



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة محمد البشير الإبراهيمي برج بو عريريج
Université Mohammed El Bachir El Ibrahimy B.B.A
كلية
Faculté des Mathématiques et d'Informatique
قسم
Département Informatique



Mémoire

En vue de l'obtention du diplôme de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Option : Technologie de l'Information et de la Communication

Intitulé

L'auto organisation en cas des pannes dans les réseaux de capteurs sans fil hétérogène (Protocole ADAPTRP-AO)

Présenté par : Saoud choayb Belmiloud mustapha

Soutenu le 30/ 09/2020 devant le Jury :

Président	Belalta ramela	Univ. M. El-Bachir El-Ibrahimi – BBA
Encadrant	Naidja miloud	Univ. M. El-Bachir El-Ibrahimi – BBA
Examineur	Mohdeb djamila	Univ. M. El-Bachir El-Ibrahimi – BBA

Dédicace

A mes très chers parents, je dédie ce travail

Aucun mot n'est assez fort pour leur exprimer la reconnaissance sincère que je leur porte pour la richesse de leurs enseignements.

A ma très chère grande mère

A mon très cher frère

A tous mes amis mes collègues mes camarades de la promotion

*Et à toute ma famille de **Saoud***

Dédicace

A mes très chers parents, je dédie ce travail

Aucun mot n'est assez fort pour leur exprimer la reconnaissance sincère que je leur porte pour la richesse de leurs enseignements.

A mes très chères deux sœurs

A mon très cher frère

A tous mes amis mes collègues mes camarades de la promotion

*Et à toute ma famille de **Belmiloud***

Remerciements

C'est pour nous un plaisir autant qu'un devoir de remercier toutes les personnes qui ont pu contribuer de près ou de loin à l'élaboration de ce projet, qui nous ont aidé, nous ont soutenu et ont fait en sorte que ce travail ait lieu.

*Nos remerciements les plus sincères vont être adressés à notre encadrant Dr. **NAIDJA MILOUD** pour ses directives, ses conseils et son soutien tout au long du projet. Nous vous remercions pour votre suivi et votre encadrement durant ces mois.*

*Nous remercions **MOHDEB DJAMILA** qui n'a épargné aucun moyen pour nous aider et soutenir, et aussi pour son judicieux encadrement ainsi que sa disponibilité afin de nous aider à élaborer ce travail.*

Résumé

Ces dernières années, nous avons assisté à l'apparition des réseaux sans fil multi-sauts. Avec des capacités distinctes, des caractéristiques différentes et des applications cibles variées, nous pouvons identifier les réseaux de capteurs sans fil RCSF, Ces derniers sont des réseaux auto-organisés et ils sont constitués d'un grand nombre de nœuds capteurs autonomes à faible ressources (capacité de calcul, de mémoire, de puissance d'émission, etc...), C'est dans ce contexte nous avons étudié L'auto organisation en cas des pannes dans les réseaux de capteurs sans fil hétérogène.

Au début, nous nous sommes intéressés à l'auto-organisation dans un contexte hétérogène. Se basant sur l'idée que les ressources au niveau des nœuds actionneurs doivent être exploitées afin de réduire la charge de communication au niveau des nœuds capteurs,

Enfin, présente un protocole de routage ADEPTRP-AO (ADEPTRP-Auto-Organisé): un protocole de routage hybride, qui est une extension du protocole ADEPTRP. La solution est efficace pour l'auto-organisation au niveau des nœuds ordinaires, cluster-heads et station de base. Elle est appropriée pour des applications nécessitant une fiabilité et une tolérance aux pannes car elle garantit que toutes les données soient transmises correctement à la station de base malgré la présence de différents types de défauts, dus à l'énergie, aux pannes matérielles, aux erreurs de communication.

Abstract

In these last years WE have Seen the emergence of multi-hop wireless network Thanks to its unique capabilities, different characteristics and various target applications, we can define RSCF wireless sensor networks

These last years are self-organized networks and they made up of a large number of autonomous sensor nodes with low resources (computing capacity, memory, transmission power etc ...)

In this context we studied the self-organization in case of failures in wireless heterogeneous sensor networks.

Initially: we were interested in self-organization in a heterogeneous context based on the Idea that resources in operator nodes must be exploited to reduce the communication load in the sensor nodes.

Finally: we present a routing ADAPTRP-AO(ADAPTRP-auto-organized a hybrid routing protocol which IS an extension of the ADEPTRP protocol the solution is effective for self-organization at the level of ordinary nodes the heads of the block and the base station it's suitable for application requiring reliability and fault tolerance as it ensures that all IS properly transferred to the base station in spite of the presence of different types of damages because of energy hard ware failure mistakes and communication.

Table des matières

Liste des figures

Liste des diagrammes et des tableaux

Liste des algorithmes

Liste des abréviations

Introduction générale

Chapitre 01 : Les réseaux de capteurs sans fil

Introduction.....	2
1. Capteur sans fil	2
2. Réseau de capteurs sans fil	3
3. Architecture de capteur	3
4. Architecture de réseau de capteurs sans fil	4
4. Domaines d'application.....	5
Conclusion	6

Chapitre 02 : Auto-organisation, Routage, RCSFH

Introduction.....	7
-------------------	---

Partie 01: Auto-organisation

1. Définition d'auto-organisation	7
2. Objectifs de l'auto-organisation	7
3. LEGOS	8
4. Hypothèses.....	8
5. Défis de routage dans les RCSF.....	8

Partie 02: Routage

1. Les principales classes des protocoles de routage dans les RCSF	9
2. Protocoles basés sur la topologie du réseau	10
3. La communication dans les RCSF	11
4. Contraintes de conception d'un protocole de routage	13
5. La contrainte d'énergie pour le routage	14
6. Critères de performance des protocoles de routage	15
7. Classification des protocoles de routage	15
8. Les principaux protocoles de routage	17
9. Le protocole ADAPTRP	20

Partie 03 : Les réseaux hétérogènes RCSFH

7. Les réseaux hétérogènes	23
8. Conditions de fonctionnement.....	23
9. Quelques modèles d'hétérogénéité	23
Conclusion	25

Chapitre 03 : Proposition d'une méthode de diagnostic dans les RCSF

Introduction.....	26
1. Les problèmes de diagnostic dans les réseaux de capteurs sans fil	26
2. Les hypothèses	26
3. Le déroulement de diagnostique	27
3.1 Problème d'énergie	27
3.2 Problème de transmission.....	29
3.4 Problème de capture	34
3.5 Problème de traitement.....	35
Conclusion	37

Chapitre 04 : Mise en oeuvre et simulation et

Introduction.....	38
1. Environnement de simulation	38
1.1 Objectifs de simulation.....	39
1.2 Choix de simulateur NS2	39
1.3 Choix de plateforme OS.....	40
1.4 Choix de langage.....	40
2. Etapes d'implémentation de notre protocole.....	41
2.1 Préparation de l'environnement d'implémentation	41
2.2 Implémentation.....	44
3. Évaluation des performances du protocole ADAPTRP-AO.....	44
Conclusion	51

Conclusion générale

Références

Liste des figures

1.1 Capteur sans fil	2
1.2 Architecture d'un capteur	3
1.3 Architecture d'un réseau de capteurs sans fil	5
1.4 Domaines d'application des réseaux de capteurs sans fil	5
2.1 Classification des protocoles de routage pour les RCSF	10
2.2 Communication multi-sauts dans un réseau de capteurs sans fil	12
2.3 Architecture de communication du protocole HEEP	13
2.4 Algorithme d'ADAPTRP	22
2.5 Exemple d'un cluster dans un RCSF et actionneur	24
2.6 Hétérogénéité de l'unité de capture	24
4.1 Positions des nœuds et de la station de base	45
4.2 Quantité de données reçues en fonction de la durée de la simulation	46
4.3 Quantité de données reçues par la SB en fonction de l'énergie consommée	46
4.4 Nombre de nœuds vivants en fonction de la durée de la simulation	47
4.5 Nombre de nœuds vivants en fonction de la quantité de données reçues par la SB.	48
4.6 Quantité d'énergie consommée en fonction de la durée de la simulation	48
4.7 Quantité de données reçues (panne de la SB).	49
4.8 Quantité de données reçues (panne de la SB, des CHs et nœuds ordinaires).	50
4.9 Nœuds défaillants détectés dans ADEPTRP et ADEPTRP-AO.	51

Liste des diagrammes et des tableaux

3.1 Diagramme de diagnostic d'énergie pour un nœud dans les RCSFs.	28
3.2 Diagramme de diagnostic de réception pour un nœud dans les RCSFs.	33
3.3 Diagramme de diagnostic de captage pour un nœud dans les RCSFs.	35
3.4 Diagramme d'un diagnostic de traitement pour un nœud dans les RCSFs.	36
4.1 Tableau des paramètres de simulation.	45

Liste des algorithmes

3.1 Algorithme de diagnostic d'énergie pour un nœud dans les RCSFs.....	29
3.2 Algorithme de diagnostic de réception pour un nœud dans les RCSFs.....	32
3.3 Algorithme d'un diagnostic de captage pour un nœud dans les RCSFs.....	34
3.4 Algorithme de diagnostic de traitement pour un nœud dans les RCSFs.....	37

Liste des abréviations

LEACH : Low Energy Adaptive Clustering Hierarchy

LEACH-C : LEACH-Centralisé

PEGASIS : Power-Efficient Gathering in Sensor Information Systems

H-PEGASIS : Hierarchical PEGASIS

HEED : Hybrid Energy-Efficient Distributed clustering

HEEP: Hybrid Energy Efficiency Protocol

APTEEN : AdaPtive Threshold sensitive Energy Efficient sensor Network protocol

TEEN : Threshold sensitive Energy Efficient sensor Network protocol

ADAPTRP: ADAPTative Routing Protocols

CH : Cluster Head

BS : Base Station

SB : Station de Base

TDMA : Time Division Multiplexed Access

SBP : Station de Base Primaire

SBS : Station de Base Secondaire

LAN : Local Area Network

GPS : Global Positioning System

RAM : Random Access Memory

ROM : Read Only Memory

RCSF : Réseau de Capteurs Sans Fil

WSN : Wireless Sensors Networks

WSANs : Wireless Sensors and Actuators Networks

ADC : Analog to Digital Converters

LEGOS: Low Energy self-organization scheme for wireless sensor networks

RCSFH : Réseau de Capteurs Sans Fil Hétérogènes

Introduction générale

Introduction générale

Les réseaux de capteurs sans fil (RCSF) sont généralement composés d'un grand nombre de nœuds capteurs qui sont déployés aléatoirement dans une zone d'intérêt pour surveiller un tel phénomène. Ces capteurs collaborent entre eux et remontent l'information de nœud en nœud selon le mode multi-sauts et via une connexion sans fil jusqu'à l'aboutissement à la station de base. Or, ce schéma de communication est coûteux en termes d'énergie puisque tout nœud joue le rôle d'un nœud relais puisqu'il renvoie l'information reçue pour la première à ses voisins. Ce processus de communication se répète jusqu'à l'arrivée de l'information à la station de base. De ce fait, pour se remédier à cet inconvénient, plusieurs protocoles de routage ont été proposés dans la littérature, d'une part pour communiquer l'information rapidement à la station de base, et d'autre part pour communiquer l'information à la station de base à un moindre coût.

Le déploiement du réseau est soit déterministe, soit auto-organisé. De nombreux schémas de routage et d'auto-organisé ont été proposés pour les réseaux de capteurs. La topologie hiérarchique dans les RCSF répond aux exigences de conservation d'énergie. Dans de tels réseaux, un algorithme spécifique est appliqué au niveau de tous les nœuds capteurs. Sur cette base, nous avons proposé une solution pour l'auto-organisation en prenant en considération l'hétérogénéité du réseau. La solution présente un protocole de routage ADAPTRP-AO (ADAPTRP-Auto-Organisé) : un protocole de routage hybride, qui est une extension du protocole ADAPTRP. La solution est efficace pour l'auto-organisation au niveau des nœuds ordinaires, cluster-heads et station de base. Elle est appropriée pour des applications nécessitant une fiabilité et une tolérance aux pannes car elle garantit que toutes les données soient transmises correctement à la station de base malgré la présence de différents types de défauts, dus à l'énergie, aux pannes matérielles, aux erreurs de communication et aux attaques malveillantes.

La suite de ce mémoire est organisée en quatre chapitres. Le premier chapitre consistera en la présentation des réseaux de capteurs sans fil. Le deuxième portera sur les concepts principaux de l'auto-organisation, de routage et des protocoles de routage, et des réseaux de capteurs sans fil hétérogènes. Le troisième, fera l'objet de l'étude conceptuelle et technique du protocole ADAPTRP-AO proposé. Dans le quatrième et dernier chapitre, nous présenterons les résultats obtenus par notre protocole après la mise en œuvre et la simulation. Enfin, nous dresserons une conclusion générale du projet.

Chapitre 01 : les réseaux de capteurs sans fil

Introduction

Les progrès réalisés ces dernières décennies dans les domaines de la microélectronique, de la micromécanique, et des technologies de communication sans fil, ont permis de produire à un coût raisonnable des composants de quelques millimètres cubes de volume. De ce fait, un nouveau domaine de recherche s'est créé pour offrir des solutions économiquement intéressantes et facilement déployables à la surveillance à distance et au traitement des données dans les environnements complexes et distribués : les réseaux de capteurs sans fil.

Les réseaux informatiques sont des ensembles d'appareils généralement ordinateurs, reliés entre eux avec ou sans fil, dans le but de permettre à ses utilisateurs de transférer des informations électroniques, Ils permettent de communiquer, partager des ressources matérielles ou logicielles.

Les réseaux mobiles sont des réseaux dans lesquels au moins deux terminaux peuvent communiquer sans liaison filaire. Grâce aux réseaux sans-fils, un utilisateur a la possibilité de rester connecté tout en se déplaçant dans un périmètre géographique plus ou moins étendu. On distingue plusieurs catégories de réseaux sans-fils.

Cette partie du premier chapitre consiste à décrire quelques généralités sur les capteurs, les réseaux et les réseaux de capteurs sans fils.

1. Capteur sans fil

Les capteurs (figure 1.1) sont des dispositifs de taille extrêmement réduite avec des ressources très limitées, autonomes, capable de traiter des informations et de les transmettre, via les ondes radio, à une autre entité (capteurs, unité de traitements...) sur une distance limitée à quelques mètres.



Figure 1.1 Capteur sans fil

2. Réseau de capteurs sans fil

Un Réseau de Capteurs Sans Fil (RCSF) ou Wireless Sensor Networks (WSN) est un nouveau paradigme informatique fondé sur les efforts collaboratifs d'une multitude de nœuds capteurs spatialement répartis qui travaillent en collaboration via des liaisons sans fil pour envoyer les données collectées vers la station de base sur les zones surveillées.

3. Architecture de capteur sans fil

Généralement, un capteur est composé de quatre unités de base : une unité de capture (*Sensing unit*), une unité de traitement (*Processing unit*), une unité d'émission/réception (*Transceiver unit*) et une unité d'énergie (*Power unit*). Il peut contenir également, suivant son domaine d'application, des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire). On peut même trouver des micro-capteurs, un peu plus volumineux, dotés d'un système mobilisateur chargé de déplacer le micro-capteur en cas de nécessité. Voici la schématisation d'une architecture de capteur sans fil (Figure 1.2).

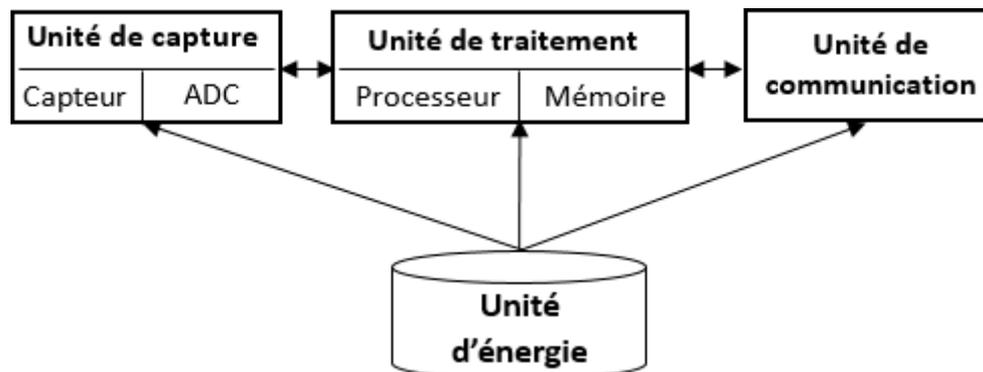


Figure 1.2 : Architecture d'un capteur

3.1. L'unité de capture (unité d'acquisition des données)

C'est l'unité qui est chargée de capter des grandeurs physiques (chaleur, humidité, vibrations, rayonnement, etc..) et de les transformer en grandeurs numériques (un signal électrique). Cette unité peut incorporer de un jusqu'à plusieurs capteurs plus un convertisseur analogique-numérique appelé ADC.

3.2. L'unité de traitement des données

Mote, processeur, RAM et Flash : On appelle généralement *Mote* la carte physique utilisant le système d'exploitation pour fonctionner. Celle-ci a pour cœur le bloc constitué du

processeur et des mémoires RAM et Flash. Cet ensemble est à la base du calcul binaire et du stockage, temporaire pour les données et définitif pour le système d'exploitation. Cette unité est chargée d'exécuter les protocoles de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits [1].

3.3. L'unité de transmission de données

L'unité de communication est composée d'un émetteur/récepteur qui permet aux nœuds capteurs du réseau de pouvoir communiquer entre eux par l'intermédiaire de liaisons radio. Ainsi, elle effectue toutes les émissions et réceptions des données sur le support sans fil. Elle peut être de type optique ou radio fréquence.

