



**HAL**  
open science

# Gestion de l'énergie et de la connectivité dans les réseaux de capteurs sans fil statiques et mobiles

Yacouba Ouattara

► **To cite this version:**

Yacouba Ouattara. Gestion de l'énergie et de la connectivité dans les réseaux de capteurs sans fil statiques et mobiles. Système multi-agents [cs.MA]. Université de Franche-Comté, 2015. Français. NNT : 2015BESA2046 . tel-01680050

**HAL Id: tel-01680050**

**<https://theses.hal.science/tel-01680050>**

Submitted on 10 Jan 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Gestion de l'énergie et de la connectivité dans les réseaux de capteurs sans fil statiques et mobiles

## THÈSE

présentée et soutenue publiquement le 16 Décembre 2015

pour l'obtention du grade de

**Docteur de l'université de Franche-Comté**  
(Spécialité Informatique)

par

**OUATTARA Yacouba**

### Composition du jury

*Président :* Laurent Phillippe, Professeur à l'Université de Franche-Comté

*Directeurs :* Hervé Guyennet, Professeur à l'Université de Franche-Comté  
Christophe Lang, Maître de Conférences à l'Université de Franche-Comté

*Rapporteurs :* Olivier Flauzac, Professeur à l'Université de Reims  
Pascal Lorenz, Professeur à l'Université de Haute Alsace

*Examineur :* Nicolas Marilleau, Ingénieur de Recherche à l'Institut de Recherche et Développement de Bondy

Mis en page avec la classe thloria.

# Remerciements

Tout d'abord, je tiens à remercier mon directeur de thèse, monsieur le professeur Hervé Guyennet et mon co-encadrant M. Christophe Lang pour leur patience, leur disponibilité, leurs idées et leurs conseils durant ces années de thèse.

Je tiens à remercier particulièrement messieurs les professeurs Pascal Lorenz et Olivier Flauzac pour avoir pris le temps de rapporter ce travail.

Mes plus sincères remerciements vont également à monsieur le professeur Laurent Philippe et à M. Nicolas Marilleau pour avoir accepté de participer au jury de cette thèse.

Ce travail a été réalisé au sein de l'équipe Cartoon du département DISC (Département Informatique des Systèmes Complexes) du laboratoire FEMTO-ST. Ainsi, je tiens à remercier tous les membres de l'équipe sans exception de peur d'en oublier un en les citant, et aussi les doctorants et tout le personnel du laboratoire de m'avoir offert un excellent cadre de travail.

Je remercie également messieurs les professeurs Stanislas Ouaro, Karifa Bayo, Bintou Sessouma et Sado Traoré pour leurs conseils et appuis multiformes pour la réalisation de cette thèse.

Un grand merci à ma famille qui m'a toujours soutenu et qui m'a donné la force et le courage nécessaires pour mener à bien cette thèse.

J'adresse aussi mes sincères remerciements à tous mes amis, qui m'ont soutenu de près ou de loin durant ces années de thèse.

Un grand merci à tous.



# Table des matières

<b>Table des figures</b>	<b>ix</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Introduction</b>	<b>1</b>
<b>Partie I Contexte et problématique</b>	<b>5</b>
<b>Chapitre 1 Les réseaux de capteurs</b>	<b>9</b>
1.1 Introduction . . . . .	10
1.2 Caractéristiques d'un réseau de capteurs . . . . .	11
1.2.1 Composants d'un capteur . . . . .	11
1.2.2 Architecture d'un nœud . . . . .	11
1.2.3 Architecture d'un réseau de capteurs . . . . .	14
1.3 Applications des réseaux de capteurs . . . . .	14
1.3.1 Le domaine militaire . . . . .	14
1.3.2 L'environnement . . . . .	14
1.3.3 Le bâtiment . . . . .	15
1.3.4 Le domaine de l'industrie . . . . .	15
1.3.5 Le domaine des transports . . . . .	15
1.3.6 Le domaine médical . . . . .	16
1.4 Système d'exploitation des réseaux de capteurs . . . . .	16
1.4.1 TinyOS . . . . .	16
1.4.2 Contiki . . . . .	16
1.5 Standards de communication des réseaux de capteurs . . . . .	17
1.5.1 Bluetooth . . . . .	18
1.5.2 Zigbee . . . . .	18

1.5.3	Infrarouge . . . . .	18
1.6	Contraintes des réseaux de capteurs . . . . .	18
1.6.1	Énergie . . . . .	18
1.6.2	Topologie dynamique . . . . .	19
1.6.3	Tolérance aux pannes . . . . .	19
1.6.4	Contraintes matérielles . . . . .	19
1.6.5	Passage à l'échelle . . . . .	19
1.6.6	Environnement . . . . .	20
1.6.7	Absence d'infrastructure . . . . .	20
1.7	Quelques axes de recherche dans les réseaux de capteurs . . . . .	20
1.7.1	Routage . . . . .	20
1.7.2	Couche MAC . . . . .	20
1.7.3	Qualité de service . . . . .	21
1.7.4	Diffusion de l'information . . . . .	21
1.7.5	Sécurité . . . . .	21
1.7.6	Mobilité . . . . .	21
1.7.7	Coopération entre la couche réseau et la couche Mac . . . . .	22
1.8	Conclusion . . . . .	22
<b>Chapitre 2 Gestion de l'énergie dans les réseaux de capteurs</b>		<b>23</b>
2.1	Introduction . . . . .	23
2.2	Consommation d'énergie dans les réseaux de capteurs . . . . .	24
2.2.1	Énergie de capture . . . . .	24
2.2.2	Énergie de traitements . . . . .	24
2.2.3	Énergie de communication . . . . .	24
2.2.4	État du module radio . . . . .	25
2.2.5	Accès au medium de transmission . . . . .	26
2.2.6	Modèle de propagation radio . . . . .	27
2.2.7	Formation des clusters . . . . .	28
2.3	Trois familles de protocoles de routage . . . . .	28
2.3.1	Routage à plat ou orienté données . . . . .	28
2.3.2	Routage hiérarchique . . . . .	29
2.3.3	Routage géographique . . . . .	30
2.4	Les techniques du clustering . . . . .	31
2.4.1	Les algorithmes et les limites . . . . .	32

---

2.4.2	Quelques protocoles de clustering . . . . .	34
2.5	Agrégation et compression . . . . .	37
2.6	Conclusion . . . . .	38

**Chapitre 3 La mobilité dans les réseaux de capteurs : Préserver l'énergie et la continuité du réseau en cas de mobilité** **39**

3.1	Introduction . . . . .	39
3.2	Avantages de la mobilité . . . . .	40
3.3	Types de mobilité . . . . .	41
3.3.1	La micro-mobilité . . . . .	41
3.3.2	La macro-mobilité . . . . .	44
3.4	Routage mobile . . . . .	46
3.4.1	Le routage pro actif . . . . .	46
3.4.2	Le routage réactif . . . . .	50
3.4.3	Le routage hybride . . . . .	52
3.4.4	L'adaptation à la métrique énergie . . . . .	57
3.5	Conclusion . . . . .	58

**Partie II Contribution** **61**

**Chapitre 4 Un algorithme de routage hybride (SHE) pour les réseaux de capteurs statiques** **65**

4.1	Motivation . . . . .	66
4.1.1	Choix des Cluster Heads (CH) . . . . .	66
4.1.2	Pannes et conséquences . . . . .	66
4.1.3	Position de la station de base . . . . .	67
4.2	Solution proposée et hypothèses . . . . .	67
4.3	Principes de notre algorithme . . . . .	67
4.4	Notre algorithme Hybride . . . . .	68
4.4.1	Les idées fondatrices . . . . .	68
4.4.2	Les algorithmes mis en œuvre . . . . .	70
4.5	Évaluation de l'algorithme hybride proposé . . . . .	73
4.5.1	Présentation de la plateforme matérielle . . . . .	74
4.5.2	Métriques . . . . .	75



4.5.3	Résultats . . . . .	75
4.5.4	Précisions et observations diverses . . . . .	83
4.6	Expérimentation avec la plateforme SensLAB . . . . .	83
4.6.1	Introduction . . . . .	83
4.6.2	Nœud SensLAB . . . . .	84
4.6.3	Expérimentation . . . . .	87
4.7	Conclusion . . . . .	90

**Chapitre 5 Adaptation de SHE aux réseaux de capteurs mobiles : utilisation de l’algorithme des chauves-souris** **91**

5.1	Topologie dynamique et mobilité de capteurs . . . . .	92
5.1.1	Exemples d’application . . . . .	92
5.1.2	Problématique . . . . .	94
5.2	Echolocalisation . . . . .	96
5.2.1	Quelques techniques de localisation . . . . .	96
5.2.2	Le principe de l’écho-localisation . . . . .	96
5.2.3	Algorithme d’optimisation par essaims de particules . . . . .	98
5.2.4	Algorithme des chauves souris . . . . .	98
5.3	SHEM : Sensor Hybrid Energy Mobility . . . . .	99
5.3.1	Description générale de la méthode . . . . .	99
5.3.2	Notre algorithme détaillé . . . . .	101
5.4	Simulation . . . . .	104
5.4.1	Déplacement . . . . .	104
5.4.2	Paramètres de simulation . . . . .	105
5.4.3	Mesures de performance . . . . .	105
5.4.4	Premier scénario . . . . .	106
5.4.5	Deuxième scénario . . . . .	110
5.4.6	Troisième scénario . . . . .	116
5.4.7	Bilan de la simulation . . . . .	116
5.5	Conclusion . . . . .	118

**Conclusion et Perspectives** **119**

**Bibliographie** **121**

**Résumé** **133**





# Table des figures

1.1	un réseau de capteurs . . . . .	10
1.2	Composants physiques d'un capteur . . . . .	11
1.3	Pile protocolaire . . . . .	12
1.4	Architecture d'un réseau de capteur . . . . .	14
1.5	Exemple de domaines d'application . . . . .	15
1.6	Quelques réseaux sans fil . . . . .	17
2.1	Schéma de consommation d'énergie d'un capteur . . . . .	25
2.2	Schéma de formule de calcul de la consommation d'énergie . . . . .	27
2.3	Algorithme de clustering . . . . .	31
2.4	Schéma de communication dans LEACH . . . . .	35
2.5	Architecture de communication PEGASIS . . . . .	36
3.1	Mouvements avec obstacles utilisant le diagramme de Voronoï . . . . .	46
3.2	Routage par inondation . . . . .	47
4.1	Illustration de la position de la station . . . . .	74
4.2	Les paramètres de simulation . . . . .	75
4.3	Exemple de division par zone (4 zones) à la position P1 . . . . .	76
4.4	Performance par zone à la position P0 . . . . .	76
4.5	Performance par zone à la position P1 . . . . .	77
4.6	Performance par zone à la position P2 . . . . .	77
4.7	Performance par zone à la position P3 . . . . .	78
4.8	Performance par zone en mode rectangulaire à la position P0 . . . . .	79
4.9	Performance par zone en mode rectangulaire à la position P1 . . . . .	79
4.10	Performance par zone en mode rectangulaire à la position P2 . . . . .	80
4.11	Performance par zone en mode rectangulaire à la position P3 . . . . .	80
4.12	Performance de l'algorithme hybride à P0 . . . . .	81
4.13	Performance de l'algorithme hybride à P1 . . . . .	82
4.14	Performance de l'algorithme hybride à P2 . . . . .	82
4.15	Performance de l'algorithme hybride à P3 . . . . .	83
4.16	Caractéristiques des sites sensLAB. . . . .	84
4.17	les quatre (4) sites SensLAB . . . . .	84
4.18	Architecture d'un noeud senslab . . . . .	85
4.19	site senslab de Lille . . . . .	85

4.20	Noeud WSN430 . . . . .	86
4.21	site senslab de Grenoble . . . . .	87
4.22	Evolution de la consommation de l'énergie du réseau en fonction du temps . . . . .	88
4.23	Energie consommée par un noeud . . . . .	88
4.24	Energie consommée par un noeud CH . . . . .	89
4.25	Energie consommée par un noeud CH . . . . .	89
5.1	Algorithme assurant la topologie dynamique . . . . .	100
5.2	Interface du simulateur . . . . .	104
5.3	Schéma de connectivité . . . . .	106
5.4	Schéma de la consommation d'énergie . . . . .	107
5.5	schéma de connectivité . . . . .	108
5.6	schéma d'énergie . . . . .	108
5.7	Schéma de connectivité à vitesse et distance variable . . . . .	109
5.8	Schéma de consommation d'énergie . . . . .	110
5.9	schéma de connectivité . . . . .	110
5.10	Schéma de consommation d'énergie . . . . .	111
5.11	Schéma de connectivité . . . . .	112
5.12	Schéma de consommation d'énergie . . . . .	112
5.13	Schéma de connectivité . . . . .	113
5.14	Schéma de consommation d'énergie . . . . .	113
5.15	Schéma de connectivité à vitesse progressive et distance variable . . . . .	114
5.16	Consommation d'énergie à vitesse progressive et distance variable . . . . .	114
5.17	Schéma de connectivité . . . . .	115
5.18	Consommation d'énergie . . . . .	115
5.19	Schéma de connectivité après injection de nouveaux noeuds . . . . .	116
5.20	Schéma de consommation d'énergie par injection de nouveaux noeuds . . . . .	117
5.21	Schéma de connectivité . . . . .	117
5.22	Schéma de consommation . . . . .	118

# Liste des tableaux



# Introduction

Un réseau ad hoc est un réseau sans fil qui fonctionne sans infrastructure définie au préalable. Chaque entité ou nœud communique directement avec son voisin en vue de transmettre un paquet d'un émetteur à un récepteur. Les entités doivent donc se situer les unes par rapport aux autres pour construire leurs tables de routage et sont limitées par la portée de leur communication radio. Lorsqu'un émetteur ne peut pas communiquer directement avec le destinataire, les informations sont transmises de proche en proche avant d'atteindre la destination souhaitée. Un protocole de routage doit prendre en compte les caractéristiques propres du réseau (variabilité de la topologie, absence d'infrastructures), l'hétérogénéité des nœuds (différences en termes de CPU, mémoire, batterie) et la bande passante limitée.

Un réseau de capteurs sans fil (RCSF) est un réseau ad hoc comportant généralement un grand nombre de nœuds qui sont des micro-capteurs autonomes. Ces nœuds capteurs sont des capteurs dit intelligents « smart sensors », accomplissant généralement trois tâches complémentaires : le relevé d'une grandeur physique, le traitement éventuel de cette information et la communication avec d'autres capteurs. Un réseau de capteurs peut être déployé pour la surveillance d'une zone géographique (feu de forêt, ennemis en guerre, catastrophe naturelle, etc).

Les nœuds capteurs travaillent de manière coopérative en partageant les informations recueillies entre eux pour une gestion efficace de leurs ressources et par rapport à un problème donné. Ces ressources sont la mémoire, l'énergie ou la puissance de transmission pour atteindre un ordinateur distant appelé station de base ou puits. Celui-ci est un nœud particulier doté d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie. Cette station sert d'interface ou de base de données pour prendre des décisions par rapport à un champ de captage ou à l'environnement. Elle peut être connectée à Internet ou posséder un lien radio de type GSM (Global System for Mobile communications) qui lui permet d'envoyer les informations (données ou alertes) à un centre de contrôle pour l'utilisateur final. On peut avoir plusieurs puits mobiles ou fixes dans un réseau de capteurs.

Les contraintes citées ci-dessus font l'objet de beaucoup de recherches scientifiques sur les réseaux de capteurs dans le but d'assurer une haute qualité de service de transfert, de stockage, de sécurité et une longue durée de vie du réseau.

Cette durée de vie du réseau est limitée par la durée de vie des batteries des capteurs. Il est souvent impossible de recharger ou de remplacer manuellement les batteries des nœuds



après leur épuisement, car les zones de déploiement sont souvent inaccessibles (champ de bataille, zones inhospitalières, etc). L'envoi des données ou la communication est la tâche qui consomme la plus grande partie de l'énergie des batteries par rapport à la capture, à l'écoute, au traitement et à l'état de veille. Une très faible quantité d'énergie disponible peut aussi engendrer une dégradation de la qualité d'information captée. Mieux encore, la forte densité du réseau permet une transmission rapide de l'information avec une dégradation de la qualité de service, car les relevés peuvent être supprimés avant que le nœud puisse les envoyer à destination à cause d'insuffisance de mémoire et de bande passante. Ces réseaux nécessitent donc des techniques originales de construction du réseau, économes en énergie de façon à satisfaire leurs capacités matérielles et les besoins de l'application. L'enjeu est donc d'étendre la durée de vie du système et la connectivité tout en maintenant la qualité de service en essayant d'anticiper la défaillance de certains nœuds. Les premiers protocoles proposés sont des protocoles de routage à plat adaptés au réseau ad hoc [IGE<sup>+</sup>03]. Cette approche considère que tous les nœuds du réseau sont dans le même niveau de hiérarchie et possèdent ainsi les mêmes rôles. Bien qu'efficace pour des réseaux de petite taille, ils ne permettent pas le passage à l'échelle quand le nombre de nœuds du réseau augmente.

Par la suite, des protocoles de routage hiérarchiques sont apparus [AKK04, IGE<sup>+</sup>03]. Cette approche consiste à découper le réseau en groupes d'entités appelés clusters en donnant au réseau une structure hiérarchique. Cette technique utilise des schémas de routage différents entre les clusters et au sein des clusters. Dans cette approche, chaque nœud maintient une connaissance pro active au sein de son cluster et seulement une connaissance partielle pour les autres clusters. Bien que ces protocoles permettent le passage à l'échelle, ils ne sont pas toujours adaptés aux changements de topologie liés, par exemple, aux contraintes extérieures au réseau.

Il faut donc développer de nouveaux protocoles pour améliorer la gestion de l'économie d'énergie en cas de mobilité des nœuds.

La mobilité est ainsi un aspect très important permettant l'extension du domaine d'application des réseaux de capteurs sans fil. Par exemple, des applications utilisent des capteurs mobiles pouvant se rapprocher de la cible ou pouvant être associés à des personnes, à des voitures ou à toute entité mobile. Actuellement, la plupart des travaux de recherche utilisent des réseaux statiques alors que de nombreuses applications peuvent obtenir de meilleurs résultats si la mobilité est prise en compte [BCM<sup>+</sup>08, JSB<sup>+</sup>06, SKJ<sup>+</sup>06]. On peut se trouver face à des cas où cette mobilité est subie par les capteurs et face à d'autres cas où les capteurs peuvent influencer sur cette mobilité pour améliorer la couverture du réseau.

L'objectif de cette thèse consiste donc à améliorer la gestion de l'énergie, tout en maintenant la connectivité, dans un réseau de capteurs statiques mais aussi mobiles.

Nous avons développé deux algorithmes permettant d'atteindre cet objectif. Dans le premier, nous nous sommes focalisés sur la consommation de l'énergie au niveau des différents capteurs pour permettre de limiter la dépense dans la construction des clusters et dans le choix des Cluster Heads (CH). Nous partons de cette contribution, dans le deuxième algorithme, et de la technique de l'écho-localisation pour prendre en compte la mobilité des capteurs, mobilité subie ou provoquée.

Cette thèse se compose de deux parties. Une première partie composée de trois cha-

---

pitres décrit les réseaux de capteurs et propose un état de l'art sur la consommation d'énergie dans les réseaux de capteurs.

Ainsi dans le chapitre 1, nous présentons les réseaux de capteurs dans leur généralité. Ce chapitre décrit les éléments constitutifs d'un réseau de capteurs et les domaines d'applications.

Dans le chapitre 2, nous abordons les différentes approches proposées dans la littérature pour la structuration d'un réseau en clusters. Cette étude révélera les limites des approches actuelles et nous servira de support pour positionner nos travaux de recherche. Nous donnons un état de l'art sur le clustering nous permettant d'exposer les différentes techniques de réduction de la consommation d'énergie dans un réseau de capteurs.

