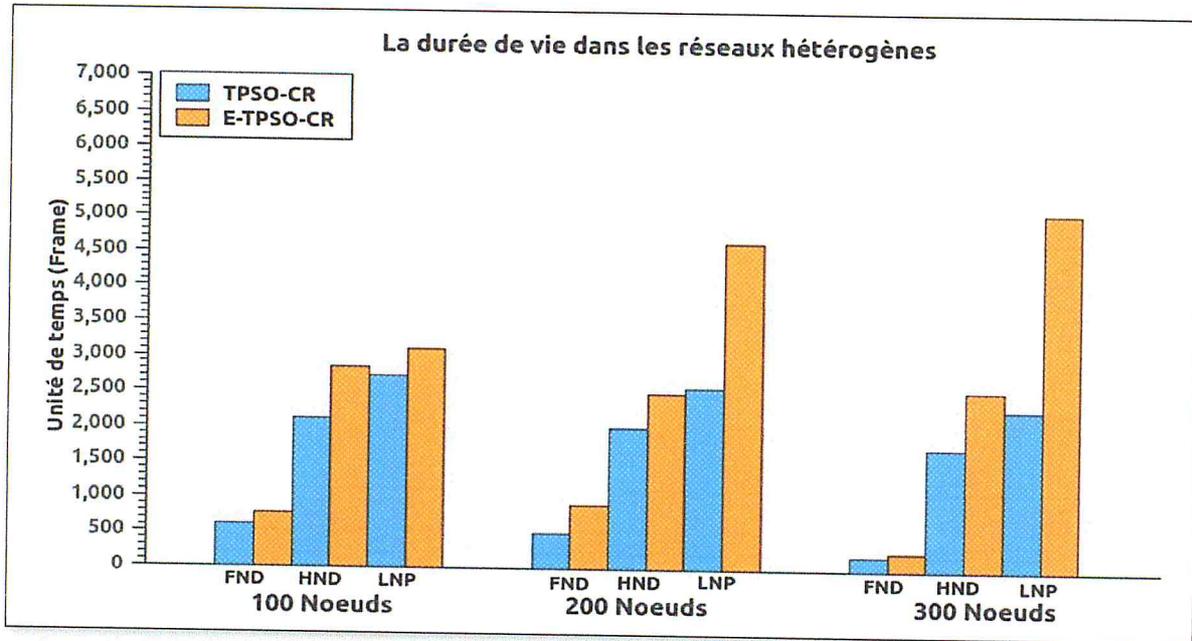


Figure 4.18. La durée de vie dans un réseau hétérogène de 100, 200 et 300 nœuds.

#### 4.2.2.2. PRR (Taux de paquets reçus au niveau de la BS)

Pour finaliser les tests des réseaux hétérogènes, on doit vérifier le facteur PRR. La figure suivante nous montre que le protocole ETPSO-CR dépasse toujours le protocole TPSO-CR en termes de nombre de paquets reçu au niveau de la BS, même dans les réseaux hétérogènes.