3.4. La source d'énergie

Un capteur est muni d'une source d'énergie qui est une batterie, elle alimente les unités que nous avons citées et elle n'est généralement ni rechargeable ni remplaçable. Donc, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs.

4. Architecture du réseau de capteurs sans fil

Un réseau de capteurs sans fil (RCSF), consiste en un ensemble de nœuds capteurs, placés de manière plus ou moins aléatoire, dans une zone géographique appelée zone de captage ou zone d'intérêt, afin de surveiller un phénomène physique et de récolter leurs données d'une manière autonome. Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées vers un nœud collecteur appelé nœud puits ou station de base. Cette dernière transmet ensuite ces données par internet ou par satellite à l'ordinateur central (Gestionnaire de tâches) pour analyser ces données et prendre des décisions.

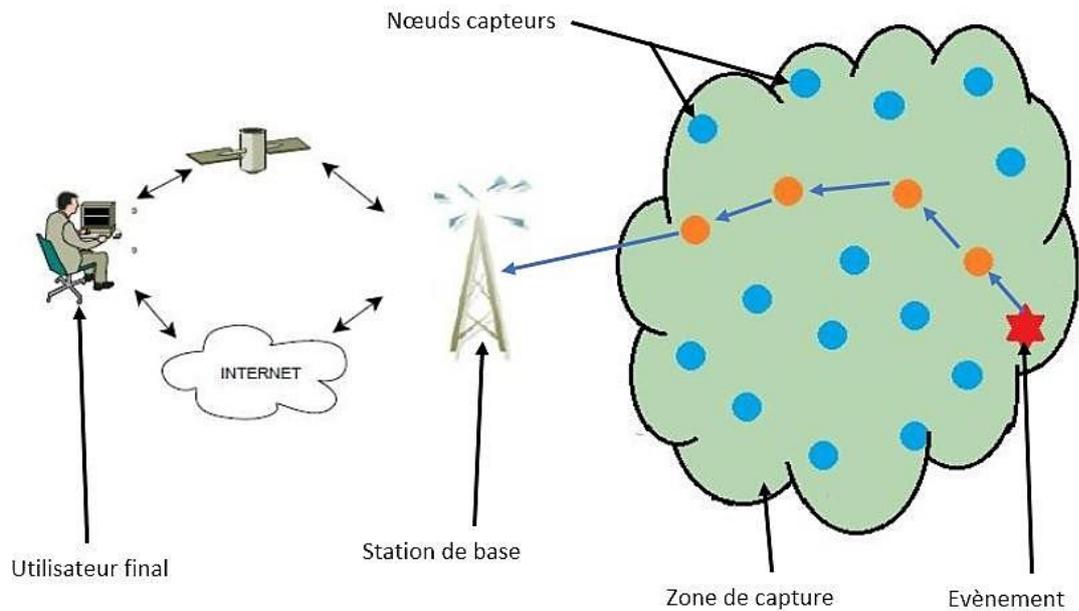


Figure 1.3 Architecture d'un réseau de capteurs sans fil

5. Domaines d'application

Les RCSF sont destinés à surveiller et observer des phénomènes physiques comme : la température, l'humidité, la pression, le taux de bruit, le mouvement des véhicules et la vitesse. Nous pouvons citer les domaines suivants : militaire, environnemental, domestique, santé, sécurité, écologie, traçabilités, etc. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous (figure 1.4) :



Figure 1.4 Domaines d'application des réseaux de capteurs sans fil

- ✓ Dans le domaine militaire, on trouve que Les RCSF sont le résultat de la recherche militaire. Ils sont utilisés dans la surveillance des champs de bataille pour connaître surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes.
- ✓ Dans le domaine médical, la surveillance permanente des patients est une possibilité de collecter des informations physiologiques de meilleure qualité facilitant ainsi le diagnostic de maladies grâce à des micro-capteurs qui pourront être ingérés ou implantés sous la peau.
- ✓ Dans le domaine environnemental, les réseaux de capteurs sans fils sont très utiles dans la protection de l'environnement, Ils peuvent être utilisés pour la surveillance de la terre, l'exploration planétaire, la détection d'incidents chimiques ou biologique, qualité de l'air, et la pollution...etc.

Conclusion

Nous avons essayé à travers ce chapitre de mettre le point sur l'architecture des RCSF, ainsi que leurs caractéristiques et les domaines d'applications. Cette mise au point nous a permis de déduire que les protocoles de routage et la dissipation d'énergie jouent un rôle déterminant et crucial dans la conception des RCSF. Cela nous a mené à faire une étude des principaux protocoles de routage dans le chapitre qui suit.

Chapitre 02 : auto-organisation, routage, RCSFH

Introduction

Le présent chapitre est constitué de trois parties. Dans cette première partie, nous commençons par introduire et définir la notion d'auto-organisation. Ensuite, nous décrivons quelques techniques d'auto-organisation en se basant sur la création d'une topologie logique au-dessus d'une topologie physique. Nous décrivons par la suite les défis de routage dans les RCSF. Dans la deuxième partie, nous décrivons les concepts essentiels de routage dans les RCSF avec ses classes principales, ses contraintes de conception, et ses critères de performances. Dans la troisième et dernière partie nous introduirons finalement les réseaux hétérogènes WSAF, leurs architectures et applications.

I. Partie 01 : Auto organisation

1. Définition d'auto-organisation

Un système constitué de plusieurs entités, est dit organisé s'il possède une structure et un ensemble de fonctions. La structure vise à créer une relation entre les entités en les disposant d'une certaine manière particulière et à faciliter la communication entre ces entités. L'ensemble de fonctions a pour rôle de maintenir la structure et l'utilisation de celle-ci pour répondre à des besoins bien déterminés [2].

2. Objectifs de l'auto-organisation

Un réseau auto-organisé crée une topologie logique au-dessus de la topologie physique. La topologie logique est construite non pas seulement pour « cacher » la dynamique locale au niveau de chaque entité appartenant à la topologie (la dynamique couvre aussi bien l'apparition/disparition des liens et des nœuds) mais aussi pour faciliter le déploiement des protocoles de communication au-dessus de cette topologie logique. Ainsi la topologie s'adapte aux changements de la topologie et se reconstruit dynamiquement et localement (dans le cas idéal). De plus, cette topologie logique doit être capable de supporter le passage à l'échelle qui est une caractéristique partagée entre les RCSF et les RSCFA. Les principaux objectifs de la construction d'une topologie logique sont :

- Minimiser la consommation énergétique au niveau du réseau ;
- Améliorer les performances du réseau ;
- Partager et gérer les ressources disponibles dans le réseau.

3. LEGOS (Low Energy self-organization scheme for wireless sensor networks)

LEGOS est un protocole d'auto-organisation des réseaux de capteurs proposé par LU, J *et al*, [3]. Il avait pour objectif de fournir un protocole d'auto-organisation simple, efficace en énergie, adapté aux topologies larges échelles qui permet de structurer le réseau et qui permet de s'adapter aux changements dans le voisinage. La structure du réseau devait répondre aux exigences suivantes :

- Seules des interactions locales doivent être utilisées pour faire émerger un comportement global et réagir aux changements dans le voisinage ;
- Eviter l'émission de paquets périodiques « hello » afin d'économiser de l'énergie,
- La complexité et la puissance de calcul requise doivent être efficaces pour être adaptés aux ressources limitées des capteurs.

4. Hypothèses

L'auteur de LEGOS ciblait les réseaux de capteurs sans fil avec les hypothèses suivantes :

1. Capteurs identiques : tous les capteurs ont les mêmes capacités en termes de mémoire, puissance de calcul, communication et d'énergie. Seul le puits peut avoir des capacités différentes mais il n'est pas pris en compte afin de garantir plus de flexibilité ;
2. Energie comme ressource critique : l'optimisation de l'énergie doit être présenté systématiquement pour améliorer la durée de vie du réseau ;
3. Aucune information de localisation géographique n'est connue de la part des nœuds et aucune information globale n'est requise.
4. Aucune hypothèse sur la distribution des nœuds sur la zone de surveillance : les nœuds sont déployés aléatoirement.
5. Aucune hypothèse en termes de synchronisation : les nœuds exécutent LEGOS indépendamment sans être nécessairement synchronisés.
6. Pas d'infrastructure.

II. Partie 02 : Routage

Le routage dans les réseaux de capteurs sans fil est la procédure d'acheminement des données d'un nœud source collecteur vers une destination (généralement la station de base) à travers un réseau de connexion. Chaque nœud est susceptible d'être à la disposition des autres nœuds capteurs pour participer à la transmission et retransmission des informations émises sur

le réseau par un ou un ensemble de nœuds n'ayant pas la possibilité d'atteindre directement la destination. Cette dernière peut être un nœud ordinaire, un chef de groupe « cluster-head » dans le cas de réseaux hiérarchiques ou la station de base pour les réseaux à plat.

1. Les principales classes des protocoles de routage dans les RCSF

Les protocoles de routage proposés dans les RCSF peuvent être classés suivant des critères différents. La figure 2.1 résume cette classification qui se base sur quatre critères : la topologie du réseau, mode de fonctionnement du protocole, les paradigmes de communication et selon le mode d'établissement des chemins. Dans notre proposition on s'intéresse beaucoup plus par la classification basée sur la topologie du réseau.

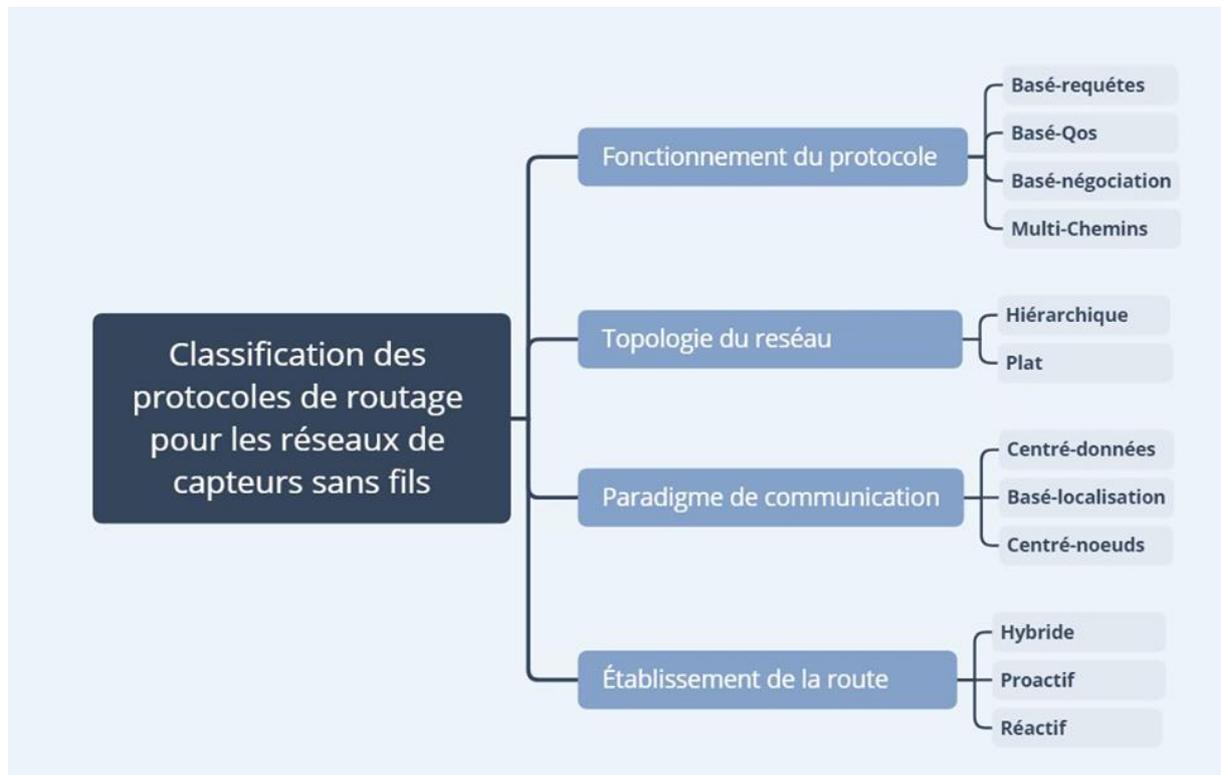


Figure 2.1 Classification des protocoles de routage pour les RCSF

2. Protocoles basés sur la topologie du réseau

En général, le routage dans les réseaux de capteurs peut être classé, selon la structure du réseau, en deux classes [4], routage à plat et routage hiérarchique.

2.1. Routage à plat

Appelé également routage centré données (*data centric*) [5] où tous les nœuds ont les

mêmes tâches à accomplir. C'est la première approche utilisée dans l'acheminement des données dans les RCSF. Elle se base sur la collaboration de tous les nœuds du réseau. Les propriétés des données sont spécifiées par un système de dénomination par attribut [6] [7] en raison de la difficulté d'affecter un identificateur global à chaque nœud vu leur nombre important. Parmi leurs avantages, la simplicité d'où la possibilité d'établir des communications sans surcoût où chaque nœud n'aura besoin que des informations de ses voisins directs. L'inconvénient est l'épuisement des ressources en énergie des nœuds proches de la station de base car tout le trafic vers cette dernière passe obligatoirement par eux.

2.2. Routage hiérarchique

Cette approche est basée sur la formation de clusters (zones communes). Le principe est de router les données récoltées par chaque nœud du cluster à son chef de zone (Cluster Head), qui et après des traitements sur leurs parties communes, les transmettra à la prochaine destination (si le CH ne pourra pas atteindre directement la station de base, les informations seront routées vers le prochain chef de zone). L'avantage est la réduction des coûts en communication et en énergie en minimisant le nombre de messages circulant sur le réseau, étant donné que les CHs appliquent des fonctions d'agrégat sur les données du cluster ce qui permet de les combiner. L'inconvénient de cette approche est lorsque la taille du réseau augmente, le processus d'élection du Cluster Head devient critique et gourmand en ressources.

3. La communication dans les RCSF

Une communication est définie par l'échange d'informations entre les capteurs du réseau. Elle peut être établie suite à un événement déclenché par un nœud capteur ou à une requête diffusée sur le réseau par la station de base, l'utilisateur final de l'application ou un nœud chef de groupe.

Les réseaux de capteurs sans fils sont souvent utilisés dans des environnements hostiles sans aucune infrastructure de base, ne possédant aucune information sur l'emplacement des nœuds capteurs et par conséquent sur la topologie du réseau. Dans des conditions similaires, les nœuds capteurs doivent s'auto-organiser et construire une infrastructure de communication autonome qui leur permet d'interagir entre eux et d'établir un cheminement de n'importe quel nœud capteur vers n'importe quelle destination sur le réseau.

Un nœud capteur joue plusieurs rôles à la fois : collecteur, intermédiaire ou collecteur et intermédiaire en même temps. Les nœuds intermédiaires participent à la mise en place

d'architectures de communications multi-sauts. Les nœuds capteurs peuvent atteindre en un seul saut. Ce type de communication ne nécessite aucune collaboration ni architecture de communication pour le routage de données et d'informations. Par contre le type de communication multi-sauts, chaque nœud capteur envoie ses informations au nœud voisin, ce dernier retransmet les informations à son nœud voisin jusqu'à ce que les informations arrivent à la station de base.

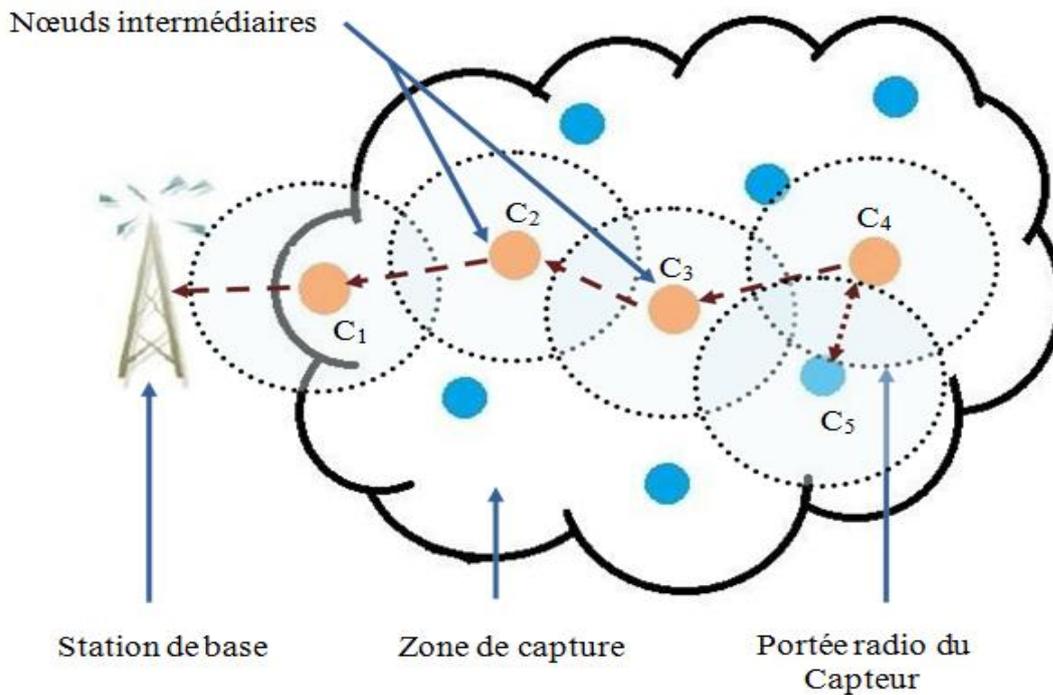


Figure 2.2 Communication multi-sauts dans un réseau de capteurs sans fil

La figure 2.2 montre un exemple d'un réseau constitué de plusieurs nœuds capteurs. Il s'agit d'une communication multi-sauts entre le nœud collecteur C_4 et la station de base et d'une communication en un seul saut entre C_1 et la station de base. A partir de cette organisation, on distingue quatre types de communications dans les réseaux de capteurs sans fil :

- i. Communication directe entre un nœud collecteur C_4 et un autre nœud collecteur C_5 ce type de communication est généralement utilisé dans des opérations locales telles que la procédure de création de routes ou pendant le processus de clusterisation [8].
- ii. Communication entre un nœud collecteur C_4 et un nœud intermédiaire C_3 : Puisque le nœud collecteur n'est pas en mesure d'atteindre directement la station de base, ses données collectées sont transmises à un nœud intermédiaire se trouvant dans sa portée radio.