Nous présentons dans le chapitre 3 la mobilité dans les réseaux ainsi que le problème de la consommation d'énergie dans ceux-ci. On définit également les techniques récentes pour réduire l'énergie consommée dans un réseau de capteurs mobiles.

La deuxième partie de la thèse présente notre contribution développée dans les chapitres 4 et 5. Ils décrivent notre proposition dans la gestion de l'énergie et de la connectivité dans les réseaux de capteurs sans fil.

Plus précisément, le chapitre 4 présente un algorithme hybride de formations des clusters et les moyens de les diagnostiquer pour une meilleure connectivité et une anticipation des défaillances.

Le dernier chapitre décrit les techniques employées pour réduire la consommation d'énergie dans un réseau de capteurs mobiles. Ce chapitre s'appuie sur le comportement des chauves souris pour proposer un algorithme permettant d'améliorer la consommation d'énergie. Il utilise les propositions développées dans le chapitre 4.

Nous effectuons, pour ces deux chapitres, des expérimentations par simulation permettant de tester divers scénarii propres à éprouver nos propositions.



Première partie

Contexte et problématique



Cette première partie de ma thèse fixe les enjeux des problèmes de l'énergie dans les réseaux de capteurs. Elle expose dans un premier temps les caractéristiques de ce type de réseaux. Puis, elle donne un état de l'art sur la consommation d'énergie et les propositions d'amélioration basées sur le routage et le clustering développées dans la littérature. Ces propositions consistent en un regroupement et un fonctionnement distribué des capteurs d'un réseau pour un but précis. Le plus souvent, ce réseau comporte des capteurs ayant les mêmes caractéristiques et avec des capteurs spécifiques jouant le rôle de CH et une station de base. Cette partie se termine sur un examen de la littérature sur la mobilité dans les réseaux de capteurs.



# Chapitre 1

## Les réseaux de capteurs

### Sommaire

---

<b>1.1</b>	<b>Introduction</b>	<b>10</b>
<b>1.2</b>	<b>Caractéristiques d'un réseau de capteurs</b>	<b>11</b>
1.2.1	Composants d'un capteur	11
1.2.2	Architecture d'un nœud	11
1.2.3	Architecture d'un réseau de capteurs	14
<b>1.3</b>	<b>Applications des réseaux de capteurs</b>	<b>14</b>
1.3.1	Le domaine militaire	14
1.3.2	L'environnement	14
1.3.3	Le bâtiment	15
1.3.4	Le domaine de l'industrie	15
1.3.5	Le domaine des transports	15
1.3.6	Le domaine médical	16
<b>1.4</b>	<b>Système d'exploitation des réseaux de capteurs</b>	<b>16</b>
1.4.1	TinyOS	16
1.4.2	Contiki	16
<b>1.5</b>	<b>Standards de communication des réseaux de capteurs</b>	<b>17</b>
1.5.1	Bluetooth	18
1.5.2	Zigbee	18
1.5.3	Infrarouge	18
<b>1.6</b>	<b>Contraintes des réseaux de capteurs</b>	<b>18</b>
1.6.1	Énergie	18
1.6.2	Topologie dynamique	19
1.6.3	Tolérance aux pannes	19
1.6.4	Contraintes matérielles	19
1.6.5	Passage à l'échelle	19
1.6.6	Environnement	20
1.6.7	Absence d'infrastructure	20
<b>1.7</b>	<b>Quelques axes de recherche dans les réseaux de capteurs</b>	<b>20</b>



1.7.1	Routage . . . . .	20
1.7.2	Couche MAC . . . . .	20
1.7.3	Qualité de service . . . . .	21
1.7.4	Diffusion de l'information . . . . .	21
1.7.5	Sécurité . . . . .	21
1.7.6	Mobilité . . . . .	21
1.7.7	Coopération entre la couche réseau et la couche Mac . . . . .	22
<b>1.8</b>	<b>Conclusion . . . . .</b>	<b>22</b>

---

## 1.1 Introduction

Les réseaux de capteurs sans fil (RCSF) apportent de nombreuses solutions en matière de détection et de surveillance de notre environnement. Un RCSF est généralement constitué de beaucoup de nœuds qui stockent, traitent et acheminent des données mesurées en utilisant les technologies sans fil. Par nature, ces nœuds se caractérisent par des ressources limitées en mémoire, en énergie, en puissance d'émission et de calcul et ayant des caractéristiques différentes des ordinateurs (figure 1.1).

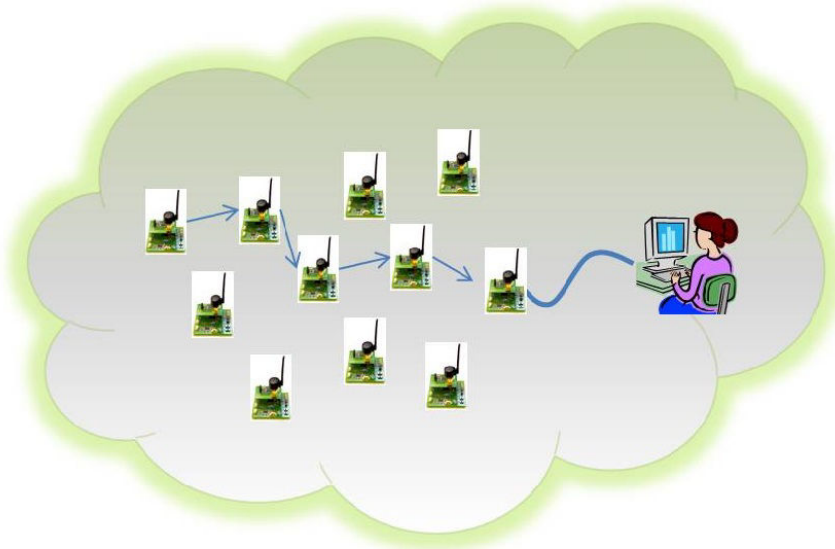


FIGURE 1.1 un réseau de capteurs

Dans ce type de réseaux, l'énergie est la contrainte principale car elle conditionne les services rendus par chaque nœud. Sachant que la transmission est le service qui consomme le plus d'énergie, un nœud de RCSF doit limiter la quantité de données transmises autant que possible afin d'augmenter sa durée de vie. Ainsi dans ce chapitre, nous traiterons successivement les points suivants :

les caractéristiques d'un réseau de capteurs ;  
 les applications de réseaux de capteurs ;  
 les systèmes d'exploitation de réseaux de capteurs ;  
 les standards de communication des réseaux de capteurs ;  
 les contraintes de réseaux de capteurs ;  
 les axes de recherche dans les réseaux de capteurs.

## 1.2 Caractéristiques d'un réseau de capteurs

Cette partie présente les éléments constitutifs d'un capteur et d'un réseau de capteurs.

### 1.2.1 Composants d'un capteur

Comme le montre la figure 1.2, un capteur est composé de quatre éléments principaux [ASSC02b, ACDFP09, RSPS02] qui sont : le microprocesseur pour le traitement, la radio pour la communication du nœud dans un réseau, l'unité de mesure jouant le rôle d'acquisition et de convertisseur, la batterie ou la pile qui alimente l'ensemble.

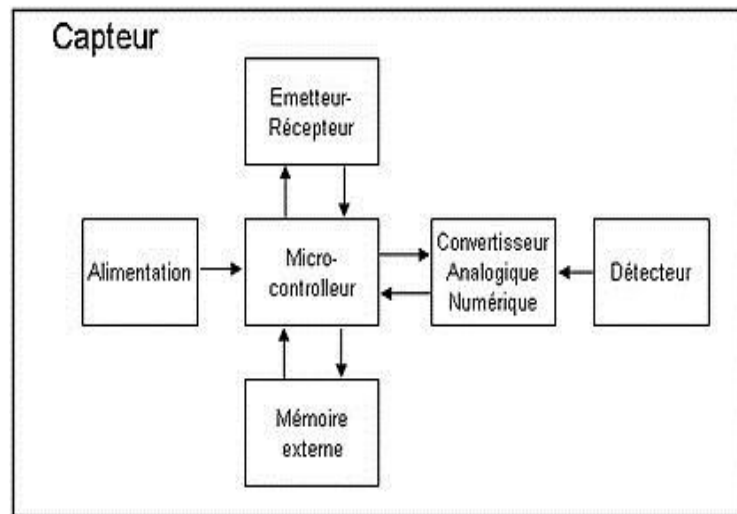


FIGURE 1.2 Composants physiques d'un capteur

Un capteur peut être doté d'autres composants additionnels tels que GPS (Global Position System) pour son déplacement [WB10].

### 1.2.2 Architecture d'un nœud

Les articles [ASSC02a, CTDC06] montrent la pile protocolaire de communication dans un réseau de capteurs. Le rôle de ce modèle consiste à standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Le modèle de communication est composé

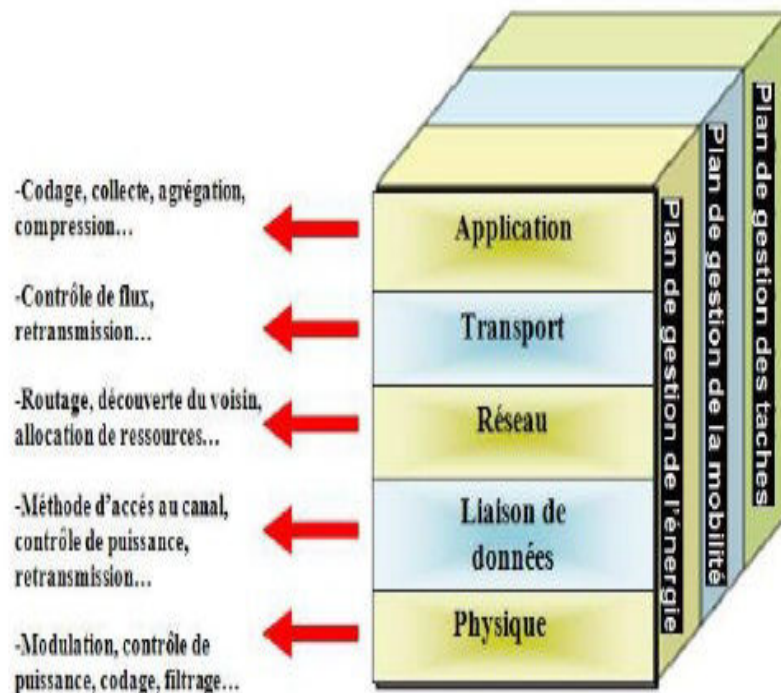


FIGURE 1.3 Pile protocolaire

de cinq couches (figure 1.3). Le nœud utilise aussi trois dispositifs pour la gestion de l'énergie, la gestion de la mobilité et la gestion des tâches [ASSC02a].

### La couche application

Cette couche assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels. La couche applicative doit permettre à l'utilisateur d'interroger le réseau, par exemple sur la valeur d'une certaine variable physique, ou encore l'endroit spécifique où une certaine variable satisfait à un critère donné. Elle permet aussi de programmer les algorithmes de routage, d'agrégation, de compression,... [ASSC02a].

### La couche transport

Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission. Elle assure donc l'acheminement de bout en bout des messages, dans un environnement peu fiable à cause des conditions du canal radio et de l'instabilité de la topologie du réseau, qui peuvent provoquer la perte de messages [ASSC02b].

## La couche réseau

Dans la couche réseau le but principal est de trouver une route et une transmission fiable des données captées par des nœuds capteurs vers le puits tout en optimisant l'utilisation de l'énergie des capteurs. Le routage des données a une importance capitale dans les RCSF et il soulève beaucoup de défis à cause de plusieurs contraintes intrinsèques liées aux RCSF comme l'absence d'adresse ou d'identifiant propre à chaque nœud, les changements topologiques fréquents (disparition d'un nœud ou sa non-participation au routage, etc.) auxquels tout protocole de routage doit s'adapter efficacement. Il existe principalement trois familles de protocoles de routage [ASSC02b, ASSC02a, AKK04] : les protocoles orientés données non hiérarchiques, les protocoles hiérarchiques et les protocoles géographiques. Nous développerons ces trois familles au niveau du chapitre 2.

## La couche liaison

Cette couche spécifie comment les données sont expédiées entre deux nœuds routeurs dans une distance à un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès au média. Elle assure la liaison point à point et multipoint dans un réseau de communication. En effet, dans les RCSF, les protocoles de la couche MAC doivent être conçus de manière à réduire les retransmissions ainsi que les temps d'écoute active pour réduire l'énergie consommée par les capteurs [ASSC02a, ACDFP09]. Aussi, les priorités d'accès au canal doivent être attribuées selon la nature du flux et selon l'application car, dans un RCSF, les nœuds sont coopératifs plutôt que compétitifs.

## La couche physique

La couche physique est responsable de la sélection de fréquence, de la génération de l'onde porteuse, de la détection du signal, de la modulation et du cryptage des données.

## La gestion des tâches

Cette opération est réalisée pour équilibrer et contrôler les tâches dans un domaine, tous les nœuds ne participent pas à l'exécution d'une tâche au même moment, mais selon la disponibilité des nœuds au niveau de l'énergie.

## La gestion de la mobilité

Cette tâche permet de détecter et d'enregistrer les mouvements des nœuds. Elle permet d'avoir une trace des voisins et dispose d'un chemin vers le récepteur à tout moment. Il est surtout utilisé par les techniques de routage pour la localisation [ASSC02a].

## La gestion de l'énergie

Cette tâche permet de spécifier comment l'énergie va être utilisée. Elle permet par exemple d'éteindre le récepteur lorsqu'il est à un état idle. Lorsque le niveau d'énergie est bas, elle permet d'envoyer des broadcast à ses voisins pour les informer qu'ils ne peuvent plus participer au routage [ASSC02a].

### 1.2.3 Architecture d'un réseau de capteurs

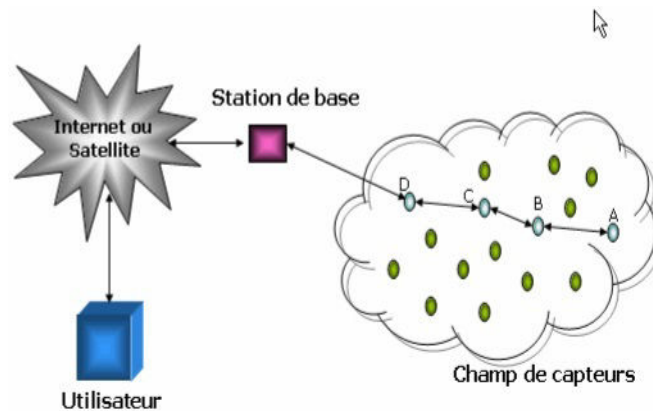


FIGURE 1.4 Architecture d'un réseau de capteur

Tous les réseaux de capteurs respectent globalement la même architecture basée sur un noyau central (figure 1.4) autour duquel s'articulent les différentes interfaces d'entrée sortie, de communication et d'alimentation. Ces capteurs sont reliés à une ou plusieurs passerelles (sink, puits ou station de base) qui permettent l'interconnexion avec d'autres réseaux (Internet par exemple) et la récupération des données pour l'analyse et la prise de décision[ASSC02a].

## 1.3 Applications des réseaux de capteurs

Les réseaux de capteurs sans fil permettent de capturer et traiter de l'information distribuée. Les domaines d'application sont nombreux et variés : détection des incendies, sécurité, agriculture, environnement, gestion des espaces urbains et des transports par exemple.

### 1.3.1 Le domaine militaire

Dans le domaine militaire, un réseau de capteurs peut être rapidement déployé dans un environnement pour la détection et la collecte d'informations sur la position de l'ennemi, pour la surveillance des zones contaminées, pour la détection d'agents chimiques, bactériologiques sur un terrain [CK03].

### 1.3.2 L'environnement

Des capteurs dispersés peuvent servir à détecter les feux de forêt, à contrôler la qualité de l'eau et de l'air. Par exemple dans les champs agricoles, les capteurs peuvent être semés avec les graines pour une agriculture de précision. Ainsi, les zones sèches seront facilement identifiées et leur irrigation sera donc plus efficace [CTDC06, ASSC02a].

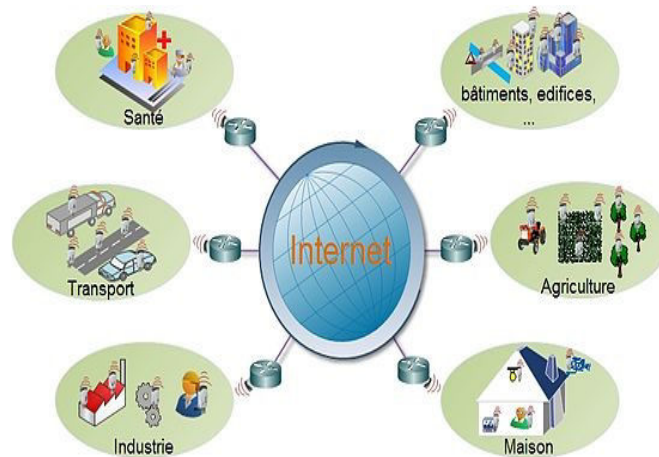


FIGURE 1.5 Exemple de domaines d'application

### 1.3.3 Le bâtiment

Un bâtiment avec un réseau de capteurs permet d'optimiser les consommations d'énergies d'une entreprise sans diminuer le confort des occupants. Le gestionnaire d'énergie dispose en effet d'une vision précise des différentes consommations d'énergie. Il est alerté en temps réel de tout dysfonctionnement (La surconsommation d'électricité, la fuite d'eau) et peut alors prendre les dispositions nécessaires. Un bâtiment communicant permet ainsi de réduire la consommation et donc la facture énergétique de l'entreprise de façon significative [MCP<sup>+</sup>02].

### 1.3.4 Le domaine de l'industrie

Les nœuds capteurs peuvent être intégrés au processus de stockage et de livraison. Ce réseau ainsi créé, pourra être utilisé pour connaître la position, la direction et l'état d'un paquet ou d'une cargaison. Ainsi, le client aura l'information en temps réel sur son paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettent de suivre le procédé de production à partir des matières premières jusqu'au produit final. Grâce aux réseaux de capteurs, les entreprises peuvent offrir une meilleure qualité de service tout en réduisant leurs coûts. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué pourra aussi maintenir une température homogène dans les pièces. [BCSV06]

### 1.3.5 Le domaine des transports

Aujourd'hui, il apparaît indispensable de maîtriser l'usage de la voiture en ville afin de réduire la pollution et d'améliorer la qualité de vie. Grâce à un réseau de capteurs, les automobilistes pourront connaître en temps réel les places disponibles dans les parkings. Ce système, en réduisant le temps consacré à la recherche d'une place, améliorera la

circulation, notamment celle des transports collectifs, aidera à une bonne gestion des aires de livraison et réduira les émissions de gaz carbonique associées au trafic urbain. Dans le réseau de capteurs routiers, il faudra voir la rapidité, la fluidité des transports et l'économie d'énergie. Ce réseau permettra de réduire la mortalité sur les routes. [CCV04]

### 1.3.6 Le domaine médical

On peut imaginer que dans un futur proche, la surveillance des fonctions vitales de l'être humain soit possible grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Actuellement, des micro-caméras qui peuvent être avalés existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de quelques heures. L'utilisation des réseaux de capteurs dans le domaine de la médecine pourrait apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le contrôle du rythme cardiaque et le diagnostic des patients [LMFJ<sup>+</sup>04].