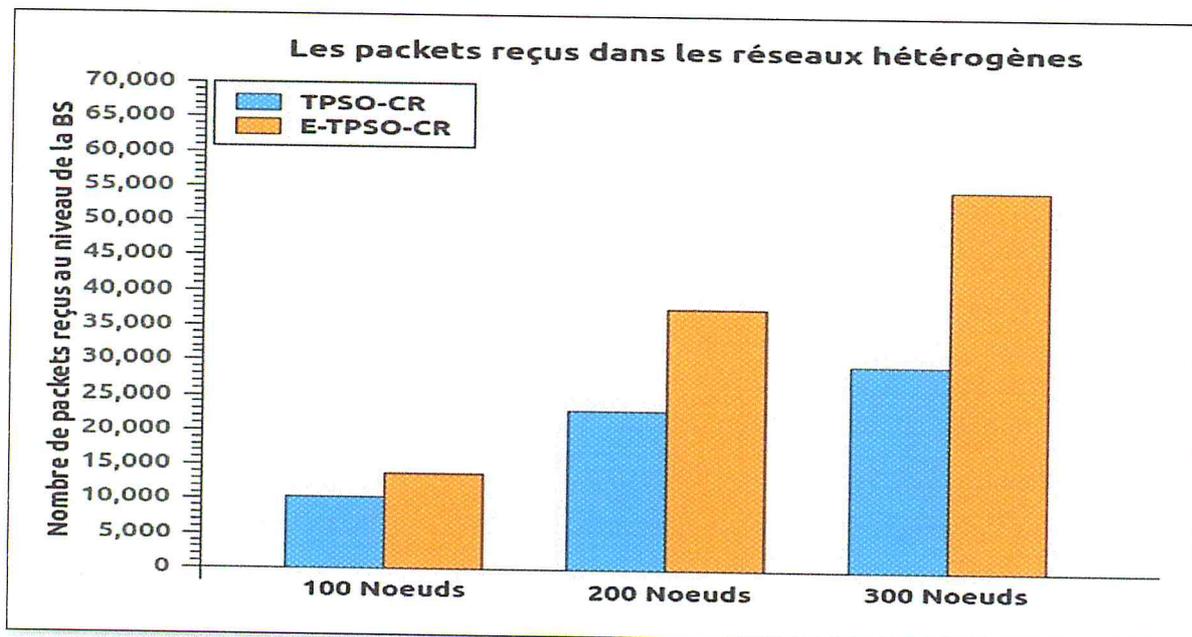


Figure 4.19. Le taux de paquets reçus au niveau de la BS dans un réseau hétérogène de 100, 200 et 300 nœuds.

Ci-après un tableau qui nous montre le pourcentage d'amélioration entre *TPSO – CR* et *E – TPSO – CR* dans les réseaux hétérogènes par rapport aux facteurs FND, FNP, HND et LNP.

	FND	FNP	HND	LNP	PRR
<b>Pourcentage d'amélioration (100 nœuds)</b>	22,94 %	36,83 %	25,45 %	12,04 %	25,33%
<b>Pourcentage d'amélioration (200 nœuds)</b>	45,72 %	21,36 %	19,70 %	44,43 %	38,89%
<b>Pourcentage d'amélioration (300 nœuds)</b>	19,60 %	23,28 %	32,11 %	54,91 %	35,82%

Figure 4.20. Le pourcentage d'amélioration entre le protocole *TPSO-CR* et *ETPSO-CR* dans les réseaux hétérogènes.

## 5. Conclusion

Au fil de ce dernier maillon de notre travail, nous avons présenté l'implémentation du protocole *TPSO-CR* ainsi que sa version améliorée, intitulée *ETPSO – CR*, que nous avons proposé. Nous avons d'abord implémenté le travail de la BS au niveau du MATLAB, ensuite nous avons passé à NS3 pour simuler le comportement des nœuds lors de l'envoi et la réception des données. Suite à une série d'expériences et des tests, nous nous sommes arrivés à des résultats très satisfaisants. En effet, les résultats montrent clairement que notre protocole surpasse le protocole *TPSO-CR* par rapport à toutes les métriques étudiées. De plus, nous avons prouvé que notre protocole garde toujours son efficacité même pour les réseaux à forte densité et les réseaux hétérogènes.

## *Conclusion générale et perspectives*

Depuis quelques années, les avancées technologiques en termes de miniaturisation des machines et des supports de communication ont rendu envisageable le déploiement et l'exploitation d'un grand nombre de nœud-capteurs, organisés en réseau. Les réseaux de capteurs sans fils ont été identifiés comme l'une des dix technologies clefs de l'avenir et ce en raison de l'incroyable potentiel applicatif qu'elle renferme. Cependant cette technologie reste relativement récente et soulève d'importantes problématiques de recherche en matière d'organisation, de communication, de gestion de la consommation d'énergie et d'exploitation des données récoltées, etc.

Le clustering dans les réseaux de capteurs sans fil est un moyen efficace pour l'optimisation de la consommation d'énergie, et donc la prolongation de la durée de vie, surtout dans les réseaux à très grande échelle.

Le travail réalisé dans ce mémoire entre dans le cadre d'un projet au CDTA qui consiste au développement des nœud-capteurs à très faible consommation d'énergie. Nous avons donc étudié dans un premier temps les réseaux de capteurs sans fils en mettant en reliefs leurs concepts, leurs caractéristiques, leurs défis de déploiement, et les différents domaines d'applications qui ont été envahi par les RCSFs. Dans une deuxième partie, nous avons présenté les concepts de base de clustering et nous avons exposé quelques protocoles de clustering. A partir de la troisième partie, nous avons consacré nos efforts pour étudier le protocole TPSO-CR qui a été choisi vu son efficacité énergétique. Au cours de l'analyse de ce protocole, nous avons constaté qu'il présente quelques inconvénients. Par conséquent, nous avons proposé des techniques pour pallier aux inconvénients recensés, dont l'objectif principal est l'amélioration de la durée de vie. A la fin nous avons mené des simulations extensives afin de comparer les performances du protocole TPSO-CR avec la version proposée ETPSO-CR. Les résultats prouvent que nos propositions ont amélioré les performances du protocole TPSO-CR par rapport à toutes les métriques étudiées. Dans certains scénarios, l'amélioration a atteint 88,84 % par rapport à la durée de vie FND, 51,29 % pour LNP et 51,92% en ce qui concerne le taux de paquets reçus dans la station de base.