- iii. Communication entre un nœud intermédiaire C3 et un autre nœud intermédiaire C2 : souvent en unicast [9], ce type de communication a pour rôle d'installer un réseau de connexion entre le nœud collecteur et la station de base.
- iv. Communication entre un nœud intermédiaire C1 et la station de base : c'est la dernière étape avant la réception finale des données par la station de base. Cette communication peut concerner également une transmission directe entre un nœud collecteur et la station de base.

4. Contraintes de conception d'un protocole de routage

Nous allons présenter dans ce qui suit quelques contraintes liées aux caractéristiques des protocoles de communication et de routage de données dans les réseaux de capteurs sans fil.

4.1. Capacités réduites des capteurs : les capacités réduites des capteurs en calcul, en mémoire et en énergie empêchent le développement de mécanismes complexes, volumineux et complets qui s'exécutent sur des processeurs puissants et qui nécessitent des tailles mémoire importantes. De même, l'une des recommandations majeures à prendre en compte est de répartir équitablement l'énergie consommée lors de la découverte et la maintenance des routes. De plus, l'agrégation de données, la clusterisation, contrôle de la puissance du signal sont, entre autres, des techniques de conservation d'énergie appliquées par un protocole de routage lors de la transmission de données entre les différentes entités du réseau, utilisant une architecture de communication adéquate.

4.2. La taille du réseau : le nombre de nœuds d'un réseau de capteurs sans fil est souvent important et sa topologie change fréquemment avec la défaillance ou l'ajout de certains nœuds capteurs. Ce changement de topologie provoque une maintenance et une mise à jour régulière des schémas de gestion de routes ; deux opérations gourmandes en énergie. Ainsi, la réorganisation du réseau et la mise à jour des tables de routage, ou parfois le changement de l'approche de routage utilisée selon les nouvelles données, implique la participation d'un grand nombre de nœuds capteurs, ce qui est difficile à gérer par les protocoles de routage.

4.3. Déploiement des nœuds : les nœuds d'un RCSF peuvent être déployés d'une manière sélective ou aléatoire. Dans un environnement de déploiement sélectif où les nœuds capteurs sont placés à la main, les protocoles de routage peuvent utiliser

directement les informations pré-chargées sur les capteurs avant leur déploiement sur la zone de captage. On parle alors de routage prédéterminé. Ce genre de déploiement ne pose pas trop de problèmes aux protocoles de routage car les informations sur l'emplacement des capteurs sont statiques et les mises à jour des tables de routage s'effectuent d'une manière déterministe. Cependant, les informations sur les positions des capteurs dispersés aléatoirement sur une zone à surveiller sont difficiles à connaître ou à prévoir et l'emplacement des capteurs est quasi inconnu. Dans ce cas, on procède alors à la définition de nouveaux mécanismes chargés de la découverte des routes (algorithmes de Dijkstra, algorithme de Nilson, algorithmes génétiques, technique du spanning tree... etc.).

- 4.4. La dynamique du réseau :** dans la plupart des applications des réseaux de capteurs sans fil, l'emplacement des nœuds sur une zone de captage est souvent statique et les positions des capteurs sont fixes et invariables, ce qui maintient régulièrement à jour l'état des liens et les informations de routage stockées sur les nœuds capteurs. Néanmoins, des facteurs externes comme les vents, les orages, et autres conditions climatiques ainsi que le facteur humain (d'une manière accidentelle ou ciblée) et animal peuvent changer le positionnement géographique des nœuds capteurs et engendrer l'apparition ou la disparition de liens entre les nœuds ce qui va altérer au processus de routage. Le problème de la gestion de la mobilité au sein d'un RcSF est l'objet de plusieurs projets de recherche proposant des solutions nouvelles [10] aux problèmes identifiés [11] mais soulevant de nouveaux défis notamment la gestion des communications entre nœuds mobiles [12] et l'adaptation des schémas anciens aux nouvelles informations volatiles et non permanentes.
- 4.5. Tolérance aux pannes :** un nœud participant à l'acheminement de données sur un réseau de capteurs est susceptible de tomber en panne suite à l'épuisement de ses réserves d'énergie, à une défection matérielle, à une compromission...etc. ce qui ne doit pas altérer le fonctionnement du protocole de routage. Un mécanisme de gestion et de prise en charge de pannes doit être prévu et mis en place.
- 4.6. Qualité de service :** un protocole de routage doit veiller au respect des délais d'acheminement et à préserver l'intégrité des données émises car un message, une donnée ou une information arrivant après un certain délai ne serait peut-être plus utilisable surtout dans des applications temps réel.

5. La contrainte d'énergie pour le routage

La consommation d'énergie est la métrique la plus importante dans l'évaluation de la durée de vie des capteurs et des performances d'un RCSF. La source d'énergie des capteurs, conçus pour fonctionner durant des mois et même des années et alimentés par des batteries de capacité limitée, doit être utilisée de façon optimale pour toute tâche effectuée, notamment la communication et l'échange d'informations. Plusieurs métriques et techniques sont utilisées par les protocoles de routage afin d'optimiser les sources de consommation d'énergie.

6. Critères de performance des protocoles de routage

- 6.1. Consommation d'énergie :** le nœud capteur doit utiliser son énergie d'une façon optimale pour ses activités de détection, de traitement et de communication. Un protocole de routage doit gérer les périodes d'activité et d'inactivité des capteurs en incluant des modes « en marche » et « en veille » et avoir la notion de temps pour se mettre en veille et se réveiller. Le mode « en veille » permet au nœud capteur d'éteindre son interface radio et d'empêcher une perte d'énergie due aux écoutes actives et en permanence inutiles de son environnement.
- 6.2. Temps de traitement :** c'est le temps pris par un nœud capteur pour effectuer des opérations de calcul sur les données récoltées ou reçues. Ce temps doit être raisonnable pour ne pas causer des retards de transmission d'informations pour des applications critiques et temps réel.
- 6.3. La mobilité des nœuds capteurs :** la position des capteurs sur la zone de captage n'est pas toujours fixe. Un nœud capteur peut devenir mobile et changer sa position selon les besoins de l'utilisateur. Des traitements spécifiques pour la maintenance des liens et la mise à jour des informations de routage sont à prévoir lors de la conception d'un protocole de routage.
- 6.4. Modes de transmission :** choisir un mode de communication adéquat à la structure et à la topologie du réseau de déploiement.
- 6.5. Sécurité des échanges :** consiste l'envoi périodique de paquets de contrôle afin d'éviter des collisions et des pertes de données. Un paquet de contrôle peut contenir le nombre de bits émis, l'adresse ou l'identificateur de destination et des informations sur le routage. Ces paquets sont nécessaires afin d'assurer la disponibilité et

l'intégrité des données transmises.

7. Classification des protocoles de routage

Les protocoles de routage sont conçus différemment pour répondre aux objectifs d'un réseau de capteurs sans fil. Plusieurs applications des capteurs exigent un routage efficace, sécurisé et économe en énergie assurant une bonne qualité de service et des temps de traitement et de transmission convenables. Le choix de conception d'un protocole de routage pour les RCSF est assez vaste et nous pouvons les classifier par différentes façons. Certaines applications de réseaux de capteurs sont mises en place pour récolter des données périodiques sur leur environnement tandis que d'autres sont chargées de répondre à des événements importants produits à l'intérieur ou à proximité de la zone à surveiller, et d'autres encore sont conçues pour capter des informations précises sur des zones ciblées.

Les techniques de transmission de données sur un réseau de capteurs sans fil diffèrent selon plusieurs critères. Le tableau ci-dessous présente une classification détaillée des protocoles de routage dans les RCSF.

Critère	Classification	Définition
Topologie du réseau	Hiérarchique	Le réseau est organisé en clusters. Le routage effectuée sur plusieurs niveaux (intra-cluster et inter-cluster).
	Plat	Tous les nœuds ont le même rôle et collaborent entre eux pour accomplir le routage.
Les fonctions	Multi-chemins	Utiliser des chemins multiples afin d'augmenter les performances du réseau en maintenant des chemins alternatifs.
	Négociation	Eliminer les transmissions redondantes et établir des communications selon les ressources du réseau.
	Qualité de service	Le réseau doit satisfaire la qualité des données avec une consommation raisonnable d'énergie.

du protocole	Requête	Les données sont envoyées à un certain intervalle ou quand le nœud capture certains événements. Les nœuds répondent aux requêtes envoyées par la destination.
Paradigme de communication	Centré - nœuds	Les communications se basent sur l'identification des nœuds participants.
	Centré - nœuds	Les communications se basent sur les données à transmettre
	Basé localisation	Les communications se basent sur la position des nœuds.
Etablissement de la route	Proactif	Les chemins sont établis à priori.
	Réactive	Les chemins sont établis à la demande selon les besoins.
	Hybride	Combine les deux techniques proactive et réactive.

8. Les principaux protocoles de routage

8.1. Leach (*Low-Energy Adaptive Clustering. Hierarchy*) : introduit par Heinzelman *et al.* est l'un des protocoles les plus populaires pour les réseaux de capteurs sans fil [13][14]. Le principe de ce protocole de routage est de former des groupes communs de calcul et de traitements en se basant sur la puissance du signal et le niveau d'énergie des nœuds capteurs. Chaque cluster est dirigé par un chef de cluster, jouant le rôle d'agrégateur et de routeur, en effectuant des traitements sur les données reçues de son cluster et leur expédition vers la prochaine destination. Ce rôle de chef de cluster est échangé entre les nœuds d'un cluster afin de répartir équitablement la consommation d'énergie entre eux.

Le protocole LEACH se déroule en rounds. Chaque round se compose de deux phases : Phase d'initialisation et phase de communication. La première phase consiste à définir les clusters et élire les Clusters Head (CH), la phase de communication est responsable de la transmission des données captées.

8.2. Leach-C : les créateurs de LEACH ont mis en place une version centralisée baptisé

LEACH-C [15] qui est en fait une extension de l'algorithme de LEACH. En effet, ce dernier s'appuie toujours sur le clustering, d'ailleurs il en garde les principales caractéristiques à quelques différences près comme l'ajout de la méthode d'optimisation de recuit "simulé" qui sert à calculer la structure des clusters au niveau de la station de base. De plus, dans LEACH-C, la SB définit à chaque itération de l'algorithme le rôle de chaque nœud capteur, il peut être Cluster-Head ou bien un nœud ordinaire.

8.3. PEGASIS: (*Power-Efficient Gathering in Sensor Information Systems*)

proposé par Lindsey et al. en l'année 2002 [16], ce protocole est l'une des améliorations de LEACH, où le réseau est vu comme un arbre, les nœuds forment des chaînes plutôt que des clusters. Un nœud transmet et reçoit uniquement les données de son voisin. Chaque nœud est considéré comme un nœud agrégateur, ses données reçues sont traitées et envoyées au prochain nœud de la chaîne. Ainsi, toutes les données capturées sont fusionnées et transmises par un seul nœud désigné pour communiquer avec la station de base. Les nœuds qui transmettent à la station de base sont choisis, pour un intervalle de temps bien défini, selon la politique de remplacement Round Robin dans le but de répartir équitablement l'énergie consommée durant un round de transmission. Le principe est d'organiser le réseau sous forme d'arbre hiérarchique où les nœuds collecteurs sont considérés comme des feuilles et la station de base comme la racine. Les données captées transitent d'une feuille à la racine par des nœuds intermédiaires formant une chaîne. A la réception d'un paquet de données, le nœud intermédiaire procède à son traitement avant son expédition vers son voisin direct de la chaîne. Le dernier nœud de la chaîne (appelé leader) transmet les données fusionnées à la station de base.

8.4. Hierarchical PEGASIS (H-PEGASIS) [17] : est une révision de PEGASIS.

Ainsi, il se fixe pour objectif la diminution des délais de transmissions des paquets vers la station de base, et de plus, il suggère une solution au problème de la collecte de données. La transmission simultanée de messages est adoptée dans le but de réduire les délais dans PEGASIS.

8.5. HEED: (*Hybrid Energy Efficient Distributed clustering*): proposé par O.

Younis *et al.* [18]. S'établit sur le schéma basic de LEACH. L'un de ses aspects le plus important, c'est la méthode avec laquelle sont sélectionnés les CHs. Dans le but d'assurer un équilibrage de puissance énergétique, la construction des clusters se fait

selon une combinaison de deux paramètres. L'un des paramètres dépend de l'énergie résiduelle des nœuds, le second dépend du coût des communications intra-cluster [19]. Les CHs sont élus périodiquement en se basant sur l'énergie résiduelle et le coût de la communication intra-cluster des nœuds candidats [20]. L'envoi des données depuis un Cluster-Head vers la SB se fait en sauts multiples.

8.6. HEEP (Hybrid Energy Efficiency Protocol) : proposé par Boubiche *et al.* en l'an 2011 dans [21], l'organisation des nœuds appartenant à la même grappe (cluster) sous forme d'une chaîne permettent d'améliorer et de réguler la dissipation d'énergie permettant de réduire la charge sur le cluster-head. En effet, les nœuds communiquent uniquement avec leurs proches voisins et non pas directement avec leur cluster-head, ce qu'économise d'avantage l'énergie et offre une meilleure utilisation de la bande passante. L'agrégation des données au niveau de chaque nœud dans la chaîne réduit la quantité de données échangées entre les nœuds et leur cluster-head, ce qui a pour effet de préserver les réserves énergétiques de ces derniers.

La figure 2.3, montre comment les nœuds seront organisés à l'intérieur des clusters. Le nœud $N0$ transmet ses données à son proche voisin $N1$ qui à son tour agrège les données reçues avec les siennes et les transmet à son voisin jusqu'à atteindre le cluster-head qui les transmet directement à la station de base. Donc, dans cette nouvelle organisation (grappe à chaînes), tous les nœuds du cluster vont transmettre leurs données collectées à leurs cluster-heads respectifs en se reliant à travers la chaîne, tandis que chaque CH doit recevoir les données collectées par les nœuds de même cluster.

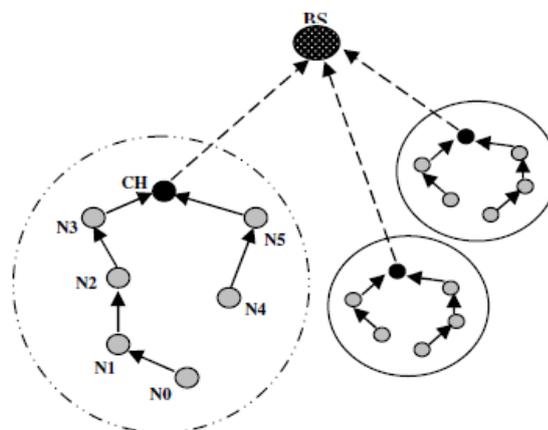


Figure 2.3 Architecture de communication du protocole HEEP [21].

8.7. TEEN (Threshold sensitive Energy Efficient sensor Network

protocol) : introduit dans [22] par A. Manjeshwar *et al.* , le protocole TEEN est conçu pour les applications se basant sur le mode de communication événementiel (les applications dédiées pour détecter les changements soudains). L'architecture du réseau de capteurs est basée sur un groupement hiérarchique multi-niveaux où les nœuds les plus proches forment des clusters. TEEN est une hybridation du clustering hiérarchique et parmi les protocoles centrés-données. Il utilise la stratégie de LEACH pour la création des clusters. Après l'établissement des clusters, le CH diffuse aux membres de son groupe le seuil maximal et minimal. Les nœuds capteurs surveillent leurs environnements continuellement. Une fois que la valeur des données dépasse le seuil maximal, ou diminue le seuil minimal le nœud transmet les données récoltées.

8.8. APTEEN (AdaPtive Threshold sensitive Energy Efficient sensor Network

protocol) : proposé dans [23] par A. Manjeshwar *et al.*, est une réforme de TEEN dédié à la fois aux applications se basant sur le mode de communication événementiel et à celles se basant sur le mode de communication périodique. La formation des clusters est faite par la station de base puis des valeurs des attributs sont diffusées par les cluster-heads vers tous les nœuds du réseau. Une agrégation de données est aussi réalisée par les Cluster-Head dans le but de minimiser l'énergie.