## 1.4 Système d'exploitation des réseaux de capteurs

Un système d'exploitation pour capteurs est nécessaire afin de faciliter la programmation en fournissant des abstractions des ressources physiques. Un SE doit fournir des facilités de haut niveau pour l'accès aux composants de communication, de captage, de la gestion d'énergie et du stockage. Ainsi les développeurs d'applications sont déchargés de la programmation au niveau plus bas. De nombreux systèmes d'exploitation spécialisés existent, parmi lesquels on peut citer TinyOs, Contiki, Mantis, Nano-RK, et LiteOS [LMP<sup>+</sup>05, BCD<sup>+</sup>05].

### 1.4.1 TinyOS

TinyOS est un système d'exploitation intégré, modulaire. Cette plate-forme logicielle ouverte, ainsi qu'une série d'outils, développés à la base par l'Université de Berkeley sont enrichis par une multitude d'utilisateurs. En effet, TinyOS est le plus répandu des OS pour les réseaux de capteurs sans-fil. TinyOS4 a une empreinte mémoire de 4Ko. La bibliothèque de composants TinyOS comprend des protocoles réseau, des applications de services distribués, des drivers de capteurs et des outils d'acquisition de données. La contrainte énergétique due à l'autonomie des capteurs implique l'utilisation de puissances de calcul réduites entraînant le développement de logiciels contraint par la capacité de la mémoire et par la vitesse d'exécution. L'utilisation du langage nesC permet d'optimiser le code et ainsi réduire l'usage de la RAM.

### 1.4.2 Contiki

Contiki est un système d'exploitation léger et flexible pour capteurs miniatures en réseau. Ces dernières années, le monde scientifique a porté une attention importante aux réseaux de capteurs sans fil. La miniaturisation des capteurs et leur prix relativement

faible, permettent d’imaginer des applications très variées dans les domaines scientifiques, militaires, industriels et domotiques. Pour faciliter le développement de ces applications, une équipe du centre suédois de recherche scientifique SICS a créé Contiki. Destiné à être embarqué dans des capteurs miniatures ne disposant généralement que de ressources limitées, Contiki propose malgré tout les principales caractéristiques et fonctionnalités d’un système d’exploitation tout en favorisant une consommation énergétique et une empreinte mémoire minimales. Ses principaux atouts sont le support des protocoles IPv6 et 6LoW-PAN, sa flexibilité et sa portabilité. Disponible gratuitement sous licence BSD, Contiki peut être utilisé et modifié, même à des fins commerciales.

## 1.5 Standards de communication des réseaux de capteurs

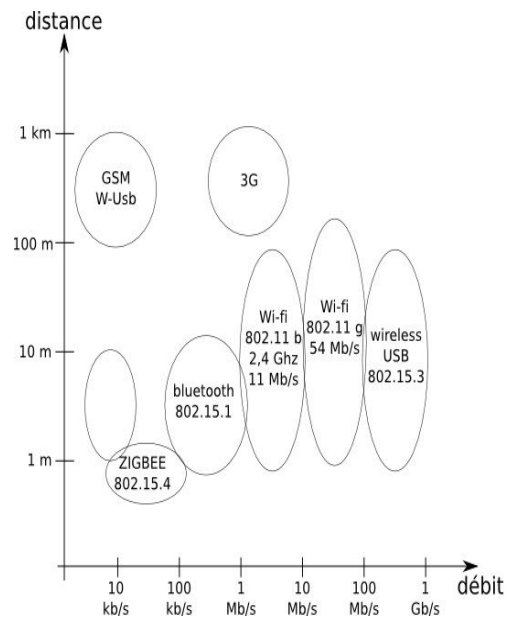


FIGURE 1.6 Quelques réseaux sans fil

Il y a une multitude de normes sans fil (figure 1.6) comme le Wi-Fi (le standard IEEE 802.11) [All04] et le WiMax (le standard IEEE 802.16) [All04], qui s’adressent au transport des données à haut débit. Certains dispositifs comme les capteurs n’ont pas besoin d’une largeur de bande très élevée, mais plutôt d’un temps de latence faible ainsi qu’une consommation d’énergie très basse, pour une longue durée de vie sur batterie et un grand nombre de dispositifs. D’où la nécessité de concevoir d’autres normes sans fil capables de répondre à ces exigences. Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs sans fil se trouvent les standards Bluetooth, ZigBee et infrarouge présentés ci-après.



### 1.5.1 Bluetooth

Connu sous le nom de la norme IEEE 802.15.1, le Bluetooth [All04] a été lancé par Ericsson en 1994, proposant un débit théorique de 1 Mbps lui permettant une transmissions de la voix, des données et des images, d'une portée maximale d'une trentaine de mètres. C'est un standard de communication sans fil utilisant une technologie radio courte distance destinée à simplifier les connexions entre les appareils électroniques. L'avantage du Bluetooth par rapport au Wi-Fi, qui offre apparemment de meilleures performances, se trouve au niveau de la consommation : en effet, la connexion sans fil est surtout utilisée pour les appareils mobiles, et donc n'étant pas directement reliés au secteur. Par conséquent, la technologie Bluetooth a une plus faible consommation que le Wi-Fi. Cependant, cette faible consommation d'énergie ne répond pas encore à des exigences des capteurs. En plus de cette consommation d'énergie, la technologie Bluetooth souffre d'un problème de passage à l'échelle.

### 1.5.2 Zigbee

Beaucoup moins connu que Bluetooth, ZigBee est le nom d'une suite de protocoles de hauts niveaux basée sur le standard IEEE 802.15.4 [All04, Ega05] pour les réseaux personnels sans fil (Wireless Personal Area Network : WPAN). Fonctionnant depuis 2000, la technologie ZigBee a pour but la communication à courte distance telle que le propose déjà la technologie Bluetooth. Sa très faible consommation électrique et ses coûts de production très bas en font un élément acceptable pour les RSCFs malgré son faible débit théorique (250Kbps théorique contre 1Mbps pour Bluetooth) car la taille des paquets n'est pas trop importante.

### 1.5.3 Infrarouge

Cette technique permet de créer des liaisons sans fil de quelques mètres avec des débits pouvant monter à quelques mégabits par seconde. Cette technologie est largement utilisée dans la domotique (télécommandes), elle souffre toutefois des perturbations dues aux interférences lumineuses.

## 1.6 Contraintes des réseaux de capteurs

La conception et la réalisation des réseaux de capteurs sans fil sont influencées par plusieurs paramètres. Ces facteurs servent comme directives pour le développement des algorithmes et protocoles utilisés dans les RSCF.[AV10]. Ce sont :

### 1.6.1 Énergie

Les nœuds capteurs embarquent en général une quantité d'énergie stockée dans une pile et très limitée. Les transmissions de données entre les nœuds consomment beaucoup d'énergie. En général déployé dans des environnements hostiles, la vie d'un réseau de capteurs se résume à la durée de vie des batteries. Le facteur énergie est donc au centre

de toutes les préoccupations sur les capteurs : protocoles de routage, technologie sans fil adaptée, redondance des équipements. Il faut minimiser les dépenses énergétiques ; car l'énergie est une contrainte clé dans les réseaux de capteurs [FJWW13, CW06].

### 1.6.2 Topologie dynamique

La défaillance d'un nœud capteur ou l'expiration de l'énergie d'un capteur peut changer la topologie des réseaux de capteurs au cours du temps. La mobilité des nœuds capteurs et des stations de bases est aussi un facteur du changement de topologie pour des réseaux de capteurs au cours de l'évolution de vie du réseau. Par exemple, on peut imaginer un réseau de capteurs pour surveiller le déplacement d'animaux. Il faut donc s'attendre à ce que la topologie du réseau soit en permanente modification. Cela signifie qu'il faut trouver des protocoles de routage efficaces tout en économisant l'énergie pour assurer une longue vie au réseau [ASFI13, LLV13, BMJ<sup>+</sup>98].

### 1.6.3 Tolérance aux pannes

Les nœuds peuvent être sujets à des pannes dues à leur fabrication. Il peut donc y avoir des capteurs défectueux. Les interactions externes peuvent aussi être la cause des dysfonctionnements. Afin que les pannes n'affectent pas la tâche première du réseau, il faut évaluer la capacité du réseau à fonctionner sans interruption. Les algorithmes et protocoles doivent tenir compte du fait qu'un nœud peut cesser de fonctionner par manque d'énergie ou parce qu'il a été détruit. Ils devront adapter leur niveau de tolérance aux pannes en fonction de l'hostilité du milieu dans lequel est déployé le réseau [DHM04, CCL03, HSA00, KPSV02].

### 1.6.4 Contraintes matérielles

La principale contrainte matérielle est la taille du capteur. Les autres contraintes sont la consommation d'énergie qui doit être moindre pour que le réseau survive le plus longtemps possible. Le capteur doit s'adapter aux différents environnements tels que les fortes chaleurs par exemple surtout s'il est déployé dans des environnements hostiles [PK00].

### 1.6.5 Passage à l'échelle

Le nombre de nœuds de capteurs augmente sur un réseau sans fil et ce nombre peut atteindre le million. Un nombre aussi important de nœuds engendre beaucoup de transmissions entre les nœuds et peut imposer des difficultés pour le transfert de données. Les protocoles et algorithmes devraient fonctionner parfaitement [DLH<sup>+</sup>14, ZK03, PK00, EGHK99].

### 1.6.6 Environnement

Les capteurs sont souvent déployés en masse dans des endroits tels que les champs de bataille, à l'intérieur de grandes machines, au fond d'un océan, dans les champs biologiquement ou chimiquement souillés,... Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiquement éloignées [PK00].

### 1.6.7 Absence d'infrastructure

Les réseaux Ad-hoc en général, et les réseaux de capteurs en particulier se distinguent des autres réseaux par la propriété d'absence d'infrastructure préexistante et de tout genre [PK00].

## 1.7 Quelques axes de recherche dans les réseaux de capteurs

Les recherches dans le domaine des réseaux de capteurs se concentrent sur plusieurs problématiques. Parmi ces problématiques, nous retenons :

### 1.7.1 Routage

Les protocoles de routage spécifient la communication entre les nœuds de capteurs, en diffusant les informations, autorisant la construction de routes entre nœuds et ils permettent aux différents types de flux d'être livrés et fusionnés pendant la transmission afin de limiter la quantité d'informations de contrôle échangée. Beaucoup de protocoles sont adaptés aux capteurs fixes. Concevoir un protocole de routage mobile et performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets est un problème crucial des réseaux de capteurs mobiles [LGF10].

### 1.7.2 Couche MAC

Un défi important dans les réseaux sans fil est la gestion des collisions dues à un transfert de données simultanées entre deux nœuds sur le même support. Les protocoles MAC [HXS<sup>+</sup>13, LGF07, CRL13] ont été développés essentiellement pour essayer d'éviter ce genre de collisions en aidant les nœuds à décider quand et comment ils peuvent accéder au support. Mais, la spécificité des réseaux de capteurs sans fil mobiles nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux pour améliorer le débit, optimiser le partage du médium afin de minimiser le délai de délivrance des paquets d'où, la minimisation de la consommation d'énergie.

### 1.7.3 Qualité de service

Les grandes quantités de données issues des capteurs, à fréquences et positions variables dans le temps, acquises dans des environnements hostiles avec une capacité d'énergie et de communication limitée, rendent les données plus imprécises et incertaines. Ceci pose le problème de la qualité des données issues des capteurs.

L'amélioration de la qualité de transfert de données pour un réseau de capteurs mobiles en proposant une nouvelle architecture de routage et des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, de limiter les pertes de paquets pour la gestion du réseau [HN15, SS12].

### 1.7.4 Diffusion de l'information

Les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes imposées [DQW14]. Dans une telle communication, un message est envoyé depuis un objet donné vers tous les autres du réseau. Ainsi, pour concevoir un protocole efficace, il faudrait assurer une couverture maximale des capteurs composant le réseau et minimiser le nombre des ré-émissions, les réceptions redondantes, ou en limitant la puissance d'émission. La résolution d'un tel problème permettra une diminution de la consommation d'énergie.

### 1.7.5 Sécurité

La sécurité [AALK12, MG10] est une nécessité pour la majorité des applications qui utilisent les réseaux de capteurs sans fil, notamment si les nœuds capteurs sont déployés dans des endroits peu sûrs, tels que les champs de bataille, les lieux stratégiques (aéroports, bâtiments critiques, etc.). Les nœuds capteurs qui opèrent dans des lieux difficiles d'accès, sans protection et sans possibilité de rechargement de batterie, peuvent être soumis à des actions malveillantes susceptibles de compromettre l'essence même du réseau de capteurs sans fil. C'est pourquoi, il est primordial de pouvoir leur assurer un niveau de sécurité acceptable. Compte tenu de leurs spécificités contraignantes, la sécurité dans ce type de réseaux relève d'un véritable challenge surtout que les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées que présentent ces réseaux.

### 1.7.6 Mobilité

A l'heure actuelle, les réseaux de capteurs sans fil ne sont plus constitués de capteurs fixes uniquement mais aussi de capteurs placés sur des éléments mobiles comme les véhicules, les animaux, etc. Ces nœuds mobiles peuvent être utilisés dans des applications telles que la surveillance animale, le contrôle de trafic, etc. L'intérêt de la mobilité dans les WSN est multiple dans la mesure où les capteurs mobiles peuvent notamment permettre d'étendre la couverture d'un réseau, d'améliorer ses performances de routage ou sa connectivité globale [DRS<sup>+</sup>05]. Cette mobilité peut se traduire par des difficultés à émettre

des données. En effet, le nœud mobile doit s'insérer dans les communications. Cette insertion peut être particulièrement complexe surtout si les communications entre les nœuds fixes sont organisées de manière précise. Peu de protocoles d'accès au médium prennent en compte la dynamique des capteurs. Un problème important est le type de contrôle de la mobilité des nœuds qu'intègre la conception du réseau [Ibn08]. Les protocoles à intervalles de temps intègrent difficilement les capteurs mobiles dans leurs algorithmes d'ordonnement et de communication. Il faut alors de nouveaux protocoles qui prennent en compte la mobilité des capteurs [WWT12] pour minimiser la consommation d'énergie, la couverture et la qualité des réseaux de capteurs.

### 1.7.7 Coopération entre la couche réseau et la couche Mac

Les communications dans les réseaux de capteurs sans fil sont assurées en utilisant les protocoles de routage et les protocoles MAC indépendamment comme dans le modèle classique OSI. Ceci présente des problèmes au niveau des métriques impliquées dans les deux couches réseau et liaison de données, comme le délai ou la consommation d'énergie. Les protocoles Cross-Layer exploitent l'indépendance entre les deux couches soit en les regroupant en une seule couche ; soit en assurant des communications entre les deux couches.

Le protocole cross-layer MAR-WSN [LFBG15] est un protocole de routage qui utilise les informations temporelles du protocole MAC et exploite la connaissance du voisinage à deux sauts. Après l'utilisation des couches MAC (TDMA) ou les protocoles de routage pour optimiser la gestion de l'énergie, on essaye maintenant de faire coopérer ses couches pour la minimisation de la gestion de l'énergie des réseaux de capteurs.

## 1.8 Conclusion

Les réseaux de capteurs sans fil présentent un intérêt important pour le futur surtout avec la miniaturisation et les progrès d'allongement de la durée de vie des capteurs. Ces progrès permettront d'étendre de nouveaux les domaines d'application. Dans ce chapitre, nous avons réalisé un aperçu sur les réseaux de capteurs sans fil en présentant leurs architectures de communication, la pile protocolaire, les diverses applications sans oublier les contraintes liées à ce domaine. En effet, les réseaux de capteurs se caractérisent par une capacité énergétique limitée rendant l'optimisation de la consommation d'énergie dans ces réseaux comme une tâche critique pour prolonger la durée de vie du réseau. Nous aborderons dans le chapitre suivant la gestion de l'énergie dans les réseaux de capteurs.

# Chapitre 2

## Gestion de l'énergie dans les réseaux de capteurs

### Sommaire

---

<b>2.1</b>	<b>Introduction</b>	<b>23</b>
<b>2.2</b>	<b>Consommation d'énergie dans les réseaux de capteurs</b>	<b>24</b>
2.2.1	Énergie de capture	24
2.2.2	Énergie de traitements	24
2.2.3	Énergie de communication	24
2.2.4	État du module radio	25
2.2.5	Accès au medium de transmission	26
2.2.6	Modèle de propagation radio	27
2.2.7	Formation des clusters	28
<b>2.3</b>	<b>Trois familles de protocoles de routage</b>	<b>28</b>
2.3.1	Routage à plat ou orienté données	28
2.3.2	Routage hiérarchique	29
2.3.3	Routage géographique	30
<b>2.4</b>	<b>Les techniques du clustering</b>	<b>31</b>
2.4.1	Les algorithmes et les limites	32
2.4.2	Quelques protocoles de clustering	34
<b>2.5</b>	<b>Agrégation et compression</b>	<b>37</b>
<b>2.6</b>	<b>Conclusion</b>	<b>38</b>

---

## 2.1 Introduction

Les nœuds-capteurs sont alimentés généralement par des batteries inéchangeables et non rechargeables. Ils doivent donc économiser leur énergie pour assurer une longue durée de vie au réseau de capteurs. Le remplacement des batteries n'est pas une option

envisageable pour des réseaux avec des milliers de nœuds ou avec l'hostilité des lieux de déploiement. La durée de vie d'un nœud capteur dépend donc fortement de la durée de vie de la batterie associée. Afin de concevoir des solutions efficaces en énergie, il est extrêmement important de faire d'abord une analyse des différents facteurs provoquant la dissipation de l'énergie d'un nœud-capteur et du réseau.

## 2.2 Consommation d'énergie dans les réseaux de capteurs

Cette partie présente les différentes consommations énergétiques des principaux composants d'un capteur. Elle décrira également certaines tâches de consommations d'énergie dans un réseau de capteurs.

### 2.2.1 Énergie de capture

L'unité de capture 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 Converter ». Le capteur est responsable de la fourniture des signaux analogique/numérique. Ce dernier transforme ses signaux en un signal numérique compréhensible par l'unité de traitement. Il existe plusieurs types de composants qui mesurent les paramètres de l'environnement comme la température, le son, l'image, la pression, etc. L'énergie de capture est utilisée pour accomplir les tâches suivantes : échantillonnage, conversion analogique/numérique et activation de la sonde de capture. En général, l'énergie de capture [ZLGZ11] représente un faible pourcentage de l'énergie totale consommée par un nœud comme le montre la figure 2.1. L'énergie consommée lors de la capture peut être réduite en utilisant des composants à faible consommation mais en réduisant ainsi leurs performances, ou bien par la suppression de capture inutile en réduisant les durées de capture.

### 2.2.2 Énergie de traitements

L'unité de traitement ou C.P.U (voir figure 2.1) regroupe un processeur et une unité de mémoire réduite qui permet de stocker les données. Cette unité exécute les tâches de perception qui lui sont assignées. Elle fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs (TinyOS par exemple). Elle exécute les protocoles de communications qui permettent de faire communiquer le nœud avec les autres. Cette partie présente les différentes consommations énergétiques des principales composantes d'un capteur [IMK11]. Elle décrira également certaines tâches de consommations d'énergie dans un réseau de capteur.