Ce projet de fin d'étude, étant une première expérience intéressante, qui nous a permis d'une part, de concrétiser nos connaissances théoriques acquises pendant le cursus universitaire, et d'autre part, d'améliorer nos compétences dans le domaine des réseaux en général et particulièrement les RCSFs.

Nos perspectives de recherche sont :

- La proposition d'une version qui assure la tolérance aux pannes des micro-capteurs étant la perspective principale de ce travail. 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 ETPSO-CR peut assigner des rôles importants (CH ou relai) aux nœuds qui sont devenus hors service. Pour remédier à cela, une technique efficace de tolérance aux pannes doit être mise en œuvre. Cela en trouvant un moyen pour détecter les nœuds qui sont tombés en pannes.
- Il est à noter que l'avantage de notre solution est la prolongation de la durée de vie du réseau, néanmoins, elle représente une approche approximative basée sur l'estimation d'énergie. Le modèle d'estimation utilisé, bien qu'il soit le plus répandu dans les RCSFs, ne modélise pas certaines sources de consommation d'énergie comme l'écoute passive (idle listening), la retransmission en cas de collisions, etc. L'utilisation d'un modèle d'estimation énergétique plus réaliste s'avère comme perspective intéressante.
- Il serait aussi intéressant d'étendre la conception du protocole ETPSO-CR pour supporter la mobilité des nœud-capteurs. Bien évidemment, cela n'est possible que pour le type d'application où les nœuds se déplacent en suivant un modèle de mobilité déterministe.

## *Références bibliographiques*

- [1]: K.Beydoun. « Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs »Thèse de doctorat, Spécialité : Informatique, l'u.f.r des sciences et techniques de l'université de Franche-Comté, 2009.
- [2]: Akyildiz, I.F.,W.Su,Y.Sankarasubramaniam, E.Cayirci, « A Survey on Sensor Networks »,IEEE Communications Magazine, August, 102-114(2002).
- [3]: QinghuaWang et Ilangko Balasingham, « Wireless sensor Network- An introduction » Wireless Sensor Networks : Application-Centric Design, Geoff V Merrett and Yen Kheng Tan(Ed.), ISBN : 978-953-307-321-7, InTech, 2010.
- [4]: B.Krishnamachari, « Networking Wireless Sensors », Cambridge University Press, 2005.
- [5]: Ayse Coskun. CSE 237A Platforms.[en ligne].  
<http://cseweb.ucsd.edu/classes/wi08/cse237a/handouts/L5-platformdev.pdf>  
[07/06/2016]
- [6]: Wikipédia. (2016). Réseau de capteur sans fil. [En ligne]  
[https://fr.wikipedia.org/wiki/Web\\_des\\_objets](https://fr.wikipedia.org/wiki/Web_des_objets) , [6 juin 2016].
- [7]: Ian F. AKYILDIZ Mehmet Can VURAN, “Wireless Sensor Networks”, John Wiley & Sons Ltd, 2010.
- [8]: A. El-Hoiydi, “Spatial TDMA and CSMA with preamble sampling for low power ad hoc wireless sensor networks,” 7th IEEE International Symposium on Computers and Communications (ISCC'02), pp. 685–692, 2002.
- [9]: P. S. Chandra, R. V. Krishna, et V. Gupta, « Real Time Communication In Sensor Networks », in QOS seminar report, submitted in IIT Bombay, Mumbai. Date.
- [10]: R. K. Ganti, P. Jayachandran, H. Luo, et T. F. Abdelzaher, « Datalink streaming in wireless sensor networks », in Proceedings of the 4<sup>th</sup> international conference on Embedded networked sensor systems, 2006, p. 209–222.