9. Le protocole ADAPTRP

Proposé par M. Naidja *et al.* [24]. C'est un protocole de routage adaptatif basé sur plusieurs concepts, dont les plus importants sont l'hétérogénéité et l'auto-adaptation. L'objectif principal de ce protocole est d'améliorer la durée de vie des réseaux de capteurs sans fil hétérogènes en réduisant la consommation d'énergie nécessaire pour créer et adapter des clusters et cluster-heads. L'ADAPTRP utilise un ensemble de mécanismes qui ont été adoptés dans des protocoles bien connus (HEEP, APTEEN, LEACH, PEGASIS et H-PEGASIS) en tenant compte de leurs forces et de leurs faiblesses. Ce protocole est basé sur des cycles (Round) (figure 2.4) et chaque cycle est constitué de deux phases :

La phase d'initialisation

La station de base reçoit des informations sur l'état du nœud, l'emplacement et l'énergie

restante de chaque nœud pour élire de Cluster-Heads (CH) par formés les clusters, Aussi bien que déterminant le protocole de routage la plus appropriée pour ce cluster. Le protocole fonctionne à ce stade dans les étapes suivantes :

- Au début de la simulation, le nombre total de nœuds dotés de capteurs N est de 100 %, et chaque nœud du réseau envoie des informations sur le niveau d'énergie restant et l'emplacement à la station de base.
- Ensuite, la station de base calcule le nombre de ces nœuds, si ce nombre est inférieur à n / p . Dans ce cas, la station de base de RCSF est considérée comme un seul cluster. Cela signifie le changement le protocole fonctionné vers protocole PEGASIS au lieu le protocole HEEP. Autrement, si le nombre de nœuds dans la RCSF est supérieur à n / p , cela fonctionnera toujours à partir du protocole HEEP et la station de base analyse les détails et détermine les cluster-heads les plus appropriées pour ce cycle, c.-à-d, la plus grande capacité d'énergie.
- Ensuite, dans chaque cluster, la station de base calcule l'énergie restée. Si cette énergie diminue à moins de 50% donc le protocole fonctionné se change vers protocole APTEEN au lieu de protocole HEEP, parce que la station de base se concentre sur la conservation de l'énergie plutôt que sur la transmission de données.
- Autrement, si l'énergie résiduelle de la masse est supérieure à 50%, alors la station de base se concentre sur la transmission de données plutôt que sur la conservation de l'énergie, c'est-à-dire que le protocole HEEP passe à LEACH ou H-PEGASIS. Pour choisir le protocole fonctionnel entre les deux protocoles (LEACH et H-PEGASIS), la station de base applique des algorithmes appliqués à la densité, la consommation d'énergie et aux liaisons de communication.

La phase de transmission

La durée de cette phase est relativement plus longue que la phase d'initialisation, car elle est divisée en trames, et la trame à son tour est divisée en périodes de temps. La longueur de la trame varie d'un cluster à l'autre, selon le nombre de nœuds dans le cluster. Cette phase permet la collecte de données de tous les protocoles existants à envoyer à la station de base. Dans notre proposition, le RCSF considéré est hétérogène, parce que le protocole de routage ADAPTRP utilise plusieurs protocoles de routage en même temps.

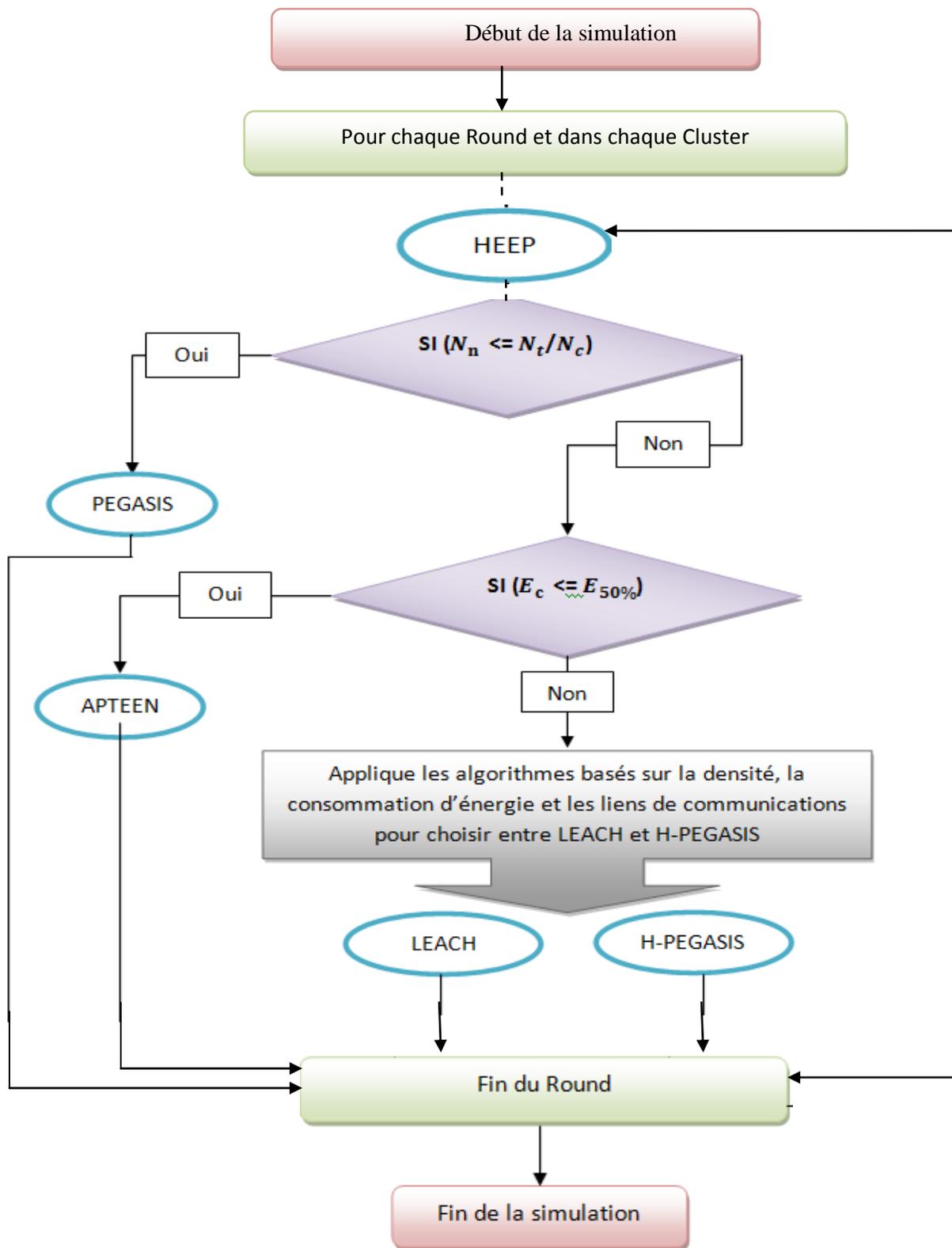


Figure 2.4 Algorithme d'ADAPTRP

N_n : Nombre des nœuds actuel. N_t : Nombre des nœuds total (début de la simulation).

N_c : Nombre de clusters dans le réseau. $E_{50\%}$: 50% d'Energie total dans ce cluster (début de la simulation). E_c : Energie total du cluster.

III. Partie 03 : Les réseaux hétérogènes RCSFH

1. Les réseaux hétérogènes

Les RCSF Hétérogènes représentent une nouvelle tendance adéquate et plus économique pour le contrôle et la surveillance des environnements à travers la capture, l'agrégation de mesure relatives à différents phénomènes physiques. Ces réseaux permettent d'opérer sur un même réseau sans avoir à intervenir pour l'adapter à chaque fois une nouvelle tâche ou à une nouvelle application. Une étude faite par les auteurs dans [2] a montré qu'un réseau hétérogène proprement déployé peut tripler le taux moyen de livraison et peut prolonger jusqu'à cinq fois plus la durée de vie du réseau.

2. Conditions de fonctionnement

D'abord, il faut que les capteurs parlent le même langage et comprennent le langage des autres nœuds du même RCSF, la communication doit être assurée entre tous les nœuds du réseau. En plus, les capteurs sont équipés des redirecteurs adéquats pour s'adresser à la station de base d'un autre système d'exploitation. Enfin, les stations de bases sont équipées des services adéquats pour recevoir les informations des nœuds capteurs d'un autre système d'exploitation.

3. Quelques modèles d'hétérogénéité

L'hétérogénéité est la possibilité que les nœuds dans un réseau soient différents. Il peut s'agir de l'hétérogénéité logicielle ou matérielle [27].

3.1. Hétérogénéité matérielle

- **Hétérogénéité des nœuds capteurs** : l'auteur dans [28] propose un modèle d'hétérogénéité matérielle, ce modèle contient un autre type de nœud dans le réseau nommé le nœud actionneur (Figure 2.5). Ainsi, un WSN (Wireless Sensors and Actuators Networks) est un réseau hétérogène par nature puisqu'il est constitué au moins de deux types de nœuds : les nœuds capteurs qui collectent les données de

l'environnement où ils sont déployés, et les nœuds actionneurs qui ne sont pas responsables seulement d'agir sur l'environnement physique.

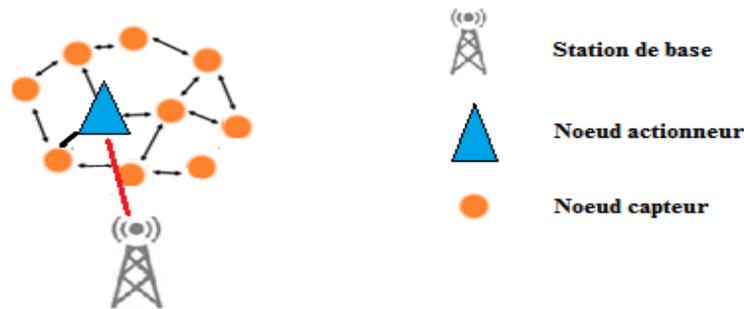


Figure 2.5 Exemple d'un cluster dans un RCSF et actionneur

- **Hétérogénéité de l'unité de capture** : l'unité de capture ou d'acquisition est composée de deux sous unités, un dispositif de capture physique qui prélève l'information de l'environnement local et un convertisseur analogique/numérique appelé ADC (Analog to Digital Converters) qui gère la conversion des signaux analogiques provenant de l'unité de capture en signaux numériques interprétables par l'unité de traitement [29]. Il est possible d'avoir un capteur avec plusieurs unités de capture (Figure 2.6), donc les données capturées sont envoyées périodiquement aux clusters-head pour collecter, agréger, traiter et envoyer le résultat final à la station de base.

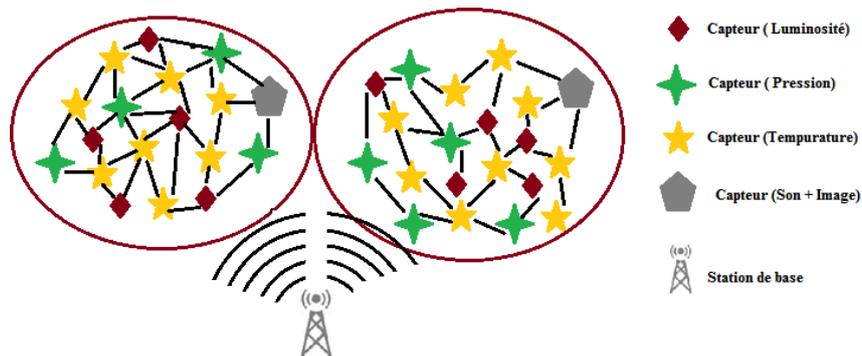


Figure 2.6 Hétérogénéité de l'unité de capture

3.2. Hétérogénéité logicielle

- **Hétérogénéité de systèmes d'exploitation** : Les systèmes d'exploitation représentent un socle sur lequel s'appuient les programmes d'application. Ils servent de lien ou d'interface entre l'architecture matérielle et la partie logicielle. Dans les réseaux de capteurs sans fil, les systèmes d'exploitation conservent ce rôle. Sachant que certaines architectures sont

plus puissantes que d'autres, il est logique d'extrapoler le concept au niveau logiciel. On peut donc avoir des systèmes d'exploitation qui offrent plus de caractéristiques, ou qui utilisent moins de mémoire flash/ram. Donc une possibilité non seulement d'hétérogénéité matérielle mais aussi logicielle.

- **Hétérogénéité du Protocole de routage** : Suivant la manière de création et de maintenance des routes, les protocoles de routage peuvent être répartis en trois catégories : proactifs, réactifs et hybrides.

Conclusion

Dans la première partie de ce chapitre, nous avons introduit la notion d'auto-organisation et nous avons présenté LEGOS le protocole d'auto-organisation qui a pour objectif principal de fournir une auto-organisation simple, efficace et adapté aux topologies à larges échelles. Dans la partie suivante, nous avons eu un aperçu sur plusieurs protocoles de couverture proposés dans la littérature. Généralement, les réseaux visés par notre proposition de protocole de routage hiérarchique basé sur une structure en clusters, comprennent des nœuds capteurs qui ne disposent pas d'information de localisation.

Nous avons pu voir dans la dernière partie les caractéristiques des réseaux hétérogènes, les WSANs comme exemple. Nous avons identifié les spécificités des WSANs, leurs architectures et leurs applications. La communication entre ces deux types de nœuds doit éviter les problèmes causés par la différence de portée de transmission. En effet, vu que les nœuds actionneurs ont généralement une grande portée de transmission par rapport à celle des nœuds capteurs, ces derniers doivent par exemple éviter les problèmes d'utilisation de liens unidirectionnels qui se présentent dans le réseau.

Chapitre 03 : Proposition d'une méthode de diagnostic dans les RCSF

Introduction

Une étape importante dans la supervision est le diagnostic de panne ou d'anomalie, qui consiste d'une part, à détecter et localiser une panne ou un dysfonctionnement, puis à rechercher la cause de cette panne, et enfin à agir sur le système pour le remettre dans un état de fonctionnement normal. Le diagnostic d'anomalie sera la généralisation du problème de détection au cas où plusieurs pannes peuvent affecter le système. Pour pouvoir appliquer le diagnostic et évaluer notre proposition, nous utilisons comme plateforme le protocole de routage connu LEACH.

1. Les problèmes de diagnostic dans les réseaux de capteurs sans fil

Les 4 principaux composants susceptibles de problèmes dans les réseaux de capteurs sans fil sont :

- L'unité de captage ;
- L'unité de traitement ;
- L'unité de transmission ;
- L'énergie.

2. Les hypothèses

Pour être effectif, un diagnostic cohérent doit tenir compte des hypothèses réalistes. Ainsi, les hypothèses suivantes de mieux cerner les problèmes :

- Aucun nœud ne couvre le nœud défaillant. Dans pareille situation, le recouvrement en cas de défaillance ne peut s'appliquer qu'à des pannes transitoires ;
- Le problème de capture d'un nœud est diagnostiqué au niveau du clusterhead associé en vérifiant que la valeur de captage reçue appartient à l'intervalle de vraisemblance (par exemple intervalle de température entre -20° et 60°), ou bien au niveau local s'il n'a aucune valeur captée ;
- Les problèmes d'énergie, transmission et traitement pouvant survenir au niveau d'un nœud sont localement diagnostiqué au niveau du nœud défaillant ;
- Un nœud défaillant au niveau captage peut être mis à profit au niveau traitement, stockage et transmission. De même, un défaillant en réception peut être utilisé en

capture, émission et stockage. Et un défaillant en traitement peut être utilisé en capture, transmission et stockage.

- Le choix de la période de la 2^{ème} chance dépend de type d'application ;
- La période de la 2^{ème} chance : c'est la période donnée à un nœud pour la confirmation de sa défaillance, c'est-à-dire à l'instant « t » si un nœud génère une défaillance de traitement par exemple, on enregistre ce nœud comme un nœud en doute et on attend un temps « r » c'est le temps de 2^{ème} chance, donc, à l'instant « $t + r$ » on lance de nouveau le traitement si ce nœud en doute génère une autre défaillance dans ce cas, il existe un diagnostic pour que le réseau reste avec un bon fonctionnement sinon pas de problème.
- On ne peut pas utiliser un nœud qui n'envoie pas. De même, un nœud qu'à un problème d'énergie ;
- Les trois types de fautes ne peuvent pas souvenir en même instant dans un même capteur.

3. Le déroulement de diagnostique

- ✓ **C'est quoi la panne ?** C'est un cas parmi la liste des quatre principaux problèmes dans les réseaux de capteurs sans fil.
- ✓ **C'est qui ?** Soit le cluster-Head ou un nœud ordinaire, ou bien station de base.
- ✓ **C'est qui le protocole de routage ?**
- ✓ **Le diagnostic :** chaque panne a une solution quelque soit le type de nœud ou bien le protocole de routage (cluster-Head ou ordinaire ou station de base). Donc chaque protocole possède diagnostic un seul saut et multi saut.

3.1. Problème d'énergie

L'énergie consommée par un nœud capteur est due essentiellement aux opérations suivantes : la capture, le traitement et la communication de données [26].