### 2.2.3 Énergie de communication

Cette unité est responsable de toutes les émissions et réception des données sur un médium sans fil. Elle est généralement de type radio-fréquence. Les unités de transmission

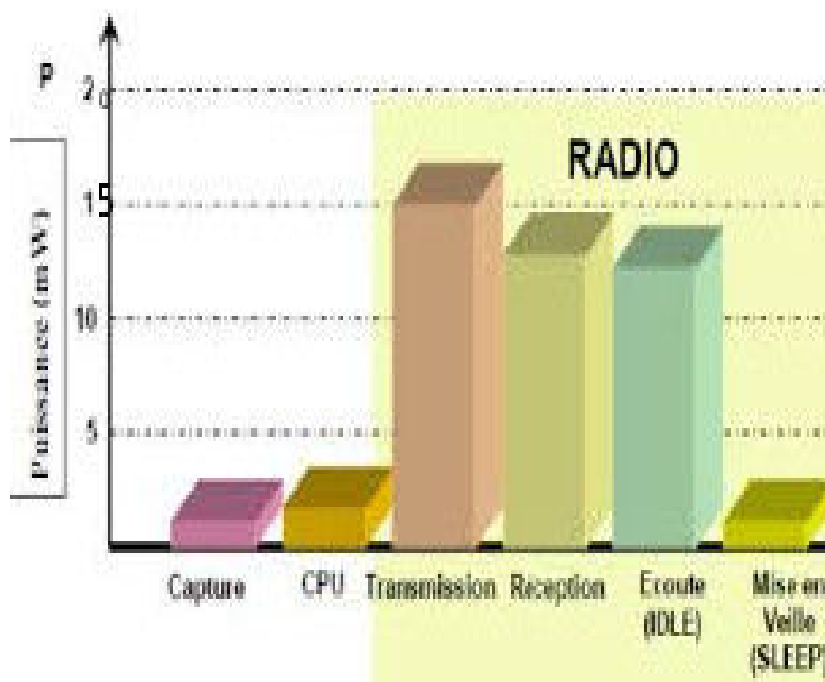


FIGURE 2.1 Schéma de consommation d'énergie d'un capteur

de type radio-fréquence comprennent des circuits de modulation, démodulation, filtrage et multiplexage. Il a été montré que la radio consomme beaucoup plus d'énergie dans les modes transmission et réception (figure 2.1). Le transport de 1 Ko de données sur 100 m consomme autant d'énergie que le traitement de 3 millions d'instructions.

Concevoir des unités de transmission de type radio-fréquence avec une faible consommation d'énergie est un défi [EH12], car pour qu'un nœud ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant et donc une énergie consommée importante. L'alternative consistant à utiliser de longues antennes n'est pas possible à cause de la taille réduite des micro-capteurs. La minimisation d'énergie pendant la communication est principalement liée aux protocoles développés pour la couche MAC et la couche réseau. Le but des protocoles réseau est de trouver les routes optimales en termes de consommation d'énergie et les protocoles MAC ont été développés essentiellement pour essayer d'éviter la collisions en aidant les nœuds à décider quand et comment ils peuvent accéder au support. En effet, la perte d'énergie due à un mauvais acheminement des paquets de données a un impact sur la durée de vie du réseau.

#### 2.2.4 État du module radio

La radio opère dans quatre modes de fonctionnement : émission, réception, écoute et sommeil. Chaque mode est caractérisé par une quantité différente de consommation d'énergie. Une observation importante dans le cas de la plupart des radios est que le mode "idle" induit une consommation d'énergie significative, presque égale à la consommation en mode réception [XHE01]. Pour économiser l'énergie en prolongeant ainsi la durée de vie d'un réseau, la solution la plus efficace consiste à éteindre la partie radio dès que possible.



Pourtant, afin d'assurer la transmission multi sauts, ces nœuds doivent aussi se réveiller de temps en temps pour relayer des données éventuelles. L'alternance de la période active et la période inactive du « transceiver » forme un cycle (peut être périodique ou non). Dans la littérature, le « duty-cycle » est défini comme la proportion de la période active sur la durée totale d'un cycle (période active + période inactive). Plus le duty-cycle est faible, moins le nœud dépense de l'énergie. Le lien entre le duty-cycle et la durée de vie de la batterie est souvent approximée par une relation linéaire, même si dans la pratique il est non-linéaire et dépend de la fréquence des sollicitations de la batterie. Toutefois, la transition entre les modes de fonctionnement implique un surplus d'énergie et de latence.

### 2.2.5 Accès au medium de transmission

Les collisions, l'écoute involontaire « overheard », l'écoute passive « idle listening » et l'utilisation de données de contrôle ont été identifiées comme sources principales de consommation d'énergie .

#### La collision

La collision est le cas où deux nœuds transmettent les trames de données en même temps vers un seul destinataire, cela génère une collision sur le récepteur. Dans le cas d'une collision entre deux ou plusieurs paquets, ces derniers vont être rejetés et par conséquent retransmis par leurs émetteurs. Cette retransmission va augmenter la consommation d'énergie des nœuds capteurs émetteurs [ISS12].

#### L'écoute passive

Cette technique est utilisée quand le nœud reste à l'écoute du médium pour recevoir des données possibles. Ce phénomène est très coûteux en énergie surtout dans le cas d'application qui ne nécessitent pas vraiment un échange de données.

#### L'écoute involontaire

Le phénomène « d'Overheard » c'est à dire que le nœud reçoit des paquets ou écoute les trames qui sont normalement destinés à d'autres nœuds. Puisque le médium est un environnement commun, lorsqu'un émetteur transmet ses trames de données, tous les nœuds qui se trouvent autour de lui sont obligés d'écouter cette transmission. Cette écoute passive est nécessaire pour déterminer le moment où le médium de transmission se libère afin de transmettre les trames de données. Ce phénomène peut constituer une cause importante de perte d'énergie dans le cas d'une zone à forte densité avec un trafic assez volumineux.

#### L'écoute non-active

C'est le temps où un nœud écoute le médium pour attendre une éventuelle transmission vers lui. Quand un nœud ne peut pas savoir le moment où les autres lui envoient des trames

de données, il doit toujours mettre son transceiver en marche. De ce fait, s'il n'y a pas de transmission vers lui, ce nœud consomme de l'énergie pour rien.

### L'utilisation de données de contrôle

Le canal sans fil est un médium de transmission bruité et il y a très souvent des erreurs de transmission. Alors, l'utilisation de paquet de contrôle est une méthode efficace pour contrôler les trames d'erreurs. Cependant, lorsqu'ils ne contiennent aucune donnée et qu'ils consomment, c'est un gaspillage d'énergie.

### Le phénomène d'overemitting

Le phénomène d'« overemitting » est causé par la transmission d'un message vers un nœud qui n'est pas encore prêt à recevoir des données. C'est aussi une perte d'énergie occasionnée. Les protocoles MAC ont été développés essentiellement pour essayer d'éviter ce genre de collisions en aidant les nœuds à décider quand et comment ils peuvent accéder au support.

## 2.2.6 Modèle de propagation radio

Pour transmettre un message de  $k$  bits sur une distance de  $d$  mètres, l'émetteur consomme comme le montre la figure 2.2 :

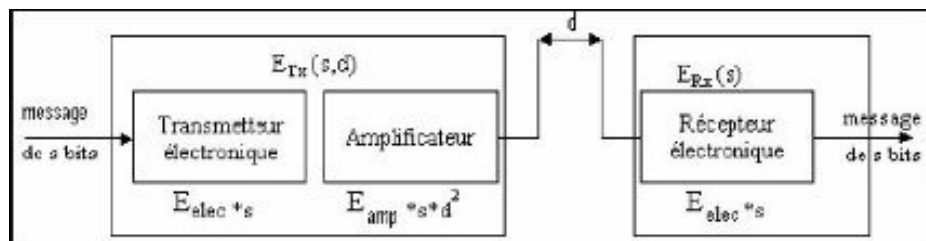


FIGURE 2.2 Schéma de formule de calcul de la consommation d'énergie

$$E_t(k, d) = E_{elec}(k) + E_{amp}(k, d) \quad (2.1)$$

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

$$E_r(k) = E_{elec}(k) = E_{elec} * k \quad (2.2)$$

Avec :  $E_{elec}$  : énergie de transmission réception électronique ;  $k$  : taille d'un message ;  $d$  : distance entre l'émetteur et le récepteur ;  $E_{Tx_{amp}}$  : énergie d'amplification ;  $E_{amp}$  : facteur d'amplification.

### 2.2.7 Formation des clusters

Les approches de clustering, proposées dans la littérature et comparées entre elles, utilisent un ou plusieurs critères pour le choix d'un CH (identifiant, degré, densité, mobilité, distance, énergie résiduelle, temps de service, sécurité, nature des informations collectées ou une combinaison de ces critères). La communication entre les capteurs dans le but d'élire les cluster-head CH et pour la formation des clusters est une source de consommation d'énergie considérable du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits. Sa consommation est en fonction du scénario de l'application, et il fait en général un compromis entre le niveau de performance et la consommation d'énergie.

## 2.3 Trois familles de protocoles de routage

Le routage est une méthode d'acheminement des informations vers la bonne destination à travers un réseau de connexion donné. Il consiste à assurer une stratégie qui garantit, à n'importe quel moment, un établissement de routes qui soient correctes et efficaces entre n'importe quelle paire de nœuds appartenant au réseau, ce qui assure l'échange des messages d'une manière continue. Les contraintes énergétiques des nœuds capteurs soulèvent des enjeux complexes pour la conception des protocoles de routage dans les réseaux de capteurs sans fil. Les protocoles proposés dans la littérature visent à équilibrer la charge, minimiser la consommation d'énergie lors de la transmission d'un paquet de bout en bout et éviter les nœuds à faible résidu d'énergie. Actuellement il existe plusieurs types de protocoles de routage pour les réseaux de capteurs, chacun ayant des propriétés différentes. Parmi les différents protocoles existants, il y a le routage à plat ou données, le routage hiérarchique et le routage géographique.

### 2.3.1 Routage à plat ou orienté données

Dans cette famille, les nœuds jouent le même rôle ou ont la même fonctionnalité. Ils collaborent ensemble pour exécuter les tâches spécifiques. Etant donné qu'il est difficile d'attribuer des identifiants pour chaque capteur, la station de base interroge le réseau ou une partie du réseau sur un attribut particulier du phénomène surveillé. Les capteurs se trouvant dans la zone spécifiée vont répondre à la station de base. Ces protocoles permettent d'économiser l'énergie en interrogeant les capteurs en fonction de leurs intérêts en matière de données. Le routage est assuré par un modèle à base d'interrogation. Parmi les protocoles de routage orientés données, nous pouvons énumérer les plus connus SPIN [AKK04] et DD [IGE<sup>+</sup>03].

#### **SPIN (Sensor Protocols for Information via Negotiation)[AKK04]**

Heinzelman et al. ont proposé une famille de protocoles appelée SPIN reposant sur un modèle de négociation afin de propager l'information dans un réseau de capteurs. Le but de SPIN est de pallier aux problèmes de l'inondation. SPIN adopte deux principes. La négociation : pour éviter le problème d'implosion, SPIN précède l'émission d'une donnée par sa

description, en utilisant la notion de métadonnées. Le récepteur aura le choix par la suite d'accepter la donnée ou non. Ce mécanisme permet aussi de régler le problème de chevauchement. L'adaptation aux ressources : d'une manière continue, les nœuds contrôlent leur niveau d'énergie. Le protocole SPIN accommode son exécution suivant l'énergie restante du capteur, et modifie en conséquence le comportement du nœud. Les communications dans SPIN se font en trois étapes. Lorsqu'un nœud veut émettre une donnée, il émet d'abord un message ADV contenant une description de la donnée en question. Un nœud recevant un message ADV, consulte sa base d'intérêt. S'il est intéressé par cette information, il émet un message REQ vers son voisin. En recevant un message REQ, l'émetteur transmet à l'intéressé la donnée sous forme d'un message DATA. L'avantage principal de SPIN est l'élimination des envois redondants des données en utilisant la négociation. En plus, les changements de la topologie n'affectent pas les performances du protocole réseau car chaque nœud ne se base que sur les informations locales pour la prise de décision ; il n'a besoin de communiquer des informations de contrôle qu'avec ses voisins directs. Néanmoins, cet avantage peut générer un inconvénient majeur qui est la non garantie de la livraison des données dans le cas où le nœud ayant des données sollicitées est injoignable. D'autres variantes de SPIN sont proposées pour combler les lacunes de la version originale telles que : SPIN-BC, SPIN-PP, SPIN-EC, SPIN-RL.

### **DD (Directed Diffusion)[IGE<sup>+</sup>03]**

DD est un protocole de propagation de données permettant d'utiliser plusieurs chemins pour le routage d'information. Le puits peut diffuser un intérêt sous forme de requête, afin d'interroger le réseau sur une donnée particulière. Les collecteurs de données transmettent un message intéressé aux capteurs et seuls les nœuds intéressés répondent. Les émissions comme les réponses aux messages d'intérêts établissent ainsi des trajets entre un collecteur de données et des capteurs intéressés. DD repose sur quatre éléments : nomination des données, propagation des intérêts et établissement des gradients, propagation des données et renforcement des chemins. La station de base diffuse l'intérêt à ses voisins qui à leur tour le diffusent à leurs voisins, et ainsi de suite.

### **2.3.2 Routage hiérarchique**

Dans ce type de routage, il y a une notion de lien entre les différents nœuds du réseau. Chaque nœud possède une autorité plus ou moins grande, ce qui permet à ses nœuds de contrôler des zones plus ou moins vastes afin de faciliter le routage. En effet certains nœuds sont élus et assument des fonctions particulières qui conduisent à une vision en plusieurs niveaux de la topologie du réseau. Ainsi, on distingue des clusters ou grappes. Chaque grappe dispose d'une tête de grappe ou CH, élue ou désignée, qui constitue le routeur ou la passerelle pour toute la grappe. Le CH récolte les données de son cluster, les fusionne, les filtre et les achemine à la station de base. Les CH peuvent former, elles aussi, un autre niveau de grappes ayant une tête, et ainsi de suite. Ceci donne au réseau une structure hiérarchique arborescente qui conserve l'énergie des nœuds et allège l'activité du réseau. Les nœuds d'une même grappe peuvent occuper à tour de rôle la fonction de CH, pour équilibrer la consommation d'énergie entre eux, comme proposé par Heinzelman et

al. (2002)[HCB02]. Nous reviendrons plus en détail sur les mécanismes de formation de grappes et leurs avantages en termes de conservation de l'énergie.

### 2.3.3 Routage géographique

Ce protocole prend en compte les positions physiques des nœuds.[XHE01] L'idée générale des protocoles de routage géographique est d'utiliser des informations géographiques pour acheminer les paquets en calculant et en passant par le chemin le plus court. Dans cette famille, les nœuds connaissent leurs positions et sont capables de connaître la position des autres nœuds du réseau. Il est facile de concevoir qu'un nœud peut facilement choisir parmi ses voisins un relais pour acheminer un paquet dont il connaît la destination finale et donc aussi sa position. Le principal inconvénient des protocoles géographiques est la connaissance de la position de tous les capteurs en utilisant le GPS qui consomme beaucoup l'énergie et aussi vu la taille de la mémoire, ce n'est pas adapté aux réseaux de capteurs. D'autres techniques de localisation plus adaptées existent comme la triangulation, la trilatération ou la proximité qui exploitent la puissance des signaux ou le temps de transmission des paquets afin de calculer la distance entre les nœuds. Cette localisation à l'aide de nœuds de références présente des erreurs de précision. Parmi les protocoles géographiques, nous pouvons citer GEAR, SAR, SPEED, GAF

#### **GEAR (Geographic and Energy Aware Routing)[YGE01]**

GEAR étend DD afin de réduire le nombre d'intérêts circulant dans le réseau. Quand l'information de la localisation des nœuds est disponible et que les requêtes portent sur la localisation, les intérêts sont diffusés à la région concernée, et pas à tout le réseau. Ce qui permet d'obtenir une réduction des communications grâce à la localisation de la diffusion. GEAR nécessite que chaque capteur connaisse sa position, ce qui est une charge énergétique supplémentaire.

#### **GAF (Geographic Adaptive Fidelity)[XHE01]**

Ce protocole partage le réseau en plusieurs zones virtuelles. Chaque nœud est affecté à une zone donnée selon ses coordonnées géographiques. Les nœuds appartenant à la même zone sont considérés comme équivalents en coût de routage. On choisit un nœud de chaque zone pour router les données et les autres sont mis en mode sommeil pour conserver de l'énergie. La durée de vie du réseau est augmentée en fonction du nombre total des nœuds inactifs. Dans GAF, il faut noter qu'un nœud est soit en mode découvert pour déterminer ses voisins de zone, soit en mode actif s'il participe au routage et soit endormi s'il ne participe pas au routage. GAF permet aussi de prendre en compte la mobilité où chaque nœud informe ses voisins de la période à laquelle, il va quitter son emplacement permettant une mise à jour future de la topologie. Cependant, GAF ne prend pas en compte l'énergie du nœud lors du choix du représentant, ce qui peut entraîner une faille dans le réseau une fois que ce dernier épuise son énergie.

## SPEED [HSLA03]

SPEED maintient uniquement l'information du voisinage immédiat du nœud. Chaque nœud du réseau dans SPEED effectue un balisage périodique pour diffuser sa localisation géographique à ses voisins immédiats. Ainsi chaque nœud dispose d'une table de voisins pour stocker l'information reçue ou par balisage périodique ou à la demande entre les voisins. Ce protocole utilise un mécanisme de contre-pression pour annoncer et faire éviter aux paquets les liens congestionnés du réseau. Il utilise également une expédition non-déterministe pour assurer un équilibrage de charge entre chemins de routage concurrents. La possibilité de passage à l'échelle du fait que toutes ses opérations sont distribuées et basées sur l'information locale au nœud ;

La prise en charge des vides de routage. SPEED fournit trois types de services de communication temps-réel :

Unicast permettant à une zone du réseau détectant une certaine activité d'alerter une station de base distante.

Multicast dans une zone définie (area-multicast) donnant la possibilité à une station de base d'envoyer une commande ou requête à un secteur du réseau.

Anycast dans une zone définie (area-anycast) permettant l'interrogation d'un seul nœud situé dans un secteur donné du réseau.

SPEED a deux principales lacunes, à savoir la manière d'éviter les vides et la non-prise en compte de l'énergie dans ses décisions de routage.

## 2.4 Les techniques du clustering

Le clustering [BFH<sup>+</sup>13] dans les réseaux de capteurs sans fil est un moyen efficace de structurer le réseau. Son but est d'identifier un sous-ensemble de nœuds dans le réseau et de lui attribuer un chef (un cluster-head ou CH). Celui-ci aura en charge des tâches spécifiques comme l'agrégation de données collectées et de les transmettre à la station de base.

Le clustering regroupe un ensemble de techniques qui visent à regrouper les capteurs d'un réseau en des groupes selon leur rapprochement les uns des autres en ne se basant sur aucune information antérieure. Un système d'analyse en clusters prend en entrée un ensemble de capteurs et une mesure de similarité entre ces capteurs, et produit en sortie un ensemble de partitions décrivant la structure générale du réseau.

Nous présentons dans cette partie les algorithmes de base de regroupement en clusters comme le montre la figure 2.3 et les protocoles qui en découlent.