- [11]: A. Makhoul, «Réseaux de capteurs : localisation, couverture et fusion de données», Thèse pour obtenir le grade de docteur de l'Université de Franche-Comté, novembre 2008.
- [12]: BAHJ (J.), MAKHOUL (A.) And MOSTEFAOUI (A.), « Localization and coverage for high density sensor networks », in: Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), p. 295–300, New York, USA, March 2007.
- [13]: Ameer Ahmed Abbasi, Mohamed Younis, « A survey on Clustering algorithms for wireless sensor networks ». Computer Communication vol. 30, no. 14-15, pp. 284-2826.2007.
- [14]: D. J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," IEEE Trans. Commun., vol. 29, no. 11, 1981, pp. 701-1694.
- [15]: F. Kuhn, T. Moscibroda, and R. Wattenhofer, "Initializing Newly Deployed Ad Hoc and Sensor Networks," Proc. ACM MOBICOM, Sept. 2004, pp. 74-260.
- [16]: A. D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh, "Max-Min DCluster Formation in Wireless Ad Hoc Networks," Proc. IEEE INFOCOM, Mar. 2000, pp. 32-41.
- [17]: H. Chan and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation," Proc. 1st Euro. Wksp. Sensor Networks, Jan. 2004, pp. 71-154.
- [18]: A. Manjeshwar and D. P. Agarwal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks," Proc. Int'l. Parallel and Distrib. Proc. Symp., pp. 195-202.
- [19]: O. Younis, S. Fahmy, HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks, IEEE Transactions on Mobile Computing 3 (4) (2004), pp.366–379.
- [20]: W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, Application specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Networking (2002).

- [21]: S. Bandyopadhyay, E. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, California, April 2003.
- [22]: 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.
- [23]: M. Younis, M. Youssef, K. Arisha, Energy-aware management in cluster-based sensor networks, *Computer Networks* 43 (5) (2003), pp.649–668.
- [24]: Mohamed Younis, Moustafa Youssef, and Khaled Arisha. Energy-aware management for cluster-based sensor networks. *Computer Networks* 43(5), pp. 649–668 (2003).
- [25]: Y Thomas Hou, Yi Shi, Hanif D Sherali, and Scott F Midkiff. On energy provisioning and relay node placement for wireless sensor networks. *IEEE Transactions on Wireless Communications* 4(5), pp 2579–2590 (2005).
- [26]: Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, pages 10–pp. IEEE (2000). vii, 12, 14, 39, 42, 55, 61, 98, 124.
- [27]: W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. “An Application-Specific Protocol Architecture for Wireless Microsensor Networks,” *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, pp. 660-670, Oct. 2002.
- [28]: Shuo Shi, Xinning Liu and Xuemai Gu, “An Energy-Efficiency Optimized LEACH-C for Wireless Sensor Networks,” in : 7th International ICST Conference on Communications and Networking(CHINACOM 2012),China , pp. 487–491, 2012.
- [29]: Heinzelman, W., Chandrakasan, A. and Balakrishnan, H., "An Application Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Transactionson Wireless Communications*, Vol.1, No. 4, pp. 660-670, 2002.
- [30]: Heinzelman, W., “Application-Specific Protocol Architectures for Wireless Networks”. PhD thesis, Massachusetts Institute of Technology, 2000.
- [31]: Akramul Azim and Mohammad Mahfuzul Islam, “ A Dynamic Round-Time Based Fixed Low Energy Adaptive Clustering Hierarchy fo Wireless Sensor Networks,” in : Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications, Kuala Lumpur, Malaysia, pp. 923–926, 15 - 17 December 2009.

- [32] : 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.
- [33]: MATLAB. MATLAB – Le langage du calcul technique - MathWorks France. [en ligne]. <http://www.mathworks.com/products/matlab/>. [Consulté le 03/06/2016].
- [34] : NS3. ns-3. [en ligne]. <https://www.nsnam.org/>. [Consulté le 03/06/2016].
- [35]: IEEE\_802.15.4. IEEE 802.15.4Wikipédia. [en ligne]. [https://fr.wikipedia.org/wiki/IEEE\\_802.15.4](https://fr.wikipedia.org/wiki/IEEE_802.15.4) .[Consulté le 03/06/2016].
- [36] : Tiago Camilo, Jorge Sà Silva, André Rodrigues and Fernando Boavida. “GENSEN: A Topology Generator for Real Wireless Sensor Networks Deployment”. Software Technologies for Embedded and Ubiquitous Systems. Vol 4762. pp 436-445.
- [37]: Min-yi Wang, Jie Ding, Wan-Pei Chen and Wen-Qiang Guan.” SEARCH: A Stochastic Election Approach for Heterogenous Wireless Sensor Networks”. 2015. pp 1-4.
- [38] : Wendi B.Heinzelman, Anantha P.Chandrakasan and Hari Balakrishnan. « An Application-Specific Protocol Architecture for Wireless Microsensor Networks ». IEEE Transactions on wireless communications. Vol. 1. No. 4. October 2002. pp 660-669.