3.1.1. Le diagnostic d'énergie

- ✓ **C'est quoi la panne ?** Un problème au niveau d'énergie ;
- ✓ **C'est qui ?** Peut-être un cluster-Head ou un nœud ordinaire.
- ✓ **Le diagnostic**

Chapitre 3 Proposition d'une méthode de diagnostic dans les RCSF

- L'avantage principal d'auto-organiser le RCSF dans le cas d'une « panne d'énergie » est l'augmentation du nombre de données envoyés vers la station de base. Il est à noter que la station de base ne peut pas tomber en panne dans ce cas (panne d'énergie) parce qu'elle dispose une réserve d'énergie illimitée.
- Dans chaque cluster, quand un nœud donné atteint un seuil critique d'énergie, il est déclaré comme un nœud défaillant dit nœud crash.
- Au niveau d'un nœud CH : Le vice chef assurera les tâches du chef principale. Ce cas est très rare parce que le cluster-Head est choisi selon sa puissance d'énergie sauf peut-être dans le dernier round ou on a aucun autre CH.
- Au niveau d'un nœud ordinaire : le cluster-Head enlève le time-slot l'envoi de ce nœud pour gagner un espace de transmission (on peut gagner un ou plusieurs frames d'envoi de plus)

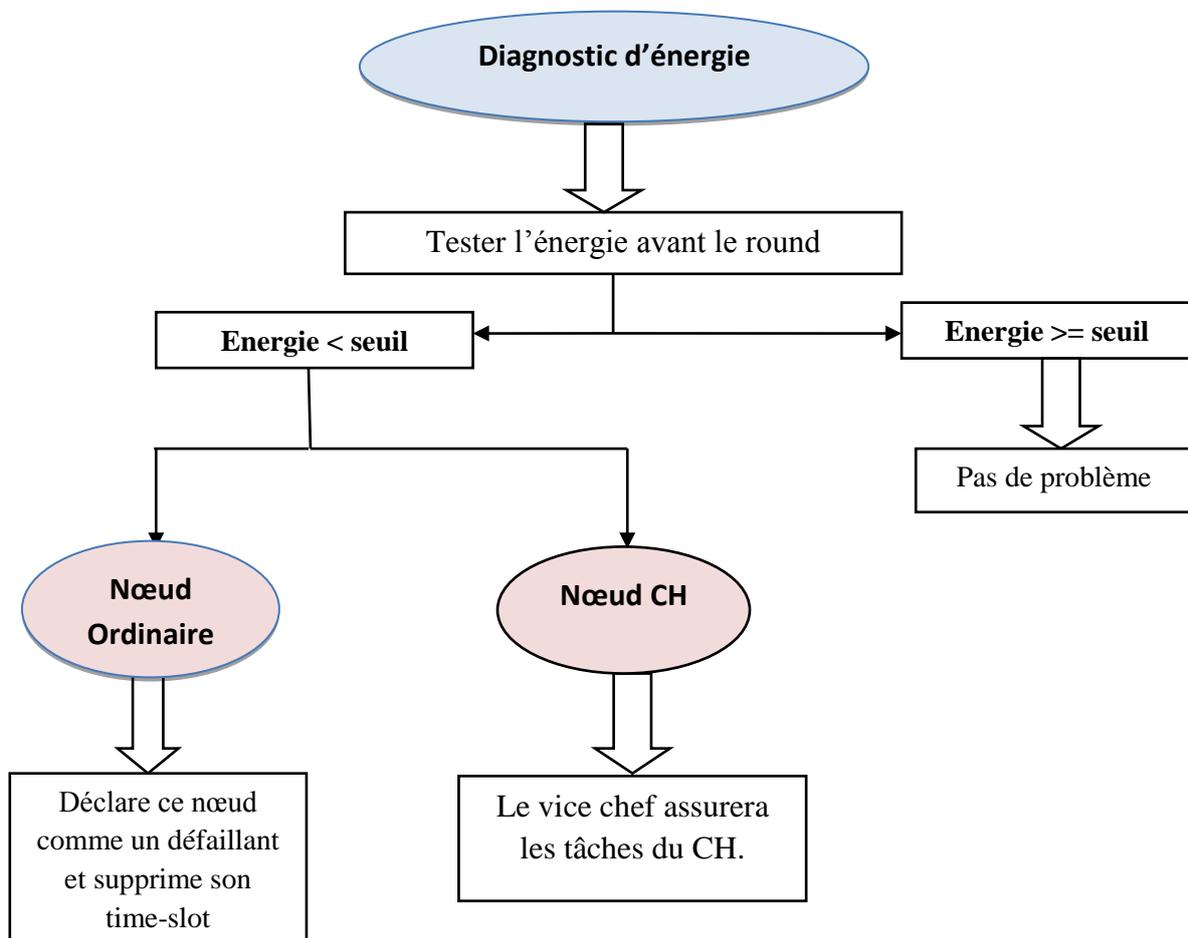
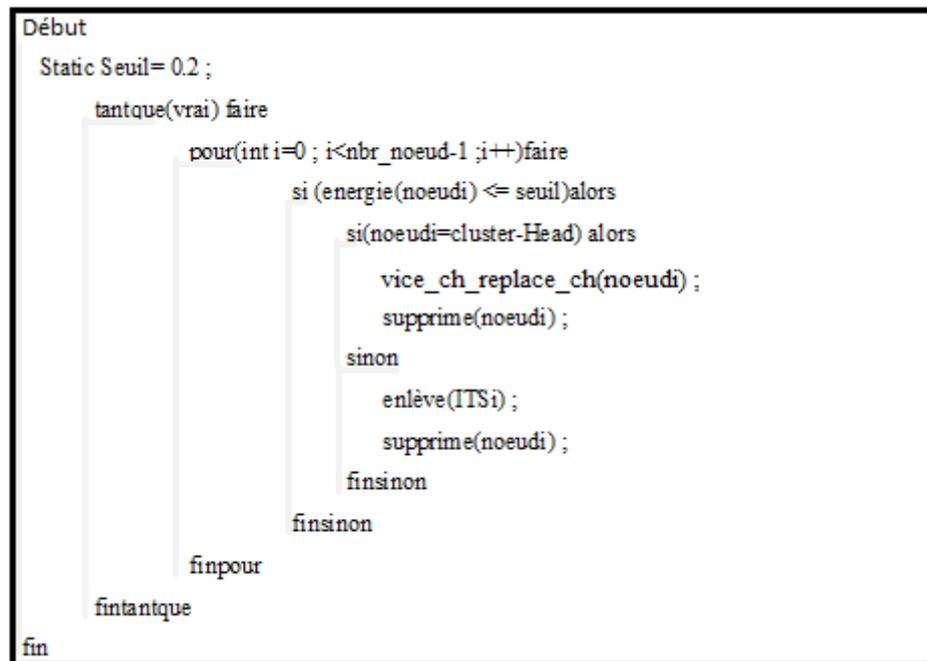


Diagramme 3.1 Diagramme de diagnostic d'énergie pour un nœud dans les RCSFs

Algorithme 3. 1 : Algorithme de diagnostic d'énergie pour un nœud dans les RCSFs



3.2. Problème de transmission

L'énergie de communication se divise en deux parties : l'énergie de réception et l'énergie de l'émission. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa puissance. Quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée.

3.2.1. Le diagnostic de transmission

L'unité de transmission est responsable d'effectuer toutes les émissions et réceptions des données sur un medium sans fil. Les composants utilisés pour réaliser la transmission sont des composants classiques, les unités de transmission de type radiofréquence (RF) sont préférables pour les RCSF parce que les paquets transportés sont de petites tailles avec un bas débit. Ainsi on retrouve les mêmes problèmes dans tous les réseaux sans fil : la quantité d'énergie nécessaire à la transmission augmente avec la distance. Pour les réseaux sans fil classiques (LAN, GSM) la consommation d'énergie est de l'ordre de plusieurs centaines de milliwatts alors que pour les réseaux de capteurs, le système de transmission possède une portée de quelques dizaines de mètres. Pour augmenter ces distances tout en préservant l'énergie, le réseau utilise un routage multi sauts. [25]

- ✓ **C'est quoi la panne ?** Un problème au niveau de l'unité de transmission, dans ce cas, il y a deux catégories de problèmes : soit un problème d'émission soit un problème de réception ;
- ✓ **C'est qui ?** Peut-être un cluster-Head ou un nœud ordinaire ou bien station de base.

✓ **Le diagnostic**

A. L'émission

1) Au niveau d'un nœud ordinaire :

- Les messages envoyés par un nœud ordinaire sont de deux types : message d'Info et DATA.
- Si un nœud n'a pas envoyé un message d'Info alors il ne peut pas recevoir de time-slot pour l'envoi des données au CH. Cette Panne peut être transitoire, c'est le cas de notre travail d'auto-organisation, donc ce nœud travail comme un CH c'est-à-dire il envoie ses données directement à la station de base, ou bien on supprime ce nœud dans le cas d'une faute permanente.
- Si un nœud n'a pas envoyé un message de DATA pendant son time-slot, il est marqué comme un nœud en doute dans une structure de donnée. Dans le cas où la communication est un seul saut alors le CH donne à ce nœud une deuxième chance, si le même problème se reproduit alors la suppression de ce nœud est établie sinon il continue son travail comme un nœud normal. Par contre, si la communication dans ce cluster est multi saut, alors le nœud ordinaire en panne sera marqué comme un nœud douteux dans une structure de données. Le successeur de ce nœud défaillant lui donne une deuxième chance et si le même problème persiste, le successeur envoie un message vers le prédécesseur du nœud défaillant pour l'informer de lui adresser les données.

2) Au niveau d'un nœud CH :

- Le message DATA est le seul message envoyé par un nœud CH vers la station de base, dans ce cas le vice-CH prend la tâche d'émission des données.

3) Au niveau de la station de base :

- Les messages envoyés par la station de base sont de type INFO et DATA.
- Si cette dernière n'a pas envoyé le message d'INFO vers un ou plusieurs nœuds dans le RCSF, alors les membres qui n'ont pas reçu ce message travail comme un CH.

- Si la station de base n'a pas envoyé le message DATA, les données n'atteindront jamais l'utilisateur final. Dans ce cas une station de base secondaire (SBS) prendra le relais pour émettre les données sans passer par l'idée de la deuxième chance, par ce que le statut de la station de base dans le RCSF est essentielle, elle représente l'ensemble des données collecter par le réseau.
- Aucun problème lorsque le CH n'a pas envoyé les deux premiers messages parce que dans ce cas, les membres qui n'ont pas reçu l'un ou les deux messages travail comme un CH.

B. La réception

1) Au niveau d'un nœud ordinaire :

- Les messages reçus par un nœud ordinaire sont deux types : message INFO et parfois DATA dans le cas d'une communication multi-saut.
- Si un nœud ordinaire n'a pas reçu le message de DATA. Dans ce cas ce nœud est enregistré comme un nœud douteux et on lui accorde une deuxième chance, quand le même problème de réception revient alors on observe deux cas, le premier entre dans le cadre d'une communication multi saut dans le cluster, dans ce cas le nœud récepteur envoie un message d'alerte vers son propre prédécesseur pour que le prochaine envoie des données sera orienté vers le successeur du nœud défaillant, par contre, dans le cas où la communication est un seul saut, ce problème est impossible, parce que chaque nœud ordinaire dans le cluster communique directement avec son CH

2) Au niveau d'un nœud CH :

- Les messages reçus par le cluster-Head sont de deux types : message de INFOJoin et DATA.
- Le CH ne compte pas un nœud ordinaire dans son TDMA lorsqu'il n'a pas reçu le message INFOJoin envoyé par la station de base pour l'informer que ce nœud est un membre dans le cluster. Ce dernier travaille donc comme un CH c'est-à-dire, il envoie les DATA directement vers la station de base quel que soit le type de la communication un sel saut ou bien multi saut, parce que dans ce cas, les membres qui ne reçoivent pas le message d'INFOADV INFOTDMA travaille comme un CH, mais le problème se pose lorsque le CH n'a pas reçu les messages de DATA. Dans ce cas, le CH est enregistré comme un nœud douteux et on lui accorde une deuxième chance. Si le même problème de réception persiste, alors il faut informer les membres qui composent ce cluster pour changer d'envois de leurs Data au vice chef, ce dernier

prendre les tâches réception et d'émission des données dans ce round, et ce supprime le CH défaillant.

3) Au niveau de la station de base :

- La station de base secondaire (SBS) envoie un message Hello vers la station de base primaire (SBP). A la réception de ce message, la SBP répond à ce message pour garantir le bon fonctionnement.
- Le message reçu par la SB est de type DATA. En cas d'un manque de réponse, les données n'atteindront jamais l'utilisateur final, dans ce cas la SBS sera chargée pour réceptionner les données envoyées par les CHs.

Algorithme 3. 2 : Algorithme de diagnostic de réception pour un nœud dans les RCSFs

```
Début
tantque(vrai)faire
  si(? reçu(nœudi,msg,nœudj))alors
    //il n'a pas un message msg reçu par nœudj depuis le nœudj
    si(nœudi==CH)alors
      si (msg == INFOjoin) alors
        //le problème de réception de msg de JOIN par le cluster-Head depuis le nœudj
        CH ne compte pas nœudj dans leur TDMA ;
        //dans ce cas le nœudj travail comme un CH
      sinon //problème de réception d'un msg de DATA
        si(tab_out[i]==1)alors
          tab_out[i]=0 ; //pour la deuxième chance pour ce CH
        sinon
          si(tab_out[i]==0)alors
            tab_out[i]=2 ; //dans le cas de deuxième chance
          sinon //tab_out[i]==2
            informe les membre pour ne pas envoyer leurs DATA ;
            vice_ch_replace_ch(nœudi);
          finsinon
        finsinon
      finsinon
    sinon //nœudi !=CH
      si (msg == INFOadv) alors
        // le nœudi n'a pas reçu un msg d' ADV depuis le nœudj « CH »
        (? envoie(nœudi,JOIN,nœudj)) //le nœudi n'envoie pas un msg de JOIN
      sinon //le msg== INFOtdma
        (? envoie(nœudi,DATA,nœudj)) ; //le nœudi n'envoie pas un msg de DATA
      finsinon
      nœudi travail comme un CH ;
    finsinon
  finsi
fintantque
fin
```

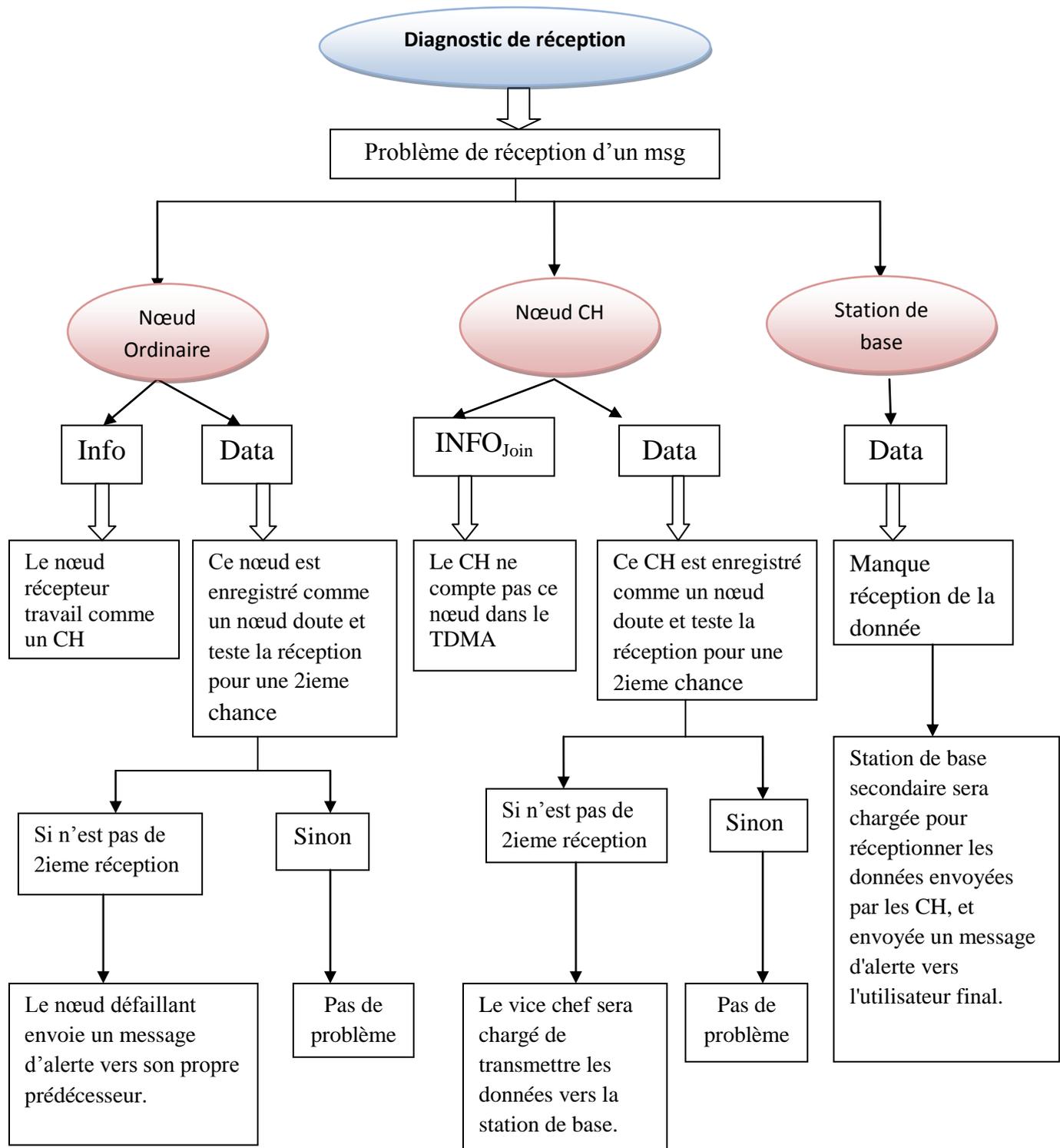


Diagramme 3.2 Diagramme de diagnostic de réception pour un nœud dans les RCSFs

3.3. Problème de capture

Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les convertisseurs analogique-numérique convertissent ces signaux analogiques en signaux numériques.

3.3.1. Le diagnostic de captage

- ✓ **C'est quoi la panne ?** Un problème au niveau d'unité de captage ;
- ✓ **C'est qui ?** Peut-être un cluster-Head ou un nœud ordinaire.
- ✓ **Le diagnostic**

Généralement, un nœud qui ne capte pas n'envoie pas, dans ce cas on a besoin de faire appel au diagnostic de transmission et exactement au diagnostic d'émission.

Il existe des applications qui peuvent utiliser des valeurs de capture proches (par exemple la température) dans ce cas aucun problème, mais le problème se pose si la différence entre les valeurs des nœuds voisins dépasse un certain seuil, pour cela deux cas possibles. Le premier, si ce nœud est un cluster-Head alors il fonctionne normal parce qu'il reçoit des DATA et les envoie vers la station de base ; et le deuxième cas c'est un nœud ordinaire : le diagnostic se fait au niveau du clusterhead à la réception de la valeur captée qui envoie un message d'alerte vers la station de base lorsque une deuxième valeur erronée est reçue depuis un autre capteurs, et comme le nombre de messages d'alerte augmente au niveau de la station de base alors, il faut vérifier cette zone où le problème est apparu.