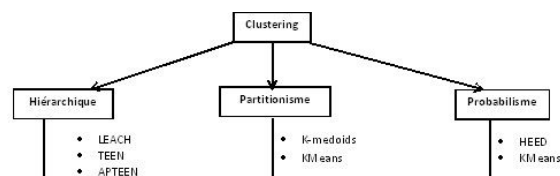


FIGURE 2.3 Algorithme de clustering

## 2.4.1 Les algorithmes et les limites

### Les algorithmes de base

Les méthodes peuvent être séparées en trois groupes [JMF99],[XW<sup>+</sup>05] :

**Algorithmes à partitionnement** Le plus simple et le plus utilisé des algorithmes partitionnels est l'algorithme K means. Il démarre d'une partition arbitraire des enregistrements sur les k clusters. A chaque itération il calcule les centres de ces clusters puis il effectue une nouvelle affectation des enregistrements aux plus proches centres. Il s'arrête dès qu'un critère d'arrêt est satisfait, généralement on s'arrête si aucun enregistrement ne change de cluster. L'algorithme peut être résumé comme suit :

---

**Algorithm 1** Algorithme général de partitionnement

---

1. Sélectionner une partition initiale contenant des enregistrements choisis arbitrairement, puis calculer les centres des clusters ;
  2. Calculer une liste des distances inter clusters et la trier dans l'ordre croissant ;
  3. Générer une nouvelle partition en affectant chaque enregistrement au cluster du centre le plus proche ;
  4. Calculer les nouveaux centres des clusters ;
  5. Répéter 2 et 3 jusqu'à ce que les enregistrements se stabilisent dans leurs clusters ;
- 

**Les algorithmes probabilistes** Les méthodes probabilistes supposent que les données suivent une certaine loi de probabilité. L'objectif est d'estimer les paramètres de cette loi et de définir un modèle de mélange de loi pour représenter les différents clusters. Ces méthodes font l'hypothèse qu'à chaque cluster  $C_i$  est associée une loi de probabilité qui permet de déterminer la probabilité d'appartenance de  $x$  à  $C_i$  . Les méthodes de clustering probabilistes cherchent à approximer les paramètres du modèle. La probabilité d'appartenance aux clusters peut être interprétée comme un degré d'appartenance à ce cluster. A noter qu'en général, les lois considérées sont supposées gaussiennes.

**Les algorithmes hiérarchiques** Dans ce type de clustering le nombre de clusters ne peut être connu à l'avance. Le système prend en entrée l'ensemble de données et fournit en sortie une arborescence de clusters. Il existe deux classes de ce type d'algorithmes : les algorithmes divisibles qui commencent à partir d'un ensemble de données et le subdivisent en sous-ensembles puis subdivisent chaque sous ensemble en d'autres plus petits, et ainsi de suite, pour générer en fin une séquence de clusters ordonnée du plus général au plus fin. La deuxième classe est celle des algorithmes agglomératifs qui considèrent chaque enregistrement comme étant un cluster indépendant puis rassemblent les plus proches en des clusters plus importants, et ainsi de suite jusqu'à atteindre un seul cluster contenant toutes les données. Un algorithme agglomératif suit généralement les étapes suivantes :

---

**Algorithm 2** Algorithme général de partitionnement

---

1. Placer chaque capteur dans son propre cluster ;
  2. Calculer une liste des distances inter clusters et la trier dans l'ordre croissant ;
  3. Pour chaque seuil de niveau de similitude préfixé  $dk$
  4. Relier tous les clusters dont la distance est inférieure à  $dk$  par des arrêtes à un nouveau cluster ;
  5. FPour
  6. Si tous les enregistrements sont membres d'un graphe connecté alors fin sinon aller au point 3
  7. le résultat est un graphe qui peut être coupé selon le niveau de similarité désiré ;
- 

**Les limites du clustering[JD88]**

Malgré l'existence d'un grand nombre de méthodes de clustering [JD88] ainsi que leur utilisation avec succès dans de nombreux domaines, le clustering pose encore de nombreux problèmes. Ces problèmes sont liés d'une part au manque de précision dans la définition de ce qu'est réellement un cluster mais également dans la difficulté de définir une mesure de similarité entre objets ou encore dans la définition d'une fonction objective pour un problème donné.

**Nombre de clusters [JD88]** Définir le nombre de clusters est un des problèmes les plus difficiles en clustering. En effet, il est souvent nécessaire de fournir le nombre de clusters souhaité comme paramètre de l'algorithme. Le choix du nombre de clusters a souvent été étudié comme un problème de sélection de modèle. Dans ce cas, l'algorithme est généralement exécuté plusieurs fois indépendamment avec un nombre de clusters différent. Les résultats sont ensuite comparés en se basant sur un critère de sélection qui permet de choisir la meilleure solution. Ce choix est toujours subjectif et fortement dépendant du critère sélectionné pour comparer les résultats. L'algorithme *isodata* [Ball et Hall, 1965] est basé sur l'algorithme *kmeans* mais permet une évolution du nombre de clusters pendant l'exécution. Des clusters sont fusionnés si le nombre d'objets dans un des clusters est inférieur à un seuil ou si deux clusters sont plus proches qu'un certain seuil. De plus, les clusters peuvent également être séparés en deux si leur dispersion (évaluée par exemple par l'écart type) dépasse un seuil. Cet algorithme permet, contrairement à *kmeans*, d'avoir un nombre de clusters variables pendant l'exécution. Cependant, ce type d'approche nécessite la définition de nombreux seuils et ne fait souvent que déplacer le problème du choix du nombre de clusters à la définition de ces seuils.

**Paramètres de clusters [JD88]**

Chaque algorithme de clustering possède un certain nombre de paramètres. Ces paramètres influent de manière plus ou moins importante sur les résultats obtenus. Ces paramètres sont nécessaires pour que l'algorithme soit relativement générique et pour que celui-ci soit applicable dans plusieurs cas. Plus le nombre de paramètres augmente plus



l'algorithme est adaptable et peut s'appliquer à une gamme de problèmes plus large. Cependant, l'augmentation des paramètres nécessite également de la part de l'expert une connaissance importante sur ses données et sur le fonctionnement de l'algorithme. Nous avons déjà soulevé le problème du nombre de clusters comme paramètre de nombreux algorithmes. Il existe bien d'autres paramètres qu'il serait impossible de citer avec exhaustivité. On pourra cependant noter le problème de l'initialisation qui est un problème important et qui concerne une grande partie des algorithmes existants.

**Choix des CH** Les approches de clustering, proposées dans la littérature et comparées entre elles, utilisent un ou plusieurs critères pour le choix d'un CH (identifiant, degré, densité, mobilité, distance, énergie résiduelle, temps de service, sécurité, nature des informations collectées ou une combinaison de ces critères). La communication entre les capteurs dans le but d'élire les CH est une source de consommation d'énergie importante.

## 2.4.2 Quelques protocoles de clustering

De nombreux travaux portent aujourd'hui sur la gestion de l'énergie consommée par les capteurs dans un réseau en prenant en considération, en premier lieu, les communications et les algorithmes de routage des données. Dans cette partie, nous allons examiner les protocoles de clustering.

### Les protocoles de base [HCB00, LR02, MA02, MA01]

Nous nous intéressons au routage hiérarchique dans le réseau de capteurs dont les fondements de bases sont : LEACH, PEGASIS, TEEN et APTEEN.

**LEACH (Low Energy Adaptive Clustering Hierarchy)[HCB00]** Al et Heinzelman ont introduit LEACH en 2000. LEACH est considéré comme le père fondateur du clustering. Cet algorithme permet de choisir aléatoirement 5% des nœuds homogènes comme CH et propose une rotation du rôle des CH. Les CH agrègent les données envoyées par les autres nœuds simples en un message pour une transmission directe à la station de base soit par l'intermédiaire des autres CH. Les CH peuvent communiquer entre eux et aussi avec la station de base qui est un ordinateur distant et souvent relié à l'Internet. Les CHs créent une table TDMA pour allouer un temps de transmission à chaque membre du cluster. Etant donné que chaque nœud connaît d'avance le slot de temps qu'il va occuper, il pourra se mettre à l'état de somnolence en dehors du slot qui lui est réservé. La table TDMA ainsi créée, pour chaque cluster, sera diffusée à tous les membres du cluster. La figure 2.4 illustre LEACH. LEACH réduit considérablement la consommation d'énergie dans les réseaux de capteurs mais soulève le problème d'une grande échelle et le choix du puits qui utilise la fonction probabiliste suivante :

$$T(i) = \begin{cases} \frac{P}{1 - P * (r \bmod \frac{1}{P})} & \text{if } i \text{ was not already elected as a pivot} \\ 0 & \text{else} \end{cases} \quad (2.3)$$

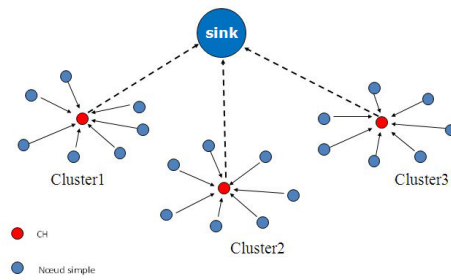


FIGURE 2.4 Schéma de communication dans LEACH

$P$  est un ratio de CH ;  $r \in [0, 1]$ . Si  $r$  est plus petit qu'une valeur de seuil  $T(i)$ , le nœud devient CH pour le cycle courant. La particularité de LEACH réside donc dans la non-nécessité de communication pour l'élection des CHs. LEACH arrive à économiser l'énergie mais le problème à grande échelle se pose.

**TEEN (Threshold sensitive Energy Efficient sensor Network protocol)** [MA01] proposé en 2001 par Arati Manjeshwar et Dharma P utilise l'architecture de LEACH. A la différence de LEACH où les CH transmettent des TDMA pour la synchronisation, les CH dans TEEN transmettent trois paramètres aux capteurs : **Attributs** : représentent la tâche demandée au capteur. **Hard threshold (HT)** : détermine la valeur critique après laquelle les membres doivent envoyer leur rapports de données. **Soft threshold (ST)** : spécifie le changement minimal obligeant le nœud à envoyer un nouveau rapport. TEEN permet d'implémenter un comportement réactif, tout en limitant le nombre de messages utilisés. Ainsi, on réduit le flux de données importantes et aussi la disparition d'un nœud dans l'anonymat total.

**APTEEN (Adaptive Periodic Threshold sensitive Energy Efficient sensor Network protocol)**[MA02] APTEEN est une extension de TEEN qui vise à l'adapter pour la surveillance périodique. APTEEN ajoute un autre paramètre à TEEN et oblige le nœud à faire un rapport d'activité au CH après un délai appelé « count time ». Ce temps est géré par TDMA comme dans LEACH. Comme dans LEACH et TEEN, la formation des grappes consomme beaucoup d'énergie.

**PEGASIS (Power-Efficient Gathering in Sensor Information Systems)** Dans [LR02] écrit en 2001 par Stéphanie L. et R. Cauligi, les nœuds du réseau sont organisés de façon à former une grande chaîne de proche en proche avec les nœuds voisins ou proches. Le principe d'une chaîne est qu'un nœud n'émet et ne reçoit qu'à partir d'un seul voisin.

Le dernier nœud à transmettre est atteint après un long moment. Ce qui pose un problème de temps et d'importance du message au niveau de la station de base.

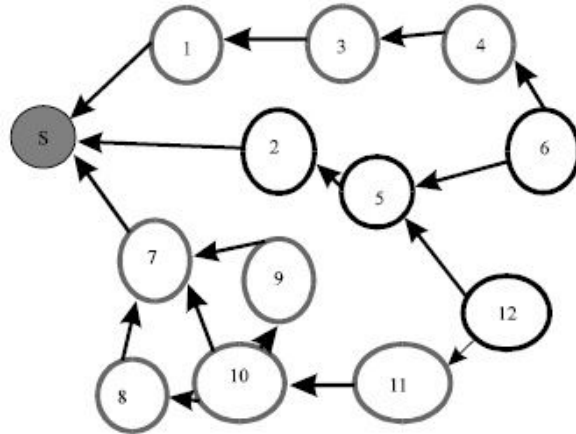


FIGURE 2.5 Architecture de communication PEGASIS

### Les protocoles dérivés des protocoles de base

Après les algorithmes de base présentés ci-dessus, les chercheurs se sont penchés sur le mode de formation des clusters qui est une source de consommation d'énergie. Un mode basé sur l'option probabiliste, l'option énergie restante des nœuds, à la distance entre les nœuds ou à une combinaison permet d'élire les CH et la formation des clusters. Un deuxième mode est la participation de la station de base dans le choix des CH et des grappes. Ainsi on distingue :

**HEED (A Hybrid, Energy-Efficient, Distributed)** En 2004, dans [YF04], Ossama Younis and Sonia Fahmy proposent HEED. Ils introduisent une fonction de coût pour chaque capteur qui est envoyée à chaque voisin. Avec HEED on choisit les CH dont la fonction de coût est minimale. Une méthode probabiliste avec itérations est également utilisée pour l'auto-élection des CH, mais avec considération de l'énergie résiduelle de chaque capteur. La formule est la suivante :

$$CH_{prob} = \max\left(C_{prob} \frac{E_i}{E_{max}}, P_{min}\right) \quad (2.4)$$

où  $C_{prob}$  est le ratio recherché de CHs sur le réseau,  $E_i$  l'énergie résiduelle du capteur  $i$  et  $p_{min}$  une probabilité minimale d'éligibilité pour assurer la terminaison de l'algorithme ( afin d'éviter un nombre d'itérations infini). A chaque itération  $CH_{prob}$  est multiplié par 2 ( pour atteindre un maximum de 1 où le capteur est garanti d'être élu CH ). Ainsi, HEED utilise l'énergie résiduelle pour le choix des CH et un coût de communication inter grappe pour les nœuds qui sont entre deux grappes. Mais il ne résout pas le passage à l'échelle.

**EECS (An Energy Efficient Clustering Scheme in Wireless Sensor Networks)** Publié en 2005, [YLCW05] traite de la collecte périodique en formant les grappes et les CH par une communication locale et en se basant sur l'énergie des capteurs. Au delà de LEACH, ses auteurs s'appuient aussi sur un équilibrage de charge entre les CH.

**MHEED (Multi-hop HEED)** A la différence de HEED, [MK07] introduit deux variantes. La première variante se base sur l'utilisation du meilleur chemin construit pour véhiculer les données. Dans la seconde, l'émetteur choisit son prochain relais de manière probabilisée pour équilibrer la charge entre les différents relais. MHEED tente de résoudre le problème de grande échelle dans HEED.

**CSOS (Cluster-based Self-Organisation algorithm for wireless Sensor network)** [LGF08, LGF10, Leh09] ont proposé une technique d'auto-organisation basée sur l'approche de clustering pour optimiser la consommation d'énergie dans ces réseaux. Cette technique regroupe les nœuds proches géographiquement en clusters. Elle implique des paramètres déterminants pour produire un nombre réduit de clusters homogènes en taille et en rayon avec une stabilité des clusters. Le poids de chaque capteur est calculé en fonction des paramètres suivants : la 2-densité, l'énergie restante et la mobilité. Le capteur ayant le plus grand poids dans son voisinage à 2 sauts devient cluster-head. En outre, la taille des clusters générés est comprise entre deux seuils qui représentent le nombre minimal et maximal de capteurs dans un cluster. Ces deux seuils sont choisis arbitrairement ou suivant la topologie du réseau. Dans un cluster, chaque capteur membre est au plus à deux sauts du cluster-head. Le processus d'élection des cluster-heads est périodique afin de distribuer équitablement la consommation d'énergie parmi les capteurs.

**CSA (Cluster-based Scheduling Algorithm)** [Leh09] propose un algorithme distribué d'ordonnancement d'activité des capteurs appelé CSA (Cluster-based Scheduling Algorithm). CSA sélectionne les capteurs qui vont passer en mode actif après l'élection des cluster-heads et la formation des clusters suivant l'algorithme d'auto organisation CSOS. Cet algorithme a contribué à assurer la k-couverture totale de zone d'intérêt en impliquant un nombre minimal de capteurs.

## 2.5 Agrégation et compression

Une manière d'économiser l'énergie pendant la collecte des données est d'agréger ou réduire les données en termes de volume ou de nombre de paquets pour limiter le nombre de messages nécessaires à leur remontée vers la passerelle d'accès. Cette technique consiste à profiter de la nature multi-saut des protocoles de routage. C'est-à-dire que chaque capteur agrège sa mesure avec celle de son prédécesseur. Par exemple il l'additionne avant de transmettre la réponse au capteur suivant. ON peut soit utiliser un opérateur classique (somme, moyenne, minimum,...) sur les mesures de différents capteurs, soit effectuer un simple ajout d'une mesure à une autre dans un même paquet réseau. Ainsi, la quantité de données est réduite tout en parcourant le réseau vers la station de base. On minimise le nombre de messages envoyés et l'énergie consommée est moindre par rapport à une solution où tous les capteurs communiquent directement avec la passerelle [KEW02]. Une agrégation de données appropriée est spécifique à l'application [FRL05, FLN03]. Il faut noter qu'avec cette technique, il n'est pas possible de calculer les fonctions d'agrégation qui ont besoin de l'ensemble des mesures avant de faire le calcul (histogramme, médiane, quantile,...). La compression de données peut être appliquée également pour réduire la

quantité d'informations transmises par les nœuds sources. Ce régime implique l'encodage d'information au niveau des nœuds qui engendrent des données, et le décodage au niveau de la station de base. Il existe différentes méthodes de compression de données citées dans [TR04, CCGF09].

## **2.6 Conclusion**

Ce deuxième chapitre nous a permis d'exposer les sources de consommation d'énergie comme la formation des clusters ou encore l'utilisation excessive de la radio dans les réseaux de capteurs fixes. A travers ce chapitre, nous avons exposé les grands concepts du clustering qui permettent d'économiser l'énergie, les principales méthodes existantes ainsi que leur évaluation et leur comparaison. Nous avons également présenté les nombreux problèmes et limites du clustering. Il en ressort qu'un nombre important de méthodes existent et qu'il est souvent difficile de faire un choix parmi celles-ci.

Ce chapitre sera une référence pour la suite de nos travaux. Il nous permettra de proposer dans la partie contribution un nouvel algorithme permettant d'économiser l'énergie en évitant par exemple la communication des nœuds pour la désignation des CHs.

# Chapitre 3

## La mobilité dans les réseaux de capteurs : Préserver l'énergie et la continuité du réseau en cas de mobilité

### Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>39</b>
<b>3.2</b>	<b>Avantages de la mobilité</b>	<b>40</b>
<b>3.3</b>	<b>Types de mobilité</b>	<b>41</b>
3.3.1	La micro-mobilité	41
3.3.2	La macro-mobilité	44
<b>3.4</b>	<b>Routage mobile</b>	<b>46</b>
3.4.1	Le routage pro actif	46
3.4.2	Le routage réactif	50
3.4.3	Le routage hybride	52
3.4.4	L'adaptation à la métrique énergie	57
<b>3.5</b>	<b>Conclusion</b>	<b>58</b>

---

### 3.1 Introduction

La mobilité est une réalité de nos jours dans les réseaux de capteurs [BR14]. Elle peut être liée à un déplacement naturel des porteurs de capteurs ou à une stratégie volontaire pour améliorer la durée de vie du réseau. Nous allons aborder cette mobilité sous la lumière de ces deux aspects dans cette partie. La durée de vie d'un réseau de capteurs est fortement liée à ses capacités de stockage et de calcul. Les batteries alimentent les capteurs qui sont déployés parfois dans des conditions d'accès difficiles. Un échange manuel de ses batteries n'est pas à proscrire mais d'autres procédures permettant de donner une longue durée de vie au réseau de capteurs peuvent être primordiales. Dans les réseaux de capteurs à nœuds fixes et à station immobile, les paquets en provenance de nœuds

capteurs traversent le réseau vers le puits en suivant un chemin multi-hop. Lorsque le puits est statique, quelques chemins peuvent être plus chargés que d'autres en fonction de la topologie de réseau et du taux de génération des paquets au niveau des sources. Généralement, les nœuds plus proches du puits relayent plus de paquets de sorte qu'ils sont soumis à l'épuisement prématuré de l'énergie, même lorsque des techniques pour conservation de l'énergie sont appliquées [LM07]. La mobilité est exploitée dans cette situation pour réduire la consommation d'énergie [ACDFP09]. Dans [SSB14, WBMP05, GDPV03, AY05], le flux de circulation peut être modifié si un dispositif mobile désigné se rend responsable de la collecte des données (données mobiles collecteur). Les nœuds ordinaires attendent le passage du nœud collecteur pour router les messages vers lui, de sorte que la communication avec le collecteur de données mobiles ait lieu à proximité (directement ou au plus avec un nombre limité multi-hop). En conséquence, les nœuds ordinaires peuvent économiser de l'énergie grâce à la réduction des erreurs de liaison, à la réduction de la distance d'envoi du nombre de sauts par rapport à un réseau statique et à la variation des nœuds proches du puits pour la retransmission.