Algorithme 3.3 Algorithme d'un diagnostic de captage pour un nœud dans les RCSFs

```
Debit
  tantque(vrai)faire
    pour(int i=0 ; i<nbr_noeud-1 ;i++)faire
      si(? captage(noeudi) )alors
        //si le noeudi ne capte pas il n'envoie pas dans ce cas il faut
        // lancer une appelle vers la fonction d'émission pour contenu
        //le traitement du diagnostic
        emission(noeudi) ;
      finsi
    finpour
  fintantque
fin
```

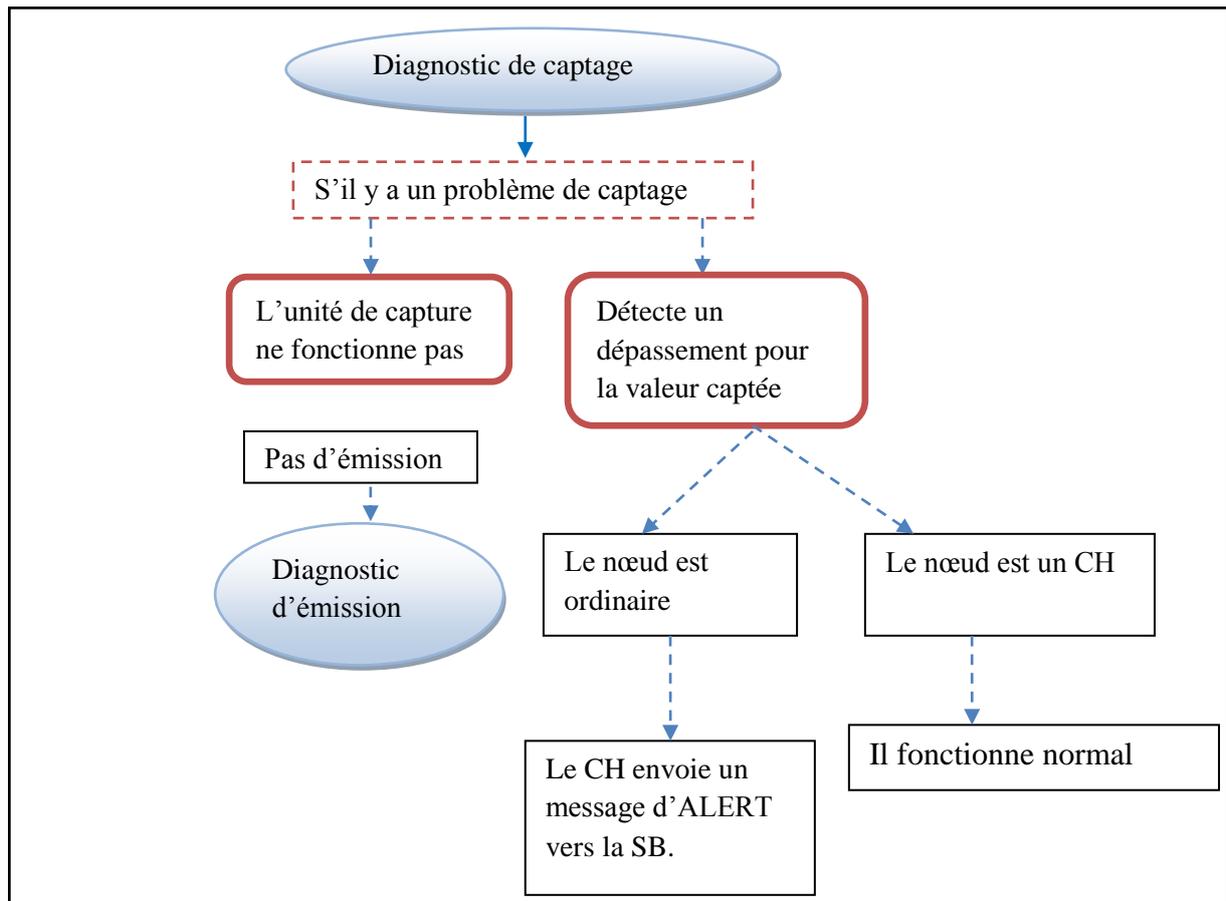


Diagramme 3.3 Diagramme de diagnostic de captage pour un nœud dans les RCSFs

3.4. Problème de traitement

L'unité de traitement comprend un processeur avec une petite unité de stockage, une RAM pour les données et une ROM pour les programmes et souvent une mémoire flash. Cette unité fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs (NS2 par exemple). Elle est chargée de gérer des procédures qui permettent à un nœud capteur de collaborer avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits [25].

3.4.1. Le diagnostic de traitement

- ✓ **C'est quoi la panne ?** Un problème au niveau d'unité de traitement c'est-à-dire qu'il y a un problème de calcul
- ✓ **C'est qui ?** Peut-être un cluster-Head ou un nœud ordinaire ou bien station de base.
- ✓ **Le diagnostic**

Chaque nœud exécute un calcul pour chaque début de round dans le cadre d'une vérification de l'unité de traitement, pour cela, on prend par exemple : le calcul de $(a + b)$?

Chapitre 3 Proposition d'une méthode de diagnostic dans les RCSF

Chaque nœud fait le calcul à son niveau ensuite, il compare le résultat obtenu avec le résultat déclaré au départ. Par exemple, dans chaque nœud on déclare $(a = 3)$, $(b = 5)$, $(c = 8)$, ensuite, chacun fait son propre calcul « $a + b$ » et compare avec « c » si les résultats sont identiques alors pas de problème, sinon les nœuds erronés sont enregistrés pour leur donner une deuxième chance. Ensuite, si le résultat d'un nœud enregistré est vrai alors pas de problème, sinon, ce nœud défaillant ne sera plus accepté comme cluster-Head ni comme station de base, mais il sera utilisé pour le captage, la transmission ou un espace de stockage.

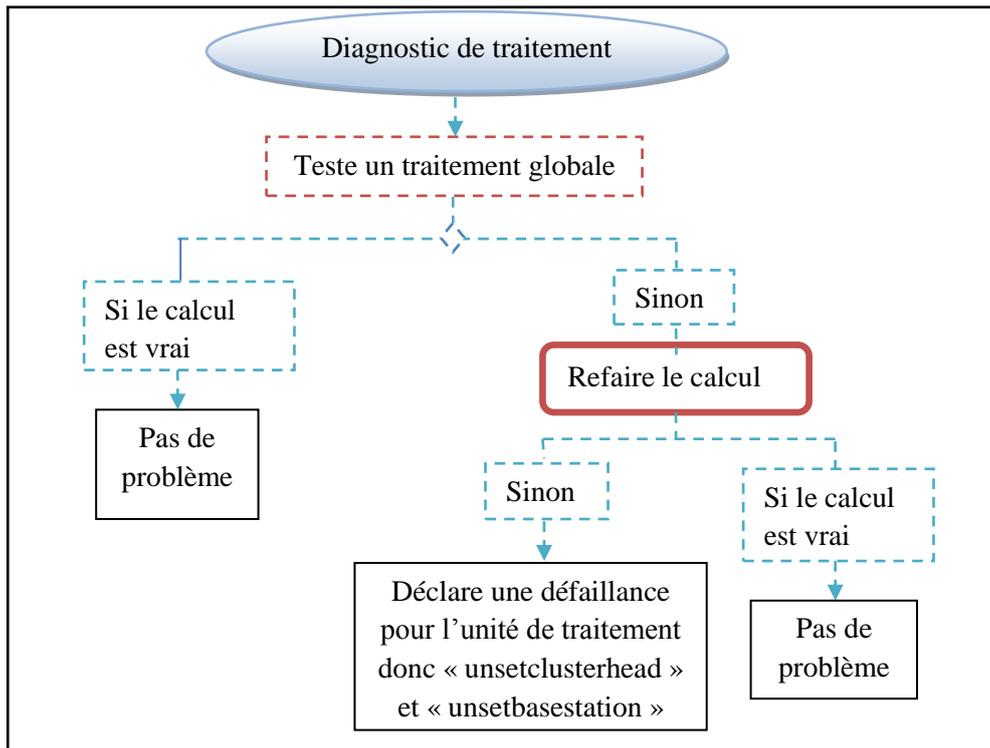


Diagramme 3. 4 : Diagramme d'un diagnostic de traitement pour un nœud dans les RCSFs.

Algorithme 3.4 : Algorithme de diagnostic de traitement pour un nœud dans les RCSFs

```
Début
static a = 3, b = 5, c = 8;
var d;
pour(chaque round) faire
    pour(int i=0 ; i<nbr_noeud-1 ;i++)faire
        d=a+b ;
        si( d == c)alors
            //pas de problème c a d un bon fonctionnement pour l'unité de traitement
        sinon
            d= a+b ;
            //il faut refaire le calcul pour une deuxième chance
            si( d == c)alors
                //pas de problème c a d un bon fonctionnement pour l'unité de traitement
                //dans la 2ieme chance
            sinon
                // Declar une défaillance de traitement pour ce noeud
                Unsetbasestation(noeudi);
                Unsetclusterhead(noeudi);
                //on ne peut pas voir ce noeud comme un CH ou une station de base
            fin sinon
        finsinon
    finpour
finpour
fin
```

Conclusion

Dans ce chapitre, nous avons proposé une méthode pour le diagnostic dans les réseaux de capteur sans fil, applicable au protocole de routage hiérarchique « LEACH » ; cette méthode consiste à traiter les défaillances apparaissant dans le réseau par la détection, la localisation et le travail à faire pour continuer le fonctionnement normal du système.

Chapitre 04 : Mise en œuvre et simulation

Introduction

Tel qu'on l'a montré au cours de l'étape de conception, l'objectif principal de notre première contribution est la mise en œuvre d'une solution qui se charge d'auto-organiser le protocole de routage **ADAPTRP**. Notre premier but est d'atteindre un niveau d'auto-organisation acceptable sans dégrader les performances du réseau. De ce fait, nous avons établi un nouveau protocole **ADAPTRP-AO** qui est en mesure de détecter et localiser les défaillances dans le protocole de routage **ADAPTRP**.

L'objectif de ce chapitre est donc de démontrer l'efficacité du protocole **ADAPTRP-AO** par rapport au protocole racine **ADAPTRP**. Nous avons pu montrer à travers les résultats de simulation obtenus que les objectifs de nos protocoles aient été atteints.

Pour cela, nous commencerons par définir les outils nécessaires pour l'implémentation et la simulation de protocole. Ensuite, nous décrirons la mise en œuvre de toutes les structures de données et processus décrits lors de la conception. Nous terminerons ce chapitre par une présentation des résultats relevés lors des tests de performances de protocole **ADAPTRP-AO**.

La simulation des réseaux de capteurs consiste principalement en la reproduction du comportement et du fonctionnement des nœuds capteurs dans un environnement informatique. Il sera donc question, pour ce chapitre, de faire la présentation d'un simulateur NS2, puis aborder les simulateurs eux-mêmes.

1. Environnement de simulation

Dans cette section, nous présentons les outils utilisés pour la mise en œuvre de protocoles **ADAPTRP-AO**. Nous commençons tout d'abord par le simulateur NS2 pour simuler et évaluer les différents paramètres tels que les données envoyées, l'énergie consommée et le nombre des nœuds vivant. Nous parlons ensuite du UBUNTU, le système d'exploitation adapté aux RCSF hétérogènes. Nous terminons cette partie par la présentation du langage de programmation TCL avec lequel nous avons programmé les codes de protocole.

1.1. Objectif de la simulation

Le déploiement d'un réseau de capteurs nécessite une phase de tests avant sa mise en place afin de s'assurer du bon fonctionnement du réseau. Une expérimentation directe effectuée sur le terrain peut se révéler coûteuse, irrationnelle ou même impossible. Il serait même inconcevable dans notre étude de mettre en œuvre un RCSF, de changer les paramètres pour comparer deux ou plusieurs algorithmes de routage.

La simulation des réseaux de capteurs consiste principalement en la reproduction du comportement des nœuds capteurs et des interactions entre eux. C'est une étape incontournable pour l'évaluation des modèles d'application ou des protocoles de communication. De plus, la simulation offre un gain considérable en temps, une flexibilité en permettant la variation des paramètres et une meilleure visualisation des résultats.

1.2. Choix du simulateur NS2

Beaucoup de simulateurs pour les réseaux informatiques ont été développés pour répondre aux attentes des utilisateurs. Parmi ces simulateurs, nous pouvons citer : OPNET, QualNet, NetSim, NS2, OMNeT++, ...etc. Nous avons choisi le simulateur NS2 (Network Simulateur, version 2).

NS-2 est un logiciel de simulation open source gratuit (libre) à évènements discrets. Il est développé dans le cadre du projet VINT qui regroupe plusieurs laboratoires de recherche comme AT&T institut de recherche à Berkeley (ACIRI), Xerox PARC et Sun Microsystems.

Ce simulateur est développé en C++ et OTCL (**O**bject **T**ools **C**ommand **L**anguage), est utilisé le langage TCL comme langage de création des scénarios de simulation. Il supporte les réseaux sans fil et filaires, avec plusieurs protocoles des différentes couches (Physique, MAC, réseaux, transport ... etc.). L'outil NAM (**N**etwork **A**nimator) associé au simulateur NS permet de visualiser des animations de la simulation (transfert des paquets d'un nœud à un autre, taille des paquets, remplissage des files d'attente, ...etc.)

Avantages

- NS est un logiciel de simulation multicouches ;
- Son développement est orienté objet ;
- Permet l'ajout de composants à la demande ;
- C'est un outil complètement libre et disponible pour plusieurs plateformes ;
- C'est l'un des simulateurs gratuits les plus documentés ;

- Vu sa popularité, plusieurs protocoles sont à priori disponibles, aussi bien pour les réseaux ad-hoc que pour les réseaux filaires.

Inconvénients

- Initialement conçu pour les réseaux filaires, le support des réseaux sans fil a été ajouté ultérieurement ;
- Contient peu de paramètres de configuration dans ses modèles standards ;
- Introduit des dépendances entre les modules ce qui rend difficile l'ajout de nouveaux modèles ;
- Les performances du simulateur sont assez limitées et ne semblent pas adaptées pour des simulations de réseaux importants.

1.3. **Choix de plateforme OS**

NS2 étant disponible sous deux grandes plateformes OS :

- Sous Windows : en l'installant sous un émulateur tel que Cygwin.
- Sous Unix.

Après étude des deux environnements, nous avons choisi Linux-UBUNTU. Ce dernier offre tous les outils logiciels nécessaires (langages, outils graphiques, ...etc.).

1.4. **Choix de langage**

Nous avons choisi comme langage d'implémentation de nos protocoles ADAPTRP-AO le langage TCL (**T**ool **C**ommand **L**anguage). Ce dernier est connu comme étant un langage de commandes interprétées extensif. En effet, les programmes écrits en TCL sont des fichiers texte constitués de commandes TCL qui est traitées via un interpréteur TCL au moment de l'exécution. L'avantage d'implémenter notre algorithme en TCL est de pouvoir facilement l'interpréter avec l'interpréteur TCL, ce qui nous permet de simuler son fonctionnement afin d'évaluer ses performances.

Nous avons choisi comme plate-forme d'implémentation, le simulateur NS2 version NS2.34, sous le système d'exploitation LUNIX UBUNTU 10.10.

2. Etapes d'implémentation de notre protocole

L'implémentation de notre protocole passe par plusieurs étapes à savoir :

2.1. Préparation de l'environnement d'implémentation

La préparation de l'environnement d'implémentation consiste à installer le simulateur de réseau NS2 sous le système d'exploitation UBUNTU. On a choisi la version NS2.34 puisqu'elle prend en considération la topologie des réseaux sans fil et le protocole que nous étudions ADAPTRP-AO se base sur ce simulateur.

L'installation de NS2.34 sous UBUNTU 10.10 s'effectue suivant les étapes qui se résument en :

- Télécharger le paquet **ns-allinone-2.34.tar.gz**.
- Placer le paquet **ns-allinone-2.34.tar.gz** dans le **home** par exemple (/home/master/ ns-allinone-2.34.tar.gz)
- Cliquer à droite de la souris sur **ns-allinone-2.34.tar.gz** ensuite choisir « extract here » pour extraire le contenu dans le même répertoire.
- Ouvrir le terminal (Applications→Accessories→Terminal)
- Changer vers la direction de ns-allinone-2.34.

```
$ cd /home/ master /ns-allinone-2.34
```

- Taper la commande suivante :

```
$ sudo apt-get install build-essential autoconf automake libxmu-dev  
gcc-4.3
```

- Ouvrir **Makefile.in** qui existe dans la localisation : **ns-allinone-2.34/otcl-1.13/Makefile.in**

- Trouver la ligne suivante : **CC=@CC@** et changer vers : **CC= gcc-4.3**

- Pour devenir un super utilisateur, taper la commande suivante :

```
$ sudo su
```

- Pour démarrer l'installation taper «./install »
- Si l'installation est terminée sans erreurs, il faut ajouter les variables d'environnement dans un fichier « .bashrc »

```
$ gedit ~/.bashrc
```

```
# LD_LIBRARY_PATH
```

```
OTCL_LIB=/home/master/ns-allinone-2.34/otcl-1.13
NS2_LIB=/home/master/ns-allinone-2.34/lib
X11_LIB=/usr/X11R6/lib
USR_LOCAL_LIB=/usr/local/lib

export
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$OTCL_LIB:$NS2_LIB:$X11_LIB:$USR_LOCAL_LIB

# TCL_LIBRARY

TCL_LIB=/home/master/ns-allinone-2.34/tcl8.4.18/library
USR_LIB=/usr/lib
export TCL_LIBRARY=$TCL_LIB:$USR_LIB

# PATH

XGRAPH=/home/master/ns-allinone-2.34/bin:/home/master/ns-allinone-2.34/tcl8.4.18/unix:/home/master/ns-allinone-2.34/tk8.4.18/unix
# Note: the above two lines starting from XGRAPH should come in the same line
NS=/home/master/ns-allinone-2.34/ns-2.34/
NAM=/home/master/ns-allinone-2.34/nam-1.14/
PATH=$PATH:$XGRAPH:$NS:$NAM
```

- A la fin taper :

```
$ source ~/.bashrc
```

- Installer **xgraph** à l'aide de la commande :

```
sudo apt-get install xgraph
```

Après avoir installé le simulateur de réseau NS2.34, il faut installer le protocole de routage ADAPTRP. Pour ce faire, il faut suivre les étapes suivantes :

- Obtenir le package d'installation du protocole ADAPTRP « ns-234-ADAPTRP.tar.gz ».
- Obtenir le fichier bash « **adaptrp-setup.sh** ».
- Placer les deux dans le chemin suivant : « **/home/master/ns-allinone-2.34/ns-2.34** »
- Décompresser le package d'installation dans le même répertoire « **ns-2.34** »
- On remarque que le répertoire d'accueil de « ns-2.34 » est autre que "/opt/...", alors nous avons besoin de trouver "/opt/ns-allinone-2.34" et le remplacer par "/home/master/ns-allinone-2.34" pour le dossier " **ADAPTRP-setup.sh**".
- Déplacer vers le répertoire "**/home/master/ns-allinone-2.34/ns-2.34**" et patch le fichier "**ADAPTRP-setup.sh**"

```
wsn@ubuntu:~$ cd /home/master/ns-allinone-2.34/ns-2.34/
```

```
wsn@ubuntu:/home/master/ns-allinone-2.34/ns-2.34$ sudo  
bash ADAPTRP-setup.sh
```

- Trouver "**/opt/ns-allinone-2.34**" et le remplacer par "**home/master/ns-allinone-2.34**" pour les deux fichiers:
 - ✓ «Makefile»: on le trouve dans le répertoire "home/master/ns-allinone-2.34/ns-2.34".
 - ✓ «Makefile.in»: on le trouve dans le répertoire "home/master/ns-allinone-2.34/ns-2.34".
- Remplacer **CC** et **CPP** de « Makefile » et « Makefile.in » existe dans « ns-234-ADAPTRP » par

```
CC = gcc-4.3
```

CPP = g++-4.3

- Exécuter les commandes suivantes :

```
wsn@ubuntu:/opt/ns-allinone-2.34/ns-2.34$./configure
```

```
wsn@ubuntu:/opt/ns-allinone-2.34/ns-2.34$make clean
```

```
wsn@ubuntu:/opt/ns-allinone-2.34/ns-2.34$make
```

- Si l'exécution de la commande précédente est réussie alors on peut tester ADAPTRP par l'exécution de la commande «./test »

```
wsn@ubuntu:/opt/ns-allinone-2.34/ns-2.34$./test
```

- Vérifier le fichier « **ADAPTRP.err** », s'il n'y a pas d'erreurs, aller vers le fichier « **ADAPTRP.out** » et voir le résultat.

2.2. Implémentation

Après avoir préparé notre environnement d'implémentation, on entame l'étape d'implémentation de nos protocoles qui consiste à modifier le code d'implémentation du protocole ADAPTRP afin qu'il corresponde à l'architecture de fonctionnement de nos protocoles. Pour cela, on a ajouté diverses fonctions et procédures pour l'auto-organisation.

En effet, le code d'implémentation proposé est composé de plusieurs fichiers TCL dont chacun joue un rôle bien précis. Ainsi, et pour réutiliser ce dernier, nous avons dû analyser et comprendre le rôle de chacun de ces fichiers. Etant donné qu'un fichier TCL est présenté comme une classe qui contient plusieurs sous classes, le protocole proposé est vu comme un ensemble de classes qui se communiquent pour réaliser une tâche commune.

Le protocole proposé est une application spécifique des protocoles d'auto-organisation. Il est implémenté comme sous classe par rapport à la classe Applications du simulateur NS2.

3. Évaluation des performances du protocole ADAPTRP-AO

Cette section présente les paramètres et les résultats de la simulation. Dans notre travail, nous utilisons le simulateur NS2 avec une centaine de nœuds répartis d'une façon aléatoire et déterministe dans une zone de $100m * 100m$, les coordonnées de position de la station de base sont $SB (50,175)$ comme indiqué dans la figure 4.1. Nous supposons que tous

les nœuds ont une position fixe pendant toute la période de simulation. Les paramètres de notre simulation sont donnés dans le tableau 4.1.

La simulation conduite évalue dans un premier temps les performances du mécanisme d’auto-organisation proposé dans les RCSF hétérogènes, puis la simulation des deux protocoles de routages ADAPTRP et ADAPTRP-AO. Ce dernier donne de meilleurs résultats pour le nombre de données reçues par la station de base et le nombre des nœuds défaillants détectés, ...etc.

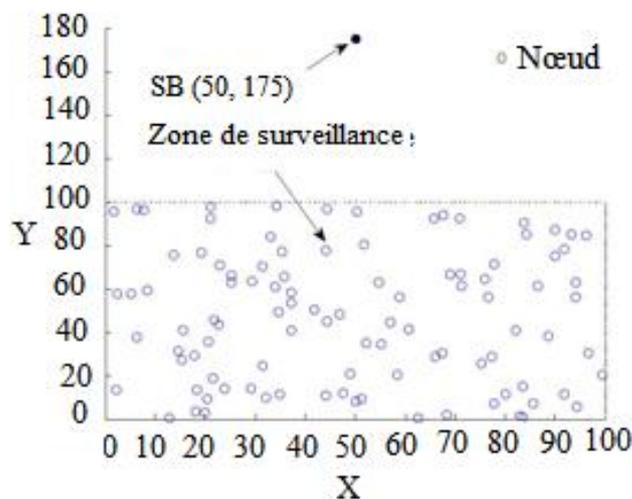


Figure 4.1 Positions des nœuds et de la station de base

Paramètres	Valeur
Localisation de la SB	(50,175)
Nombre des noeuds	100
Nombre de clusters	5
Nombre des nœuds sophistiqués	20
La taille des paquets	512 Bits
La taille des paquets de contrôle	8 Bits

Energie des nœuds initiaux	70 nœuds → 2J BS → 5000J
Energie des nœuds ajoutés	30 nœuds → 16J
Simulator	NS2

Tableau 4.1 Paramètres de simulation

❖ Flux de données transmis

La figure 4.2 représente la quantité de données envoyée en fonction du temps de simulation. On remarque que le nombre de paquets envoyés par le protocole de routage ADAPTRP (Multi-protocole) est strictement grand.

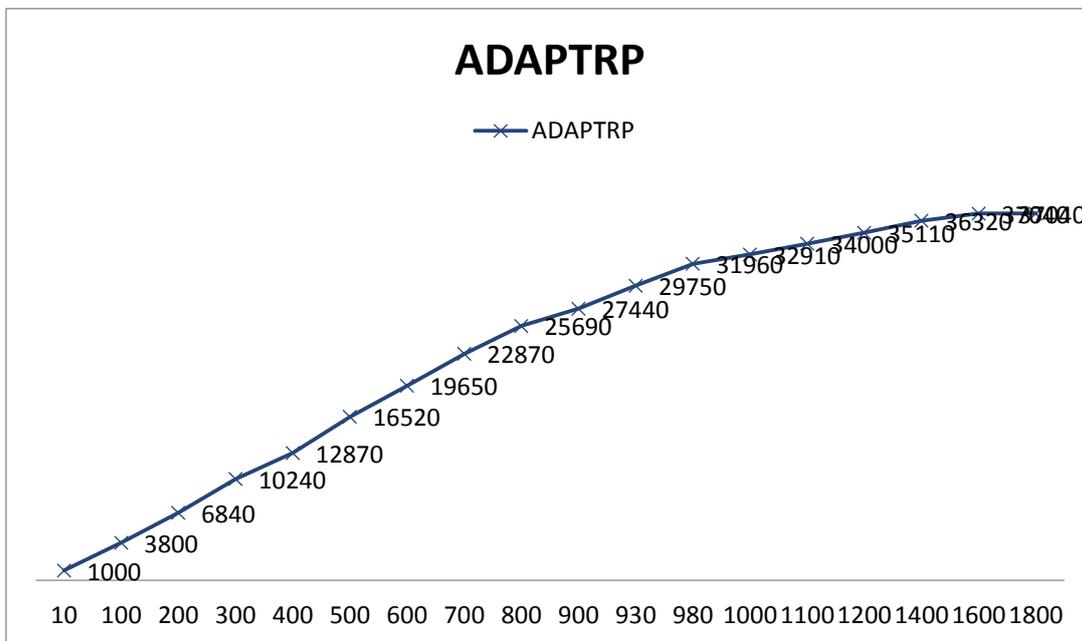


Figure 4.2 Quantité de données reçues en fonction de la durée de la simulation

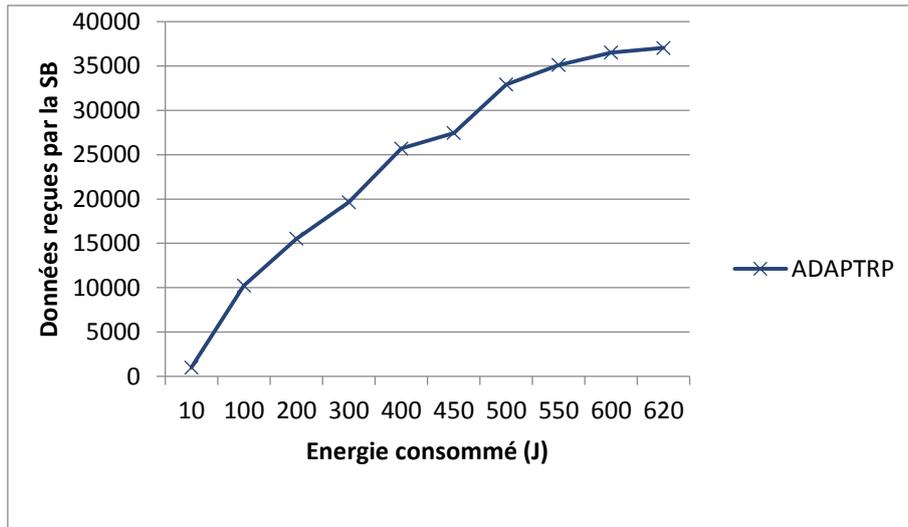


Figure 4.3 Quantité de données reçues par la SB en fonction de l'énergie consommée

La figure 4.3 représente la quantité de données reçue par la station de base en fonction de l'énergie consommée. A partir de cette figure, on remarque que le nombre de paquets envoyés par le protocole de routage ADAPTRP (Multi-protocole) vers la station de base est strictement grand. Donc, il est évident que considérant la quantité de données reçue par la station de base en fonction de la durée de la simulation ou la quantité de données transférée vers la station de base en fonction de la consommation d'énergie, parce que dans le protocole ADAPTRP, la station de base est consciente des informations sur l'emplacement et l'énergie de tous les nœuds du réseau, ce qui permet à la station de base d'appliquer les algorithmes basés sur la densité, la consommation d'énergie et les liens de communication.

❖ **Durée de vie**

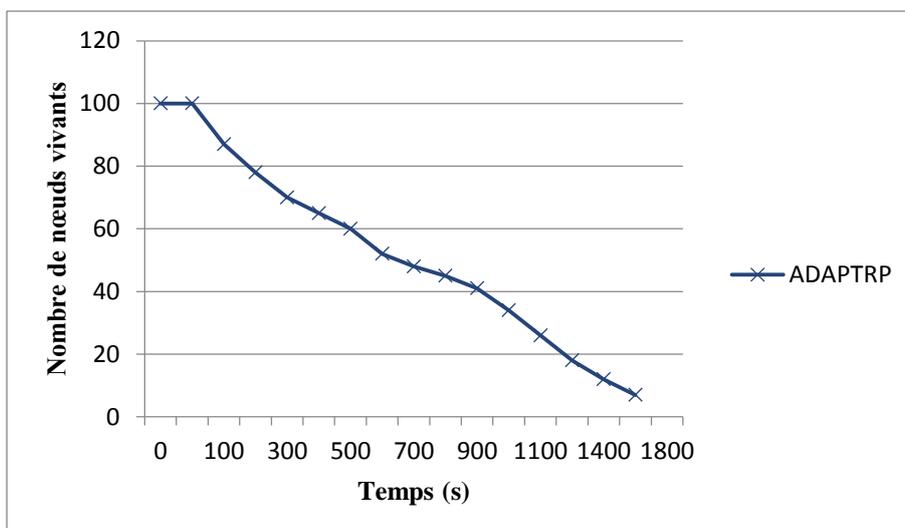
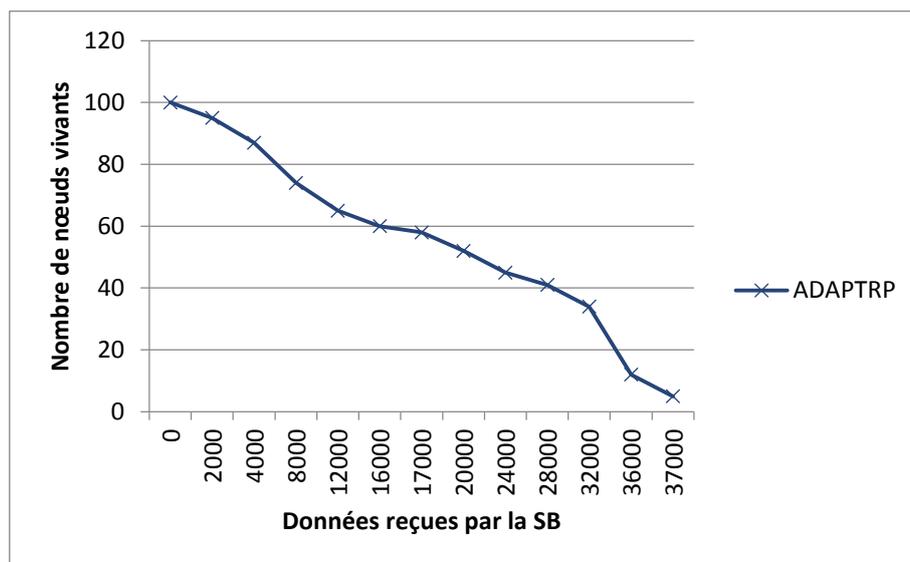


Figure 4.4 Nombre de nœuds vivants en fonction de la durée de la simulation

Le premier objectif d'auto-organisation pour les RCSF hétérogènes est d'augmenter la durée de vie ; pour approfondir l'évaluation des performances de protocole, nous avons mesuré le pourcentage de nœuds morts pendant toute la durée de la simulation, ce qui nous a donné le résultat, comme montré dans le graphique représenté dans la figure 4.4.

Initialement, tous les nœuds sont vivants, mais après un certain temps de simulation, on remarque la mort et la diminution des nœuds vivants faiblement mais la durée de vie du réseau pour le protocole ADAPTRP est beaucoup longue car ce protocole basé sur des protocoles de routage hétérogènes et prend en compte les avantages et évite les inconvénients des autres protocoles pour prolonger la durée de vie du réseau dans la figure 4.5.

**Figure 4.5** Nombre de nœuds vivants en fonction de la quantité de données reçues par la SB

❖ L'énergie consommée

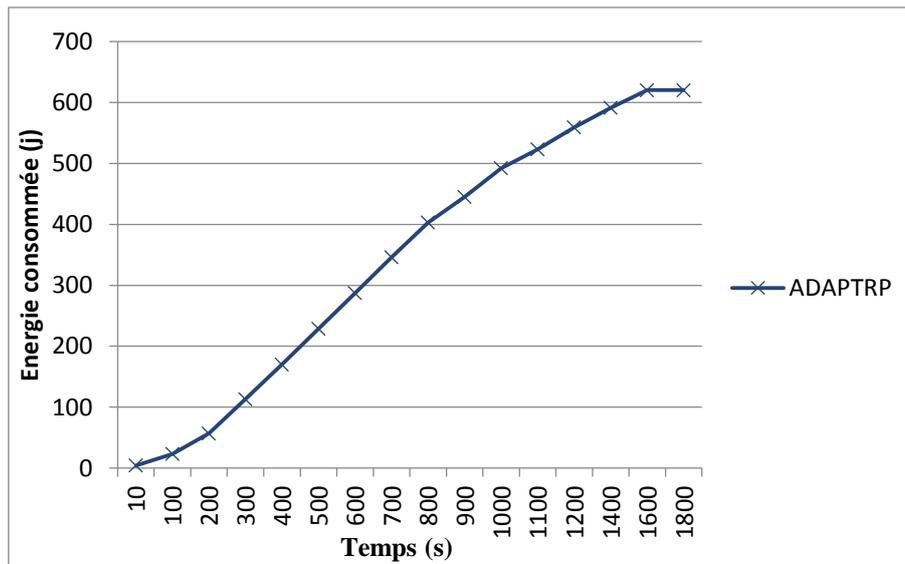


Figure 4.6 Quantité d'énergie consommée en fonction de la durée de la simulation

On remarque dans la figure 4.6 que la consommation d'énergie mise en évidence par le protocole ADAPTRP est faible car ce protocole utilise le concept d'optimisation basé sur la consommation d'énergie pour la sélection du meilleur protocole de routage dans chaque cluster.