Nous allons donc, dans cette partie, présenter les travaux de la littérature sur ce problème. Il nous a semblé opportun de nous concentrer, dans un premier temps, sur les types de mobilité avant d'aborder le routage dans des réseaux de capteurs à entités mobiles. Ce chapitre va nous donner les bases nécessaires pour la compréhension de notre contribution sur ce type de réseau.

## 3.2 Avantages de la mobilité

Au départ, les spécialistes des réseaux de capteurs ont considéré l'utilisation des périphériques mobiles comme une charge supplémentaire surtout au niveau du réseau informatique. Ensuite, les chercheurs ont trouvé une technique pour économiser l'énergie à travers la mobilité. La durée de vie d'un réseau de capteurs est fortement limitée par la capacité des nœuds statiques alimentés par leur batterie inéchangeable et irremplaçable à cause de l'hostilité des zones de déploiements. De plus, la radio est un consommateur majeur de l'énergie d'un capteur [ASSC02a].

L'énergie dépensée pour relayer les données pourrait être diminuée si les données étaient transmises sur un seulement ou deux sauts. Ceci peut être réalisé en ayant un dispositif mobile proche des nœuds de capture de telle sorte que les données peuvent être envoyées au dispositif mobile sur un très petit nombre de sauts. Le dispositif mobile n'a pas d'énergie limitée car il peut retourner à la base et se recharger.

[BCM<sup>+</sup>08, JSB<sup>+</sup>06, SKJ<sup>+</sup>06] donnent un autre avantage de la mobilité concernant la fiabilité des données. La probabilité d'erreur augmente avec le nombre de sauts qu'un paquet de données doit parcourir vers la station de base. Si nous réduisons le nombre de sauts, on réduit immédiatement la probabilité d'erreur. Ceci augmente non seulement la qualité des données reçues, mais réduit l'énergie dépensée au niveau des nœuds statiques en réduisant les retransmissions.

Un des avantages réside aussi dans la gestion des réseaux clairsemés et déconnectés. Cette approche consiste à mettre à profit la mobilité des capteurs pour rétablir une certaine forme de connectivité entre les sous réseaux de capteurs ou entre des clusters. En effet, il

arrive qu'un ensemble de capteurs soit isolé du reste du réseau [JSB<sup>+</sup>06]. Cela permettra d'économiser de l'énergie d'émission.

### 3.3 Types de mobilité

Les stratégies reposant sur la mobilité peuvent être classées en deux ensembles : la micro-mobilité et la macro-mobilité. Dans la micro-mobilité, les déplacements de chaque nœud sont déterminés indépendamment les uns des autres. La macro-mobilité prend en compte la corrélation de déplacements entre certains nœuds. Elle divise les nœuds en plusieurs groupes et définit une relation entre les unités mobiles appartenant à un même groupe [CBD02, SJ04, LW07]. Il existe en général trois types de loi de mouvement : aléatoire, déterministe et hybride. Les modèles aléatoires présentent des déplacements arbitraires et sans contraintes d'environnement [KSJ<sup>+</sup>04]. Ce sont des modèles simples à mettre en œuvre et qui sont très utilisés. Les modèles déterministes, quant à eux, s'appuient sur des traces (comportements d'utilisateurs observés dans des systèmes réels). Les modèles hybrides réalisent un compromis entre simplicité et réalisme mais restent difficiles à mettre en place.

#### 3.3.1 La micro-mobilité

La micro-mobilité [JD88] s'intéresse généralement aux modèles de traces qui sont observés dans les systèmes réels [HGPC99]. Les traces fournissent des informations précises, particulièrement quand elles impliquent un grand nombre de participants pendant une longue période d'observation. La modélisation des environnements des réseaux de capteurs mobiles est très complexe et nécessite l'utilisation des modèles synthétiques. Les modèles synthétiques de mobilité d'entité les plus courants pour le réseau Adhoc sont :

Les modèles de mobilité aléatoire : On peut repartir ces modèles en trois sous modèles. Un modèle de mobilité simple qui choisit des directions et des vitesses de façon aléatoire, un modèle qui inclut des pauses entre les changements de direction et de vitesse et un modèle qui force les nœuds mobiles à se déplacer vers les bords du secteur de simulation avant de changer direction et de vitesse.

Le modèle de mobilité dans une région de simulation illimitée est le seul qui permet aux nœuds mobiles de se disperser dans le secteur, en éliminant les effets de la frontière sur l'évaluation des performances.

Le modèle de mobilité de Gauss-Markov utilise un paramètre d'accord pour changer l'aspect aléatoire dans le modèle de mobilité.

Le modèle probabiliste aléatoire utilise un ensemble de probabilités pour déterminer la prochaine position d'un nœud mobile.

Le modèle de mobilité des sections de ville : un secteur de simulation qui représente les rues dans une ville.

#### Le modèle de mobilité aléatoire

Les nœuds mobiles aléatoires font partie du champ capteurs. Ils font partie de l'environnement. Il peut être divisé en sous points comme :



**Le modèle purement aléatoire [MRS14, SSSM64, Kac47]** Dans le modèle Random Walk, un nœud mobile se déplace de son emplacement actuel vers un nouvel emplacement en choisissant aléatoirement une direction et une vitesse de déplacement. Dans ce modèle, le déplacement du nœud se fait pendant un temps  $t$  ou d'une distance  $d$ . Par la fin de laquelle une nouvelle direction et une nouvelle vitesse sont calculées. Le vecteur vitesse de chaque mobile est défini par sa norme et sa direction. Ce vecteur est modifié périodiquement et la nouvelle valeur de ses paramètres est choisie uniformément sur  $[V_{min}, V_{max}]$  pour la vitesse et  $]0, \Pi]$  pour la direction.  $V_{min}$  et  $V_{max}$  sont deux constantes représentant respectivement la vitesse minimale et maximale à laquelle un mobile peut se déplacer. Le modèle de mobilité Random Walk est un schéma de mobilité sans mémoire, la vitesse et la direction d'un nœud mobile sont indépendantes de son passé. Ce modèle n'était pas prévu pour décrire le déplacement de mobiles dans un réseau de capteurs mobile. Mais de nombreux travaux l'utilisent et étudient son influence sur les protocoles de communication.

**Le modèle de mobilité aléatoire avec pause [YLN03, MRS14]** Le modèle Random WayPoint (RWP) est similaire au modèle de mobilité Random Walk si le temps de pause est nul. Ce modèle prévoit un temps de pause entre chaque déplacement. Un nœud mobile commence par séjourner dans un endroit pendant une certaine période de temps. Une fois cette période terminée, le nœud mobile choisit une destination aléatoire et une vitesse qui est répartie uniformément entre  $[V_{min}, V_{max}]$ . Le fonctionnement de ce modèle est résumé par l'algorithme suivant : Dans ce modèle la distribution initiale des nœuds

---

**Algorithm 3** Algorithme Random WayPoint appliqué pour chaque mobile M

---

$P_{DestM} \leftarrow$  coordonnées cartésiennes du point vers lequel se dirige le mobile M ;

$V_M \leftarrow$  vitesse du mobile M définie sur l'intervalle  $[V_{min}, V_{max}]$  ;

$Pause_p \leftarrow$  temps de pause du mobile M défini sur l'intervalle  $[P_{min}, P_{max}]$  ;

1. Choisir une nouvelle valeur pour  $P_{DestM}$  uniformément dans chacune des directions ;
  2. Choisir une nouvelle valeur pour  $V_M$  uniformément sur l'intervalle  $[V_{min}, V_{max}]$  ;
  3. Déplacer ce mobile M vers  $P_{DestM}$  à la vitesse  $V_M$  ;
  4. Choisir une nouvelle valeur pour  $Pause_p$  uniformément sur l'intervalle  $[P_{min}, P_{max}]$  ;
  5. Effectuer une pause d'une durée  $Pause_p$  ;
  6. Recommencer à l'étape 1 jusqu'à la fin de la simulation ;
- 

est cruciale. De plus, on remarque une concentration des nœuds au centre de la zone de mobilité.

**Le modèle de mobilité de direction aléatoire [YLN03]** Random Direction pour corriger les défauts de Random WayPoint. Ces auteurs ont proposé un nouveau modèle : le Random Direction (RD) dont le fonctionnement est le suivant. Initialement, l'ensemble des mobiles est placé sur la zone de simulation en choisissant leur position uniformément

dans les deux directions de l'espace. Chaque mobile tire une direction et une vitesse, toutes deux choisies uniformément sur  $[0, 2\Pi]$  pour la direction et sur  $[V_{min}, V_{max}]$  pour la vitesse. Les mobiles vont ensuite se déplacer dans la direction choisie à la vitesse choisie jusqu'à ce qu'il rencontre le bord de la zone de simulation. Lorsqu'un mobile atteint cette bordure, il effectue une pause dont la durée est choisie uniformément sur  $[P_{min}, P_{max}]$ . A la fin de cette pause, le mobile choisit une nouvelle direction et une nouvelle vitesse. Contrairement au premier tirage, l'ensemble des valeurs possibles pour la direction est restreinte pour empêcher le mobile de sortir de la zone de simulation. Par exemple, dans le cas d'un mobile atteignant le bord supérieur de la zone de simulation, l'intervalle de définition sera réduit de moitié afin d'obliger ce mobile à redescendre. Le tirage de la vitesse se fait toujours uniformément sur  $[V_{min}, V_{max}]$ . Contrairement au Random WayPoint, RD ne concentre pas les mobiles au milieu de la zone de simulation. Il limite ainsi le phénomène d'oscillation du degré.

**Le modèle probabiliste aléatoire** La version probabiliste du modèle aléatoire utilise la probabilité pour déterminer la position suivante d'un nœud mobile. Cette position est représentée par trois différents états pour la position x et trois différents états pour la position y. L'état 0 représente la position courante ( x ou y ) pour un nœud mobile donné, l'état 1 représente la position précédente ( x ou y ) et l'état 2 représente la prochaine position du nœud s'il continue à se déplacer dans la même direction. [CS01] présente le Chiang Model. C'est un modèle utilisant une mémoire. Un nœud mobile mémorise 3 positions :  $(pos_{t-1}, pos_t, pos_{t+1})$ . La probabilité  $P(pos_a; pos_b)$  prédéfinie entre chaque paire de positions est la probabilité qu'un nœud passe de la position  $p_a$  à la position  $p_b$ . Ces probabilités sont utilisées pour calculer la nouvelle position d'un nœud. Ce modèle supprime les changements soudains. Son point faible est la difficulté de choix des probabilités.

### Le modèle de Gauss-Markov [WGG10]

Ce modèle de mouvement se caractérise par l'absence de temps de pause ou de changement brusque d'angle et de vitesse comme le montre le modèle à trajectoire aléatoire. Il utilise les chaînes de Markov pour faire varier le tirage aléatoire des vitesse et direction du modèle Random Walk tout en tenant compte des vitesse et direction précédentes en les faisant évoluer suivant un processus markovien. Dans le modèle de mobilité de Gauss-Markov, le temps est discrédité en intervalles de durée constante. A chaque instant t, un nouveau vecteur vitesse, est choisi. L'expression du vecteur vitesse dans un espace à deux dimensions est définie par les formules.

$$s_{n+1} = \alpha s_n + (1 - \alpha)\bar{s} + \sqrt{(1 - \alpha^2)}s_{x_n} \quad (3.1)$$

$$d_{n+1} = \alpha d_n + (1 - \alpha)\bar{d} + \sqrt{(1 - \alpha^2)}d_{x_n} \quad (3.2)$$

où  $s_{n+1}$  et  $d_{n+1}$  sont les nouvelles vitesse et direction du nœud mobile à l'instant n+1.  $\alpha, 0 \leq \alpha \leq 1$ , est le paramètre d'accord utilisé pour l'aspect aléatoire;  $\bar{s}$  et  $\bar{d}$  sont des constantes représentant la valeur moyenne de la vitesse de la direction quand n tend vers l'infini; et  $s_n$  et  $d_n$  sont des variables aléatoires d'une distribution de Gauss.

### 3.3.2 La macro-mobilité

Dans les modèles de déplacement énuméré, la trajectoire de chaque élément du réseau est indépendante de celle des autres mobiles. Il est plus réaliste et souhaitable de prendre en compte le mouvement d'un nœud mobile dans un ensemble de nœuds pour un but commun. En effet, les membres d'un même groupe suivent globalement la même trajectoire et sont proches les uns des autres. La durée de vie des connexions à l'intérieur d'un groupe est donc plus élevée qu'entre des mobiles appartenant à des groupes différents. Des travaux proposent des modèles qui prennent en compte cette dépendance. Le modèle exponentiel aléatoire corrélé utilise une fonction mathématique pour générer les mouvements de façon corrélée. Le modèle de mobilité de colonne est un modèle où l'ensemble des nœuds mobiles forment une colonne et se déplacent uniformément en avant dans une direction particulière. Le modèle de mobilité de communauté nomade est défini comme un ensemble de nœuds mobiles se déplaçant ensemble d'un endroit à un autre. Le modèle de mobilité de poursuite est un modèle où un ensemble de nœuds mobiles suivent une cible particulière. Le modèle de mobilité d'un groupe avec le point de référence est un groupe de nœuds suivant le chemin parcouru par un centre (nœud) logique du groupe.

#### Le modèle exponentiel aléatoire corrélé

(Exponential Correlated Random Mobility Model)[CBD02] est l'un des premiers modèles de mobilité de groupe proposé. Il utilise une fonction de mouvement pour créer les mouvements du nœud mobile. Le groupe est vu comme un bloc en déplacement, étant donnée une position du groupe à un temps  $t$ , une fonction est utilisée pour définir la position à  $t + 1$ .

#### Le modèle de mobilité de colonne

Le modèle de mobilité de colonne [CBD02] en anglais Column Mobility Model s'avère utile pour modéliser une recherche sur le terrain. Ce modèle représente un ensemble de nœuds mobiles qui se déplace globalement en parallèle devant un axe donné. (Par exemple, une ligne de soldats marchant ensemble vers leur ennemi ou des opérations de recherche pour désactiver des mines antipersonnel). Chaque nœud a son propre point de référence autour duquel il a un certain degré de liberté. Ces points de référence sont alignés et mis à jour régulièrement. Pour le référencement, la zone de simulation est découpée en une grille.

#### Le modèle de mobilité de communauté nomade

Comme les nomades qui se déplacent d'un endroit à un autre, le modèle de la communauté nomade (Nomadic community Mobility Model) [CBD02] représente des groupes de nœuds mobiles qui se déplacent ensemble d'un point à l'autre. Dans chaque communauté, les individus maintiennent leur propre espace personnel où ils se déplacent de manière aléatoire. Il donne un plus grand degré de liberté de mouvement que Column Mobility Model. Chaque nœud utilise un des modèles de mobilité d'entité autour d'un point de référence donné. Quand le point de référence change, tous les nœuds mobiles dans le

groupe se déplacent vers le nouveau secteur défini par le point de référence et commenceront à errer autour du nouveau point de référence. Les paramètres pour le modèle de mobilité d'entité définissent à quelle distance un nœud mobile peut errer autour du point de référence. Comparés au modèle de mobilité de colonne où chaque nœud a un point de référence individuel, les nœuds dans le modèle de mobilité nomade partagent un point de référence commun.

### Le modèle de mobilité de poursuite

Le modèle de mobilité de poursuite [CBD02] ou en Pursue Mobility Model essaie de représenter, comme son nom l'indique, des nœuds mobiles qui suivent une cible particulière. Ce modèle consiste en une équation de mise à jour simple pour la nouvelle position de nœud. La nouvelle position est fonction de l'ancienne position et de l'accélération de la cible.  $Nouvelle_{position} = ancienne_{position} + acceleration(target - ancienne_{position}) + random_{vector}$  où  $acceleration(target - ancienne_{position})$  est l'information sur le mouvement du nœud poursuivi, et  $random_{vector}$  est un décalage aléatoire pour chaque nœud. La valeur de  $random_{vector}$  est obtenue par un modèle de mobilité d'entité et l'aspect aléatoire des nœuds est limité afin de maintenir un chemin efficace du nœud poursuivi. Ce modèle est très réaliste et largement répandu dans la pratique.

### Le modèle de mobilité d'un groupe avec point de référence

Le modèle de mobilité d'un groupe avec point de référence [CBD02] en anglais Reference Point Group Mobility Model représente le mouvement aléatoire d'un groupe de nœuds aussi bien que le mouvement aléatoire de chaque nœud individuellement dans le groupe. C'est le mouvement aléatoire d'un groupe autour d'un centre logique. C'est le modèle le plus général dans la catégorie des modèles de groupes. Les mouvements du groupe sont basés sur le chemin parcouru par un centre du groupe qui est utilisé pour calculer le mouvement du groupe par l'intermédiaire d'un vecteur de mouvement qui peut être choisi aléatoirement ou être prédéfini. Le mouvement du centre du groupe caractérise complètement le mouvement de son groupe (la direction et la vitesse) Les nœuds individuels se déplacent aléatoirement par rapport à leurs propres points de référence prédéfinis, dont les mouvements dépendent du mouvement du groupe.

### Le modèle de mobilité avec obstacles

Le modèle de mobilité avec obstacle [JBRAS03, BSH03] a été conçu pour modéliser le mouvement des nœuds mobiles dans les terrains qui ressemblent à des topographies réelles. Des objets modélisent les bâtiments et d'autres structures qui empêchent les mouvements des nœuds, ainsi que leur transmission sans fil. En modélisant un tel terrain, un utilisateur peut définir les positions, les formes et les tailles de ces objets. Ce modèle peut manipuler des formes de positions arbitraires pour les objets, permettant de modéliser beaucoup de terrains réels. Les diagrammes de Voronoï permettent de partitionner un espace. Ce partitionnement est déterminé par un ensemble fini de points appelés germes. Un élément de cette partition, nommé cellule, associé avec un germe  $g$  est constitué des points de l'espace plus proches de  $g$  que de n'importe quel autre germe. Plusieurs modèles ont utilisé

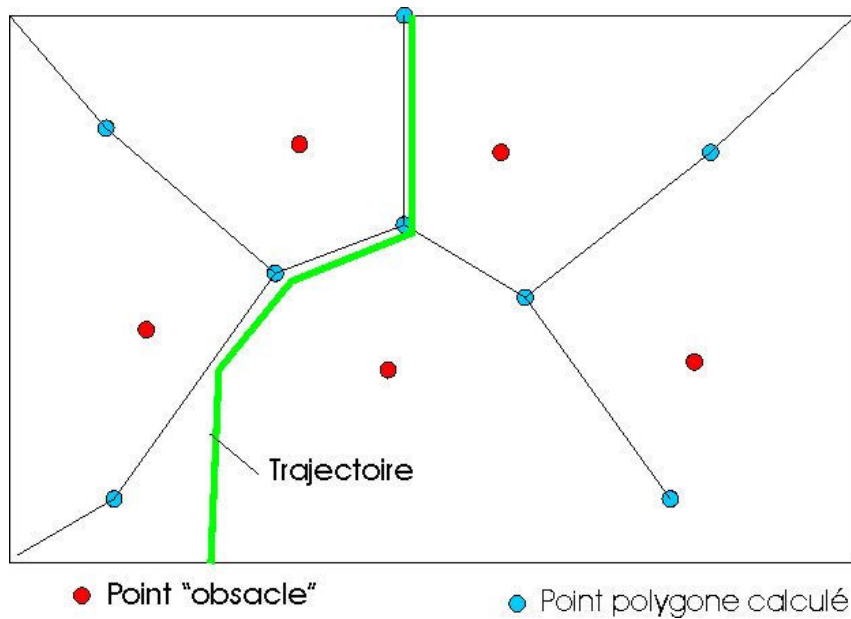


FIGURE 3.1 Mouvements avec obstacles utilisant le diagramme de Voronoï

ce diagramme pour permettre aux mobiles de contourner les obstacles présents dans la zone de simulation.

## 3.4 Routage mobile

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion établi. Son intérêt consiste à trouver le chemin optimal en tenant compte d'un certain nombre de critères de performance comme la bande passante, le délai de transmission et l'énergie des nœuds. Il doit aussi être capable de s'adapter aux événements venant perturber le réseau en cas de panne d'un nœud, d'une congestion. Dans le cas des réseaux de capteurs sans fil, l'architecture étant caractérisée par une absence d'infrastructure fixe, le réseau doit donc s'organiser automatiquement et réagir rapidement aux différents mouvements de ses capteurs qui le compose. Chaque capteur devient donc un nœud susceptible d'être mis à contribution pour participer au routage [SS14]. Il existe deux types de routage distinct pour l'acheminement des données. Le routage réactif et le routage pro actif. Dans le routage réactif, les routes sont créées à la demande, tandis que dans le routage pro actif, les routes sont maintenues en permanence. Il existe d'autres méthodes mixtes qui combinent le routage pro actif et le routage réactif.

### 3.4.1 Le routage pro actif

Les protocoles pro actifs sont basés sur la même philosophie que les protocoles de routage utilisés dans les réseaux filaires conventionnels. C'est-à-dire qu'elle est fondée sur la méthode "état de lien" et "vecteur de distance". La méthode état de lien repose sur le moindre changement dans le réseau. La méthode vecteur de distance est basée sur

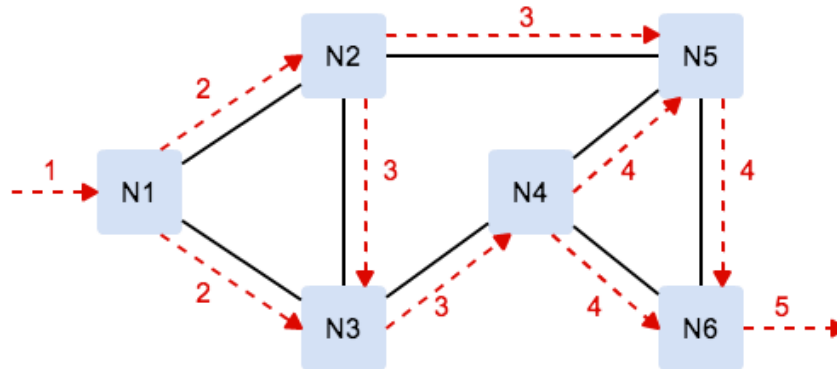


FIGURE 3.2 Routage par inondation

la propagation d'information concernant sa distance à tous les nœuds du réseau. Elles exigent des mises à jour périodique. Ces deux méthodes, en se basant sur des algorithmes de routage, recherchent toujours le plus court chemin entre l'émetteur et le récepteur. Ce calcul repose sur le nombre de sauts, ces sauts pouvant être pondérés selon différents critères comme le taux d'utilisation du lien, le délai. Le maintien des tables de routages est réalisé par inondation, lors des mises à jour périodique ou lors d'un changement d'état d'un lien. L'inondation consiste à propager à l'ensemble du réseau une information. L'émetteur initial envoie à tous ses voisins une information, ces derniers se chargeant de la rediffuser à leurs tours comme le montre la figure ???. Actuellement ils existent plusieurs protocoles pro actifs (DSDV, OLSR, FSR, WRP, GSR, HSR, ZHLS, CGSR, DREAM).

### Le Protocole de routage DSDV

Cet algorithme de routage de Perkins a été conçu pour les réseaux mobiles. Il a été créé à partir de l'algorithme distribué de Bellman-Ford. La métrique utilisée par DSDV [RT99, PB94] est le nombre de nœuds séparant un nœud de la destination. La table de routage de ce protocole prend en compte :

- toutes les destinations possibles
- le nombre de sauts nécessaires afin d'atteindre la destination
- le numéro de séquence associé à chaque nœud de destination

le numéro de séquence est utilisé pour faire la distinction entre les anciennes et les nouvelles routes, ce qui évite la formation des boucles de routage.

Les paquets de mise à jour contiennent :

le nouveau numéro de séquence incrémenté, du nœud émetteur.

les informations concernant chaque nouvelle route :

l'adresse de la destination.

le nombre de sauts entre le nœud et la destination.

le numéro de séquence des données reçues de la destination.

Lors d'une mise à jour, les données reçues sont comparées avec celle déjà disponibles. La route avec le plus grand numéro de séquence donc la plus récente sera conservée. Si elle possède le même numéro de séquence alors celle avec la meilleure métrique sera retenue. Avec le protocole DSDV, chaque modification de la table de routage locale d'un nœud est aussitôt diffusée à l'ensemble de ses voisins. Les routes reçues par une diffusion seront aussi envoyées quand le récepteur procédera à l'envoi de ses paquets de routage. Sans oublier qu'il devra incrémenter les métriques des routes reçues avant l'envoi car il représente un nœud en plus.

L'unité mobile doit alors attendre la prochaine mise à jour initiée par la destination afin de mettre à jour l'entrée associée à celle-ci rendant ainsi le DSDV lent. De plus, DSDV utilise les mises à jour périodique et basées sur les événements causant un contrôle excessif au point de vue de la communication.

### Le protocole de routage WRP

Ce protocole est basé sur les algorithmes de recherche de chemin nommé PFA signifiant en anglais Path-Finding Algorithm. Afin d'éviter le problème de métrique de mesure infinie, les PFAs gardent en mémoire le nœud prédécesseur du chemin le plus court correspondant à chaque destination. Par contre, le problème des PFAs est l'apparition des boucles de routage temporaires dans le chemin spécifié du prédécesseur avant la convergence, c'est à dire lorsqu'ils auront une vue précise et cohérente de la nouvelle topologie du réseau. Pour résoudre ce problème, dans WRP [RT99], chaque nœud maintient :

une table de distance

une table de routage

une table des coûts des liens

une liste de retransmission de messages

La table de distance d'un nœud  $x$  est une matrice contenant pour chaque destination  $y$  et pour chaque voisin  $k$  de  $x$ , la distance  $D_{xyk}$  avec le prédécesseur  $P_{xyk}$  de  $k$ . La table de routage d'un nœud  $x$  est représentée par un vecteur dont chaque entrée est associée à une destination  $y$  connue. Une entrée doit comporter l'adresse de destination, la distance vers celle-ci ( $D_{xy}$ , du plus court chemin), le prédécesseur ( $P_{xy}$ ), le successeur ( $S_{xy}$ ) et une marque ( $\text{tag } xy$ ). Cette marque est utile lors de la mise à jour des tables de routage. En cas de boucle ( $\text{tag } xy = \text{error}$ ) ou en cas de destination non marquée ( $\text{tag } xy = \text{null}$ ) sinon ( $\text{tag } xy = \text{correct}$ ). Les mises à jour s'effectuent après réception des données de mise à jour d'un voisin ou lors d'un changement d'état de lien d'un voisin. Les voisins présents dans la MRL doivent acquitter après réception du message. Si la table de routage n'a pas été modifiée lors de la dernière mise à jour, un message « HELLO » est cependant

---

envoyé afin de s'assurer de la connectivité. Le protocole WRP vérifie ainsi sa connexion avec ses voisins à chaque fois qu'un changement d'état d'un lien est détecté. Le protocole WRP permet ainsi d'aider à l'élimination des boucles de routages mais aussi de minimiser le temps de convergence du protocole. Ce protocole est beaucoup utilisé dans la mobilité pour permettre la mise à jour des tables de routages par la découverte des voisins.

### **Le protocole de routage GSR**

GSR [CG98] est semblable à DSDV mais comporte quelques améliorations comme la suppression du mécanisme d'inondation des messages de routage et l'utilisation de la vue globale de la topologie du réseau comme les protocoles à états de liens. Les messages de routage sont générés suivant les changements d'états des liens. Lors de la réception d'un tel message, comme chaque nœud ajoute un identifiant qui lui est propre et qui est incrémenté à chaque transmission, afin de dater les messages, le nœud met à jour sa table de topologie uniquement si le message reçu porte un identifiant plus récent. Par la suite, le nœud reconstruit sa table de routage et diffuse les mises à jour à ses voisins.

### **Le protocole de routage FSR**

Ce protocole peut être considéré comme une évolution du protocole GSR. Cette évolution vise encore la diminution de la consommation de la bande passante. Pour cela, FSR [PGC00] utilise la technique dite de « l'œil du poisson » proposée par Kleinrock et Stevens. L'œil d'un poisson capture avec précision les points proches du point focal. La précision diminue quand la distance, séparant le point vu et le point focal, augmente. Cette technique de l'œil du poisson permet la réduction du volume d'informations nécessaire pour les données graphiques. Elle sous-entend une diminution du détail et de la précision plus la distance augmente. On définira la portée ou le champ de vision du poisson en nombre de sauts, plus un nœud est proche plus les données maintenues envers celui-ci seront plus précises. La réduction du volume des données de mise à jour est obtenue en utilisant des périodes d'échanges différentes pour les différentes entrées en fonction de leur distance. Les entrées qui correspondent aux nœuds les plus proches sont envoyées aux voisins avec une fréquence élevée (donc avec une période d'échange relativement petite). Ainsi un grand nombre de données de routage est évité, ce qui réduit le volume des messages qui circule sur le réseau. L'utilisation du protocole de routage « GSR » dans un réseau de grande taille entraîne des messages de mise à jour pouvant consommer de façon importante la bande passante. Avec « FSR » ce volume peut être réduit sans toucher à la consistance et la précision des données.

### **Le protocole de routage OLSR**

OLSR [CJA<sup>+</sup>03] est le résultat du travail de l'équipe d'HIPERCOM. Au niveau de ce protocole, les nœuds ne déclarent qu'une sous-partie de leur voisinage grâce à la technique des relais multipoints. Cette technique permet d'optimiser la diffusion des messages de routage économisant une grande partie de la bande passante du réseau. Les relais multipoints consistent essentiellement, en un nœud donné, à ignorer un ensemble de liens et de voisins directs, qui sont redondant pour le calcul des routes de plus courts chemins. Plus



précisément, dans l'ensemble des voisins d'un nœud, seul un sous-ensemble de ces voisins est considéré comme pertinent. Il est choisi de façon à pouvoir atteindre tout le voisinage à deux sauts (tous les voisins des voisins), cet ensemble est appelé l'ensemble des relais multipoints. Ces relais multipoints sont utilisés pour diminuer le trafic dû à la diffusion des messages de contrôle dans le réseau, et aussi pour diminuer le nombre de retransmission à tout le réseau puisque les routes sont construites à base des relais multipoints

### **3.4.2 Le routage réactif**

Le routage réactif est basé sur le principe de l'ouverture de route à la demande. Ainsi lorsqu'un équipement veut communiquer avec une station distante, il est obligé de déterminer une route dynamiquement. Cette technique permet de ne pas inonder le réseau de paquets de contrôle de routage et de ne conserver que les routes utilisées. Lorsque le réseau à besoin d'une nouvelle route, un processus de découverte de route est alors lancé, ce processus dépend des différents protocoles de routages réactifs. Il existe plusieurs protocoles de découvertes de routes, le backward learning et le routage source sont les plus couramment utilisés. Le backward learning consiste à ce que la source diffuse par inondation une requête de recherche de route vers une destination. Lors de la réception d'une telle requête, les nœuds intermédiaires complètent leurs tables de routage. Une fois la destination atteinte, celle-ci peut envoyer une réponse en utilisant le chemin tracé par la requête. Évidemment si un chemin vers la destination est connu par un des nœuds intermédiaires, celui-ci est utilisé réduisant ainsi le temps et le travail nécessaire à l'établissement d'une route. Le routage réactif induit une lenteur lors de l'ouverture d'une nouvelle connexion, et on ne peut prévoir à l'avance la qualité de la route découverte (délai, bande passante, etc.). Cependant ces inconvénients sont mineurs par rapport au gain apporté à la bande passante, non saturé par des paquets de contrôle de routage. Actuellement ils existent plusieurs protocoles réactifs (DSR, AODB, TORA, ABR, SSR, LAR, RDMAR).

#### **Le protocole de routage DSR**

Le protocole de routage DSR [JM96], qui signifie Dynamic Source Routing, utilise la technique du routage source. Le routage source consiste à ce que la source détermine un chemin et envoie dans chaque paquet de données tous les nœuds à traverser pour atteindre la destination. Chaque nœud intermédiaire retire son adresse du paquet avant de le retransmettre. Cette technique nécessite la connaissance de la route à utiliser de la part de la source. Cette connaissance des routes est obtenue par une table de routage maintenue dans chaque nœud. Il faut donc dans un premier temps découvrir les routes, puis les conserver tant qu'elles existent. Pour établir ces routes, chaque nœud peut initier une découverte dynamique de route. Pour cela le nœud qui lance une telle procédure va inonder le réseau d'une requête découverte de route qui identifie la source. Si la requête parvient à destination, celle-ci renvoie le paquet à la source. Le paquet contient la liste des nœuds à traverser pour l'atteindre. En plus de l'adresse de la source le paquet contient la liste de tous les nœuds jusqu'à présent visité, ainsi chaque nœud qui reçoit le paquet peut dresser à partir de celui-ci une table de routage qu'il pourra par la suite utiliser. Chaque paquet de requête de route contient un identificateur unique permettant de détecter les

duplications de ce paquet. Chaque nœud du réseau maintient ainsi une liste de couple « adresse de l'initiateur, identificateur de requête » des requêtes reçues, chaque entrée de la liste possède un temps de vie limité. Les protocoles de routages pro actifs maintiennent les chemins découverts en échangeant périodiquement des informations de mise à jour de leur table de routage, ce n'est pas le cas de ce protocole. Le protocole DSR maintient une route et l'utilise jusqu'à ce qu'un nœud du chemin détecte une erreur de transmission, cette erreur est détectée par la couche de liaison de données. Lorsque cela se produit, le nœud qui a découvert l'erreur de transmission envoie un message à la source indiquant que le chemin n'est plus valide après lui. Ainsi la source peut adapter sa table de routage et doit relancer une nouvelle requête de découverte de route. L'avantage d'un tel protocole réside dans le fait que les nœuds intermédiaires n'ont pas à maintenir la table de routage pour les paquets qu'ils reçoivent étant donné que ces derniers possèdent déjà toutes les décisions de routages. De plus on évite les boucles de routage. Cependant le chemin utilisé n'est pas forcément optimum, et la découverte d'un chemin demande du temps.

### **Le protocole de routage AODV**

Le protocole de routage AODV [CBR04] est une amélioration du protocole DSDV. Cette amélioration consiste à ne plus maintenir l'ensemble des routes mais uniquement celle dont on a besoin. Il conserve cependant les principes de numéros de séquences et de routage nœud par nœud. Les numéros de séquences permettant de dater une route, et d'utiliser ainsi la plus récente.

Cette notion est importante dans un réseau de capteurs mobiles qui est supposé être très dynamique. La découverte des routes suit le même principe que DSR, cependant les informations stockées au niveau de chaque nœud au sein des tables de routages diffèrent. De plus la maintenance des routes n'est pas gérée de la même manière. Au niveau de la découverte des routes, AODV conserve sur chaque nœud de transit des informations sur la route découverte, les tables de routages AODV contiennent :

- l'adresse de destination
- le nœud suivant
- la distance en nombre de nœuds à traverser
- le numéro de séquence de destination
- le temps d'expiration de l'entrée de la table.

Tout comme DSR, AODV ne permet pas de décider du chemin optimum, cependant il évite lui aussi les boucles de routage.

### **Le protocole de routage TORA**

TORA [BMJ<sup>+</sup>98] s'attaque aux problèmes d'économie de la bande passante en tentant de minimiser l'effet des fréquents changements de la topologie, particularité des réseaux de capteurs due à la mobilité des nœuds. Afin d'y parvenir, la recherche du meilleur chemin est délaissée non pas en terme de calcul mais en terme de procédure. De cette manière un protocole pourra choisir un plus long chemin entre la source et le nœud destination dans le but d'éviter le processus coûteux de découverte de nouveau voisin. De plus, TORA conserve plusieurs chemins vers une même destination et non plus seulement le meilleur

chemin ce qui a pour conséquence de limiter les effets induits par une modification de la topologie sur le routage des données. Le protocole est aussi caractérisé par la limitation des messages de contrôle à l'ensemble des nœuds proches de l'évènement. Le protocole AODV est un protocole réactif qui n'agit qu'en cas de demande d'établissement de route. Le trafic de contrôle augmente avec la mobilité des nœuds. Cette augmentation est due à la fréquence des liaisons cassées provoquant l'émission des paquets RERR. Lorsque la vitesse de déplacement des nœuds augmente considérablement, [HBT<sup>+</sup>03] observe que le trafic de contrôle devient de plus en plus indépendant de cette augmentation de mobilité et tend vers une valeur presque constante. Ceci montre que le protocole AODV est bien adapté aux changements fréquents de la topologie du réseau.

### 3.4.3 Le routage hybride

Le modèle hybride apparaît comme un bon compromis qui d'un côté utilise une procédure de détermination sur demande et de l'autre un coût de recherche limité.

Il existe plusieurs protocoles hybrides tels que le protocole de routage « ZRP » et « CBRP » que nous allons expliciter.

#### Le protocole de routage ZRP

Le protocole de routage ZRP [HPS02] est un modèle hybride entre un schéma pro actif et un schéma réactif. Il est basé sur deux procédures : le protocole de routage intrazone, nommé IARP et le protocole de routage interzone, nommé IERP.

IARP est utilisé uniquement à l'intérieur de la zone de routage. Cette zone est définie pour chaque nœud et comporte une taille de rayon correspondant à une valeur de nombre de saut. Par exemple, pour un nœud, si cette valeur vaut « deux » alors tous les nœuds ayant une distance supérieure à deux sauts ne feront pas parti de la zone de ce nœud. Les nœuds se trouvant à une distance de deux nœuds seront les nœuds périphériques à ce nœud. La valeur doit être fixée par l'administrateur réseau et est équivalente pour chaque nœud du réseau. Elle est importante car elle détermine la performance du protocole ZRP. Plus un réseau est instable plus il est essentiel que la valeur soit faible. IARP est généralement implémenté par des protocoles pro actifs variés mais comme cela n'est pas vraiment spécifié, il peut l'être à partir de différents protocoles comme des dérivés de protocoles à vecteur de distance. Le protocole doit de toute façon déterminer la distance qui sépare le nœud des autres nœuds du réseau afin qu'il délimite sa zone de routage. Cependant quel que soit le choix, le protocole nécessite d'être modifié afin de le restreindre à la zone de routage du nœud ce qui limite ainsi les mises à jour. Le contenu de la zone est connu par le nœud mais en contre partie il n'a aucune information concernant les autres nœuds appartenant à l'interzone.

Grâce à ce partage entre intrazone et interzone, les changements de topologies du réseau ont uniquement un impact local et n'est plus répercuté à l'autre bout du réseau ce qui réduit l'utilisation de la bande passante du réseau.