❖ **Flux de données transmis (Panne SB)**

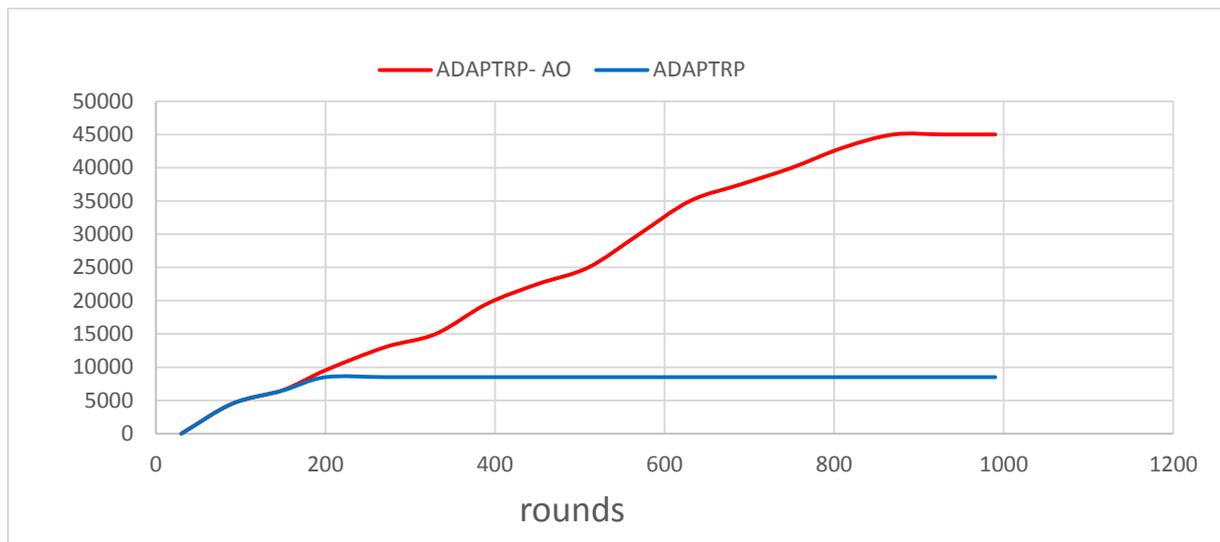


Figure 4.7 Quantité de données reçues (panne de la SB)

La figure 4.7 représente la quantité des données en réception dans le protocole ADAPTRP utilisant une seule station de base et le protocole ADAPTRP-AO avec deux stations de base en fonction du temps de simulation. On remarque que le nombre de paquets

envoyés par le protocole ADAPTRP-AO est nettement supérieur à celui envoyé par le protocole ADAPTRP. On observe que le nombre de données reçues est de 8475 depuis l'instant ($t = 200\text{ s}$) pour le protocole ADAPTRP suite à une panne de la station de base et que tous les paquets envoyés par les *CH* depuis cet instant sont perdus par contre dans le protocole ADAPTRP-AO proposé, on constate une augmentation dans le nombre de données reçues ; grâce à la deuxième *SB* (*SBS*) qui prendra le relais qui minimisera les paquets perdus.

❖ Flux de données transmis (Panne SB+CH+nœuds ordinaires)

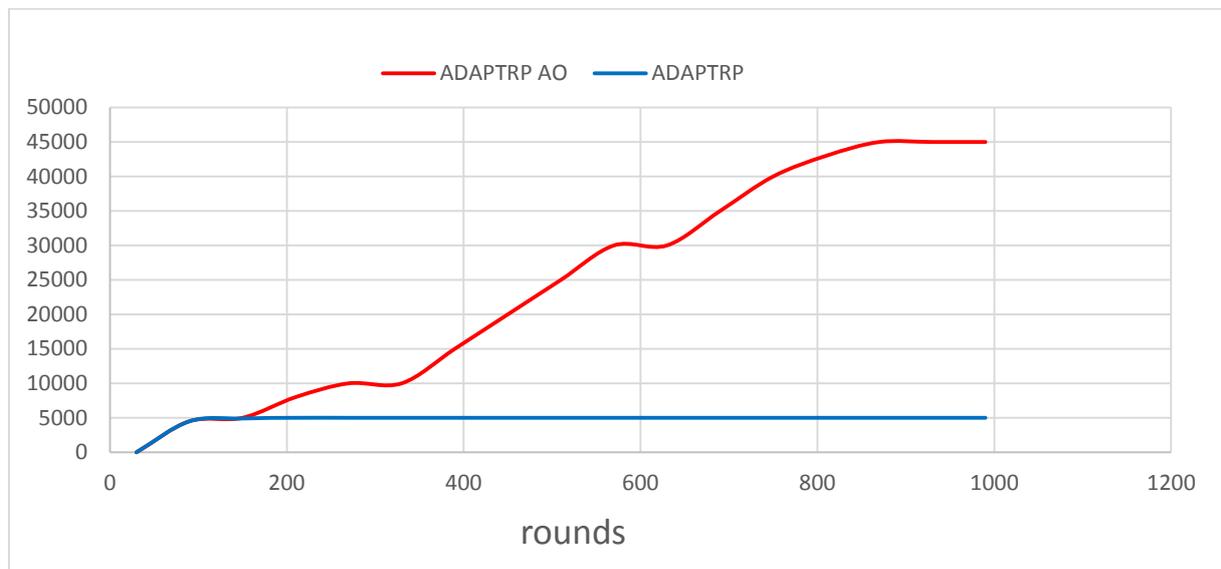


Figure 4.8 Quantité de données reçues (panne de la SB, des CHs et nœuds ordinaires)

Dans ce cas, on injecte les trois pannes au niveau de la *SB*, des *CH* et nœuds ordinaires en même temps et à l'instant $t = 200\text{ s}$.

Dans l'intervalle de temps $[0 \dots 200\text{ s}]$, on observe que le nombre de données reçues augmente dans ADAPTRP-AO par rapport au protocole ADAPTRP, à cause des vice-CH qui récupèrent les données et assurent le rôle des CH défaillants.

A l'instant ($t = 200\text{ s}$), la *SB* dans ADAPTRP et la *SB* principale dans ADAPTRP-AO sont toutes les deux en pannes, mais on remarque que le nombre des données reçues dans ADAPTRP est fixe depuis l'instant de panne, par contre dans ADAPTRP-AO ce nombre continue à croître grâce à la présence de la deuxième *SB* (*SBS*) qui assure la réception des données (figure 4.8).

❖ Nœuds défaillants détectés

Le premier objectif de l'auto-organisation dans les RCSF hétérogènes est le diagnostic qui relève de la détection et la localisation des nœuds défaillants. La figure 4.9 est dédiée à ce phénomène (le nombre des nœuds défaillants détectés en fonction du temps de simulation). Initialement, il n'y a aucun nœud défaillant mais après un certain temps de simulation : temps d'injection des fautes au niveau de quelques nœuds, on remarque que le protocole ADAPTRP ne détecte pas les nœuds défaillants mais seulement les nœuds morts par épuisement d'énergie. Par contre, dans ADAPTRP-AO, on remarque que la détection des nœuds défaillants est plus efficace, car la panne est détectée quel que soit son type (énergie, transmission, captage et traitement).

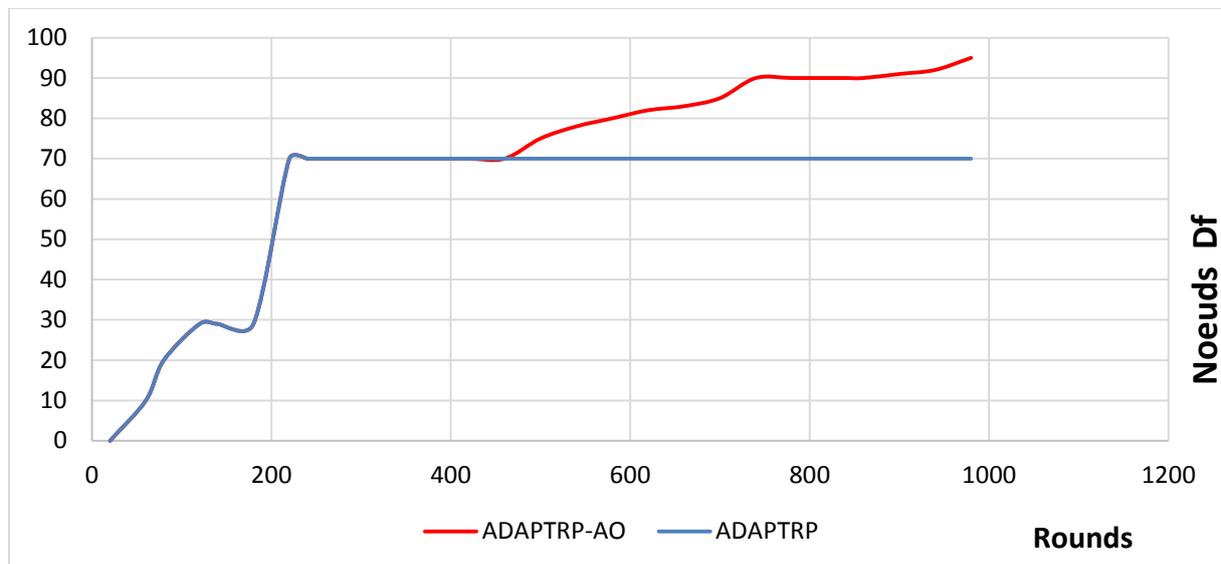


Figure 4.9 Nœuds défaillants détectés dans ADAPTRP et ADAPTRP-AO

Conclusion

Dans ce chapitre, nous avons présenté quelques environnements de simulation existants sur le marché. Nous avons également présenté NS2 en général avant de passer au choix de la plateforme OS et le choix du langage. Ensuite, nous avons présenté l'implémentation ainsi que l'évaluation de protocole ADAPTRP-AO. Le langage TCL, sous UBUNTU 10.10 et NS2, ont été utilisés lors de cette évaluation.

Nous avons également implémenté l'auto-organisation qui vise ADAPTRP-AO. En effet, une simple réinjection de données peut falsifier les résultats dans ADAPTRP. Cependant, ces défaillances sont détectées par le protocole ADAPTRP-AO grâce aux algorithmes d'auto-organisation qu'il offre.

Par ailleurs, nous avons constaté que les tests de performances effectués sur la consommation d'énergie, les nœuds défaillants détectés et les données transmises, ont montré que le protocole ADAPTRP-AO répond bien aux critères de performances souhaités. En effet, ce protocole permet de ne pas surcharger les nœuds capteurs, et de ne pas dégrader les performances du réseau.

Conclusion générale

Conclusion générale

Les réseaux de capteurs sans fil ne cessent d'envahir les marchés, vu leur nécessité dans les différents domaines d'application. Toutefois cette technologie émergente est dépendante d'une contrainte qui influence sa durée de vie ainsi que la durée de vie du réseau en entier, c'est la contrainte d'énergie. Pour cela les concepteurs des réseaux de capteurs se sont consacrés à la réalisation d'un tas de protocoles de routage hiérarchiques visant à minimiser l'énergie de consommation. Dans cette vision, ce type routage s'est avéré comme un aboutissement prometteur pour la conservation d'énergie et faciliter le routage de données des nœuds capteur vers la station de base.

Nous avons proposé une méthode pour l'auto-organisation des nœuds défaillants dans les RCSF, spécifiquement appliquée au protocole de routage ADAPTRP. Cette méthode consiste à gérer la défaillance dans le réseau après sa détection et sa localisation. Les résultats des simulations ont montré l'efficacité de notre proposition par rapport au protocole de référence.

Nous proposons en perspective comme suite à nos contributions pour améliorer les performances :

1. De considérer la méthode proposée avec une redondance adaptée aux capteurs de nœuds et éventuellement de mettre en œuvre de protocoles ADAPTRP-AO.
2. Auto-organiser le RCSF hétérogène en cas des pannes par l'introduction d'un mécanisme permettant la détection et la localisation de la panne par le cluster-head ou la station de base, ou par les nœuds ordinaires.

Références

Références

- [1] Développement d'une bibliothèque de capteurs, Rapport .25 /04/2008. Master, spécialité : Informatique professionnelle et Recherche en informatique, parcours : CASCAR.
- [2] Romdhani, Bilel. Exploitation de l'hétérogénéité des réseaux de capteurs et d'actionneurs dans la conception des protocoles d'auto-organisation et de routage. 2012. Thèse de doctorat
- [3] Lu, Jha-liang, Valois, Fabrice, et Barthel, Dominique. Low-energy self-organization scheme for wireless ad hoc sensor networks. In: 2007 Fourth Annual Conference on Wireless on Demand Network Systems and Services. IEEE, 2007. P. 138-145.
- [4] Akyildiz, Ian F., Su, Weilian, Sankarasubramaniam, Yogesh, et al. A survey on sensor networks. IEEE Communications magazine, 2002, vol. 40, no 8, p. 102-114.
- [5] B. Krishnamachari, D. Estrin, S. Wicker. "Modelling data-centric routing in wireless sensor networks", IEEE INFOCOM. 2002
- [6] M. Achir and L. Ouvry. "A routing protocol for wireless ad-hoc sensor networks: Multi-Path Source Routing Protocol (MPSR)", ICN'05: 4th International Conference on Networking (IEEE), Ile de la Réunion. Avril 2005.
- [7] N. Sadagopan, B. Krishnamachari, A. Helmy. "The ACQUIRE mechanism for efficient querying in sensor networks", Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska. Pp. 149-155. Mai 2003.
- [8] N. Mitton, "Auto-organisation dans les réseaux sans fil multi-sauts à grande échelle", Thèse de Doctorat en Informatique et Réseaux de l'insa de Lyon, INRIA Rhone Alpes, Lyon, France. Mars 2006.
- [9] T. Nieberg, S. Dulman, P. Havinga, L. V. Hoesel, J.Wu, "Collaborative algorithms for communication in wireless sensor networks ", Ambient Intelligence : Impact on Embedded Systems, Kluwer Academic Publishers. Octobre 2003.
- [10] D. Roth, J. Montavont, T. Noel, "MOBINET : gestion de la mobilité à travers différents réseaux de capteurs sans fil", Inria. 2010.
- [11] G. Yang, B. Tong, D. Qiao, W. Zhang, "Sensor-aided overlay deployment and relocation for vast-scale sensor networks", INFOCOM'08. 2008.
- [12] R. Kuntz, T. Machiavel, "Accessing the medium in mobile and dense wsn", In PIMRC'09. 2009.
- [13] M. Ali, S. K. Ravula, "Real-time support and energy efficiency in wireless sensor networks", Technical report, IDE0805. January 2008.
- [14] K. Akkaya, M. Younis, "A Survey on Routing Protocols for Wireless Sensor Networks", Journal of Ad Hoc Networks, Vol. 3 (3). Pp. 325-349. May 2005.

Références

- [15] A.P. Chandrakasan W. Heinzelman and H. Balakrishnan. An Application-Specific Proto- col Architecture for Wireless Microsensor Networks, In IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660-670. 2002.
- [16] S. Lindsey, C. Raghavendra, “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”, Proceedings of the IEEE Aerospace Conference. Vol. 3. Pp. 1125-1130. Big Sky, MT, USA. March 2002.
- [17] S. Lindsey, C.S. Raghavendra, K. Sivalingam, “Data gathering in sensor networks using the energy delay metric”, In Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing. April 2001.
- [18] O. Younis and S. Fahmy. Distributed clustering in ad-hoc sensor networks : A hybrid, energy- efficient approach. In Twenty-third annual joint Conference of the IEEE Computer and Com- munications Societies, volume 1, 2004.
- [19] S. Waware, N. Sarwade, and P. Gangurde. A review of power efficient hierarchical routing protocols in wireless sensor networks. International Journal of Engineering Research and Applications, 2(2) : pages 1096–1102, 2012.
- [20] X. Liu. A survey on clustering routing protocols in wireless sensor networks. Sensors, 12(8) : pages 11113–11153, 2012.
- [21] Boubiche, Djallel Eddine et Bilami, Azeddine. HEEP (Hybrid Energy Efficiency Protocol) based on chain clustering. International Journal of Sensor Networks, 2011, vol. 10, no 1-2, p. 25-35.
- [22] A. Manjeshwar and D. P. Agrawal, “TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks,” Proceedings of 15th IEEE International Parallel and Distributed Processing Symposium, San Francisco, 23-27 April 2001, pp. 2009-2015.
- [23] Manjeshwar, Arati et Agrawal, Dharma P. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In: ipdps. IEEE, 2002. P. 0195b.
- [24] Naidja, Miloud. Auto-adaptation dans les RCSF hétérogènes pour des e-applications. 2018. Thèse de doctorat.
- [25] Boumerzoug, Hayette. Gestion de la sécurité dans les réseaux de capteurs sans fil. 2011. Thèse de doctorat. Université du Québec à Trois-Rivières.
- [26] Galos, Mihai. Reconfiguration dynamique et simulation fine modélisée au niveau de transaction dans les réseaux de capteurs sans fil hétérogènes matériellement-logiciellement. 2012. Thèse de doctorat. Ecole Centrale de Lyon.

Références

- [27] Yarvis, Mark, Kushalnagar, Nandakishore, Singh, Harkirat, et al. Exploiting heterogeneity in sensor networks. In: INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. p. 878-890.
- [28] Boulanouar, Ibtissem. Algorithmes de suivi de cible mobile pour les réseaux de capteurs sans fils. 2014. Thèse de doctorat. Paris Est.
- [29] Gamez, Nadia, Fuentes, Lidia, et Aragüez, Miguel. Autonomic computing driven by feature models and architecture in FamiWare. *Software Architecture*, 2011, p. 164-179
- [30] Chouha, Adel. « Traitement et transfert d'images par réseau de capteurs sans fil », mémoire de magister, Université Hadj Lakhder – Batna, 2011.