IERP est responsable d'établir des liens avec les nœuds situés dans l'interzone. Pour cela, il s'appuie sur les techniques de bordercasting (via BRP : Protocole de Résolution Bordercast) qui permet d'envoyer un paquet à tous les nœuds périphériques. Lors d'une

demande de route, IERP vérifie tout d'abord que le destinataire ne soit pas présent dans l'intrazone. S'il est présent alors aucun processus de connexion n'est nécessaire. En revanche, si il ne s'y trouve pas alors la source fait une demande d'établissement de route « Route Request » à tous les nœuds périphériques. Les nœuds périphériques, à la réception du message, effectuent la même opération. Chaque nœud recevant la requête inscrit son identificateur à l'intérieur avant de la renvoyer, cela s'appelle le processus d'accumulation de route. Le nœud périphérique contenant la destination dans sa zone de routage lui répond, à l'aide des identificateurs présents dans la requête, par un signal « Route Reply » en lui indiquant le chemin à emprunter pour l'atteindre. Comme il est préférable qu'une requête de type « Route Request » ne soit pas retransmise à une zone qui a été déjà parcourue, IERP utilise deux mécanismes. Le premier tue les messages qui contiennent un identificateur d'un nœud présent dans son intrazone. Le suivant est un mécanisme complémentaire qui enregistre l'identificateur de l'hôte dans sa liste de requête uniquement en période de « Route Request » afin d'ignorer une requête déjà formulée auparavant. IERP dispose également d'un mécanisme de réponse réactive aux erreurs de route lorsque le saut suivant est déterminé comme inaccessible. Un paquet « Route Failure » est alors envoyé à la source pour l'avertir et la voie de communication ayant expiré est retirée de la table de routage interzone. IERP peut être configuré afin de réparer localement la route interzone endommagée avec une procédure d'établissement de route vers le nœud inaccessible.

### Le protocole de routage CBRP

Dans CBRP [Jia99], le réseau est décomposé en groupes. Chaque groupe est constitué de nœuds membres et un représentant de groupe. Un nœud n'ayant pas de statut de membre, ni de représentant, active un timer et diffuse un message « HELLO ». Lorsqu'un représentant d'un groupe reçoit ce message, il envoie immédiatement une réponse à la source. À sa réception, le nœud peut alors devenir membre de ce groupe à condition que l'attente de la réponse n'ait pas dépassé le timeout. En cas de dépassement du timeout et si le nœud possède au moins un lien bidirectionnel vers un nœud voisin, le nœud peut se proclamer représentant d'un groupe. Sinon il reste dans l'état indécidé et répète la même procédure mais la dynamique des réseaux de capteurs fait que l'attente des nœuds indécidés soit souvent courte.

Les nœuds étant mobiles, il faut maintenir une table des voisins dont chaque entrée est associée à un voisin. Elle indique l'état des liens uni ou bidirectionnel et le statut du voisin. Le représentant ou CH d'un groupe regroupe les informations concernant les membres de son groupe et possède une table contenant les groupes adjacents. Une entrée de cette table est associée à un groupe voisin. La table des groupes adjacents contient l'identificateur du groupe, l'identificateur du nœud de son groupe permettant la liaison avec ce groupe, et l'identificateur du représentant de ce groupe.

La requête de demande de chemin s'effectue par inondation et est destinée uniquement aux représentants des groupes voisins. Un représentant de groupe recevant ce message vérifiera, en utilisant sa table de membres de groupes, l'existence du nœud de destination dans son groupe. Si il s'y trouve, il lui envoie directement la requête. Dans le cas contraire, il fait poursuivre le message aux autres représentants voisins en y inscrivant au préalable

son adresse. Un représentant pourra ainsi ignorer les requêtes déjà traitées. Quand la requête parvient au destinataire, celui-ci répond par l'envoi du chemin qui a été sauvegardé dans le paquet de la requête. Si le nœud source ne reçoit pas de réponse au bout d'une certaine période, il envoie de nouveau la requête.

Lorsqu'un nœud détecte un lien défaillant pendant l'acheminement des données, il retourne un message d'erreur à la source et applique un mécanisme de réparation locale. Ce mécanisme cherche si le nœud inaccessible ou le nœud après celui-ci dans le chemin peut-être atteint à travers un autre voisin. En cas de réussite, les données sont acheminées en utilisant le chemin réparé.



Tableau comparatif des différents protocoles dans les réseaux mobiles

Contrainte	AODV	DSR	CBRP	OLSR	GSR	DSDV	FSR	ZRP
Catégorie	Réactif	Réactif	Réactif	Pro actif	Pro actif	Pro actif	Pro actif	Hybride
Type de protocole	Vecteur de distance	Schéma à état de lien	Schéma à état de lien	Schéma à état de lien	Schéma à état de lien	Vecteur de distance	Schéma à état de lien	Indéfini
Routes maintenues	table de routage	table de routage	table de routage	table de routage	table de topologie	table de routage	table de routage	table de routage
Liberté de boucle	oui	oui	oui	oui	non	oui	oui	oui
Routes multiples	non	oui	non	oui	non	oui	oui	oui
Multicast	oui	non	oui	oui	oui	oui	oui	oui
Surcharge	modéré	modéré	minimale	minimale	modéré	modéré	modéré	modéré
Diffusion	possible	possible	possible	possible	pas possible	possible	possible	possible
Méthodes de reconfiguration des routes	suppression des routes et notification à la source	message d'erreur et suppression du chemin sauvegardé	messages de contrôle envoyés en avance afin d'augmenter la réactivité	messages de contrôle envoyés en avance afin d'augmenter la réactivité	paquets de mise à jour	des paquets de mise à jour et numéro de séquence	échanges périodiques	dépend du protocole réactif utilisé
avantages et inconvénients	adapté au topologie dynamique mais ne supporte pas des routes multiples	pas de stockage des informations sur les routes, la découverte d'un chemin demande du temps.	réduit la communication mais problèmes de boucle de routage	diffusion optimisée des messages de contrôle.	Routes multiples, pas de stockage des informations sur les routes, demande un grand délai	simple, absence de problème de boucle, consomme beaucoup l'énergie lors des demandes de mise à jour	réduit le nombre de paquets de mise à jour mais non optimal des routes vers les destinations	réduit la charge de communication, découverts rapides des routes, chevauement des zones entre les nœuds

### 3.4.4 L'adaptation à la métrique énergie

Nous présentons quatre protocoles d'adaptation au niveau routage qui reposent sur un calcul de métrique prenant en compte l'état de l'énergie disponible sur un nœud.

[Toh01] calcule sa métrique à partir de l'information locale au nœud : sa puissance disponible. « Min-Max Battery Cost Routing » utilise aussi la même information mais nécessite une vue globale pour faire son adaptation.

Le choix d'une route se fait en ayant la connaissance de l'énergie disponible sur tous les nœuds. « Maximizing network lifetime for reliable routing in wireless environments » utilise une métrique dont le calcul repose sur deux informations : la puissance disponible mais également l'état du canal. C'est l'état du canal qui permet de calculer l'énergie nécessaire à la transmission. La métrique est alors le ratio puissance disponible sur puissance nécessaire.

#### Minimum Battery Cost Routing (MBCR)

[BM02, Toh01] décrit un protocole de routage qui calcule la somme de la puissance résiduelle des nœuds mobiles avant de choisir la route avec la somme maximale de puissance résiduelle. La métrique d'adaptation est la capacité résiduelle disponible sur un nœud. Pour le choix d'une route d'un nœud  $n$  à un instant  $t$ , les auteurs utilisent les fonctions suivantes :

$$f_n(C_n^t) = \frac{1}{C_n^t} \quad (3.3)$$

La valeur de la fonction de coût pour le nœud  $n$  augmente avec la diminution de capacité de la pile. Le coût de pile  $R_j$  pour la route  $n$ , comprenant  $D$  nœuds est

$$R_j = \sum_{n=0}^{D_j-1} f_n(C_n^t)$$

$$R_n = \min\{R_j \mid j \in A\} \quad (A \text{ est l'ensemble des routes possibles})$$

La route choisie est celle qui a un coût  $R_n$  minimal parmi toutes les routes possibles. Avec ce critère de sélection la route choisie peut passer par des équipements ayant beaucoup de batteries alors que certains équipements auront une batterie très faible qui sera épuisée par les premiers relayages.

#### Min-Max Battery Cost Routing (MMBCR)

MMBCR [BM02, Toh01] utilise une métrique équivalente à celle de MBCR mais propose un critère différent pour choisir une route, afin d'éviter d'avoir une route qui va se rompre rapidement si certains des nœuds qui la composent ont des batteries faibles. Le coût d'une route est défini à partir de la capacité disponible minimale sur les nœuds qui la composent. La fonction de coût d'une route est modifiée comme suit :  $R_j = \max_{n \in \text{route } j} f_n(C_n^t)$  et le critère de choix de la meilleure route est également modifié par  $R_n = \min\{R_j \mid j \in A\}$  La durée de vie de la route ayant le coût de batterie le plus important est plus petite que celle de toutes les autres routes. Elle est déterminée par le premier nœud de la route qui épuise sa batterie. L'objectif est alors d'éviter la route avec les nœuds ayant le moins de capacité parmi tous les nœuds dans toutes les routes possibles. Bien qu'il ne puisse pas être certain que la consommation totale de puissance soit optimisée, la vie de la route est notablement augmentée par rapport au cas précédent.



### MRPC (Maximizing network lifetime for reliable routing in wireless environments)

[MB02] définit une autre métrique que l'état de l'énergie disponible. Elle prend en compte l'énergie qui sera dépensée pour émettre un paquet sur un lien spécifique. MRPC choisit le chemin qui a la plus grande capacité à émettre de futurs paquets au nœud "critique" : le nœud qui a la plus petite capacité résiduelle de transmission de paquet. La formulation de la capacité est :  $C_{n,j} = \frac{B_n}{E_{n,j}}$  Où  $B_n$  est la puissance résiduelle de pile dans le nœud  $n$  et  $E_{n,j}$  est l'énergie nécessaire à la transmission entre le nœud  $n$  et le nœud  $j$ . Notons que cette expression rend compte du coût de transmission lié à la distance entre les nœuds, c'est-à-dire du bruit de la liaison. Le fonctionnement de l'algorithme est alors le même que MMBCR, sauf qu'il maximise le minimum du  $C_{n,j}$ . La distinction est simplement liée à une formulation inverse de la métrique (c'est-à-d. avec  $1/C_{n,j}$ , l'algorithme serait identique). De cette façon il maximise la vie du réseau.

$$Life = \min_{(n,j \in P)} \{ C_{n,j} \}$$

$$P_{candidate} = \operatorname{argmax} \{ Life_p \mid P \in \text{toutes les routes possibles} \}$$

### e-OLSR [SR02, MM08]

L'efficacité énergétique du routage dans un réseau mobile dépend de l'exactitude du niveau d'énergie instantané. Les contraintes de bande passante, de coûts de communication, de taux de perte élevés et de dynamique de la topologie des réseaux mobiles, complexifie considérablement ses collectes et mise à jour. L'erreur relative correspond à la différence moyenne entre les niveaux d'énergie perçus (par le nœud qui prend la décision du routage) et réel (connue suite au routage). Elle augmente avec le débit. Quoique le paramétrage de base du protocole OLSR ne permette pas de la corriger, différentes techniques sont proposées et la réduisent significativement. Notamment, la prédiction intelligente rend un niveau d'énergie précis quelque soit la densité du trafic [SR02, MM08]. [BMTM04] propose d'autres mécanismes d'optimisation énergétique dans les réseaux mobiles utilisant OLSR. Lorsque par exemple 2 nœuds arrivent ex-æquo dans l'algorithme de choix des MPR, le fait de choisir celui qui a le plus d'énergie augmente l'autonomie globale du réseau mobile.

## 3.5 Conclusion

Ce chapitre a été axé sur le concept de la mobilité dans les réseaux de capteurs mobiles en préservant l'énergie pour la continuité du fonctionnement du réseau. Il a présenté les différents types de mobilité et les protocoles de routage de la mobilité. Cependant les environnements mobiles sont caractérisés par de fréquentes déconnexions et des restrictions sur les ressources utilisées, surtout si tous les capteurs du système sont mobiles. Les stratégies de routage doivent prendre en compte les changements fréquents de la topologie, de la consommation de la bande passante, de la radio ... Ces limitations transforment certains problèmes, ayant des solutions évidentes dans l'environnement classique, en des problèmes complexes et difficiles à résoudre. Parmi ces problèmes figure le problème de la déconnexion que nous allons aborder dans la partie 2 en apportant notre contribution pour maintenir la connectivité en cas de mobilité des capteurs. Assurer la connexion de

tous les nœuds d'un réseau de capteurs mobiles est un problème très complexe vu la dynamique et l'évolution rapide de la topologie. En effet les unités mobiles sont dynamiquement et arbitrairement éparpillées d'une manière où l'interconnexion peut changer à tout moment. Nous tenterons d'apporter une réponse avec notre contribution dans la partie 2 du document.



# Deuxième partie

## Contribution



La partie contribution est la deuxième partie de ce travail qui présente notre apport pour l'amélioration de la gestion de l'énergie et de la connectivité dans les réseaux de capteurs. Cette partie s'appuie sur la première partie décrivant un état de l'art sur la gestion de l'énergie pour proposer des solutions d'amélioration en utilisant des algorithmes permettant de réduire la consommation d'énergie pour la formation des clusters et aussi pour le choix des clusters heads. On propose également des solutions pouvant améliorer le déplacement, la mobilité des capteurs en se basant sur le comportement des chauves souris en déplacement groupées et en quête de nourriture. Tous ces algorithmes sont complétés par des tests de simulations faisant ressortir l'évolution de la consommation d'énergie par rapport à d'autres méthodes et algorithmes dans la littérature.



# Chapitre 4

## Un algorithme de routage hybride (SHE) pour les réseaux de capteurs statiques

### Sommaire

---

<b>4.1</b>	<b>Motivation</b>	<b>66</b>
4.1.1	Choix des Cluster Heads (CH)	66
4.1.2	Pannes et conséquences	66
4.1.3	Position de la station de base	67
<b>4.2</b>	<b>Solution proposée et hypothèses</b>	<b>67</b>
<b>4.3</b>	<b>Principes de notre algorithme</b>	<b>67</b>
<b>4.4</b>	<b>Notre algorithme Hybride</b>	<b>68</b>
4.4.1	Les idées fondatrices	68
4.4.2	Les algorithmes mis en œuvre	70
<b>4.5</b>	<b>Évaluation de l'algorithme hybride proposé</b>	<b>73</b>
4.5.1	Présentation de la plateforme matérielle	74
4.5.2	Métriques	75
4.5.3	Résultats	75
4.5.4	Précisions et observations diverses	83
<b>4.6</b>	<b>Expérimentation avec la plateforme SensLAB</b>	<b>83</b>
4.6.1	Introduction	83
4.6.2	Nœud SensLAB	84
4.6.3	Expérimentation	87
<b>4.7</b>	<b>Conclusion</b>	<b>90</b>

---

Une des contraintes majeure des RCSF est l'énergie. Les capteurs étant dotés de piles et déployés généralement dans des zones où il est quasiment impossible de changer ou de recharger leurs piles, la durée de vie est donc liée à la manière de consommer cette énergie.



Le module radio est le module le plus consommateur en énergie. Il est donc nécessaire de minimiser l'utilisation de la radio et le nombre de bits à transmettre dans un RCSF. Dans ce domaine, les algorithmes de clustering ont montré leurs preuves dans la réduction de la consommation d'énergie.

Nous proposons dans ce chapitre un nouvel algorithme "hybride" permettant à la fois de construire des clusters et d'évaluer chaque nœud capteur participant au réseau dans le but d'augmenter la durée de vie du réseau de capteurs tout en préservant la connectivité.

## 4.1 Motivation

Dans cette partie, nous présentons les problèmes et les solutions envisagées pour justifier la mise en place de notre solution. Notre algorithme permet de définir un certain nombre de zones qui vont nous permettre d'économiser l'énergie dépensée avec la construction des clusters. Notre algorithme permet aussi une communication efficace des données en fonction de l'énergie restante des nœuds.

### 4.1.1 Choix des Cluster Heads (CH)

Le premier problème vient du fait de la surconsommation de l'énergie au moment de l'élection des CHs. Par sa fonction, le nœud CH consomme plus d'énergie que les autres nœuds du réseau. En effet le CH coordonne le fonctionnement des nœuds membres de son cluster et agrège leurs données. De ce fait, il dissipe plus d'énergie créant un déséquilibre de la distribution de l'énergie sur le réseau. C'est le cas dans les algorithmes de clustering comme LEACH ou HEED. Ils sont basés sur l'élection d'un ensemble de CHs parmi les capteurs du réseau. LEACH propose une rotation périodique du rôle du CH. Cette rotation a ses limites créant une surconsommation pour la diffusion et ne garantissant pas une distribution homogène des CHs sur le réseau car le critère d'élection du CH est une probabilité aléatoire. D'autres auteurs proposent une collaboration entre capteurs pour désigner les CHs ayant une capacité d'énergie suffisante pour ce rôle de CH [GVSOTCBA09]. Cette collaboration génère des communications qui augmentent la consommation d'énergie.

Le choix des CHs et la formation des clusters sont des problèmes NP-complets [GD90].

### 4.1.2 Pannes et conséquences

Le deuxième problème est celui de la défaillance d'un nœud. En effet les nœuds capteurs sont fragiles. Ils risquent de tomber en panne à cause de l'épuisement des batteries ou de la destruction par des événements externes. Ils peuvent capturer et communiquer des données erronées en raison de l'influence de l'environnement sur les composants de détection. Ceci est très dangereux par exemple dans le suivi d'un malade dans le domaine médical ou dans le suivi des ennemis en temps de guerre. En effet, au cours du fonctionnement du réseau, les nœuds épuisent doucement leur énergie. Ils en perdent jusqu'à un niveau où ils ne sont plus utiles au réseau. Si leur énergie n'est pas totalement épuisée, ils génèrent une source de consommation d'énergie supplémentaire. Ils captent des données mais ne

peuvent pas les envoyer. Ils alertent les nœuds voisins ou le CH qui restent éveillés pour la réception de ses données. La fiabilité et la sécurité de l'information dépendent alors de la quantité d'énergie restante des nœuds.

### 4.1.3 Position de la station de base

Le troisième problème réside dans le fait que la position de la station de base influe sur la consommation d'énergie des capteurs. Il faut donc trouver une bonne position pour une rentabilité optimum.

## 4.2 Solution proposée et hypothèses

Pour pallier les problèmes cités ci-dessus :

Il est souhaitable d'impliquer dans la vie du réseau, les nœuds qui ont une capacité d'énergie supérieure à un certain seuil pour éviter les mesures inutiles. Ce seuil doit permettre aussi aux capteurs de ne pas envoyer des données erronées dues à une baisse d'énergie. Il peut permettre une meilleure organisation du temps de sommeil pour les nœuds qui entourent le nœud défaillant.

Il faut réduire la surconsommation d'énergie pour la construction des clusters et la désignation des CH.

Notre réseau de capteurs sera composé d'un grand nombre de nœuds à communication sans fil. Ces capteurs seront dispersés dans un champ de captage rectangulaire. Les tâches d'acquisition de données sont déclenchées périodiquement ou à la demande par la station de base. Nous allons considérer que :

- les capteurs sont stationnaires et uniformément répartis ;
- les capteurs ont les mêmes capacités de capture, de traitement et de communication ;
- les capteurs utilisent une seule antenne ;
- les capteurs se rendent compte de leur voisin direct et peuvent mesurer des distances par rapport à leurs voisins ;
- la station de base est externe au champ de captage ;
- lorsque la clusterisation est terminée à l'initialisation, chaque capteur appartient à un seul cluster ;
- la procédure de clustering est distribuée.

## 4.3 Principes de notre algorithme

Rappelons que la communication est l'une des principales causes de la consommation élevée de l'énergie dans un réseau de capteurs. Elle est très utilisée pour la mise en place du réseau en grappe et aussi pour le choix des têtes de grappes.

L'objectif de notre algorithme est de réduire l'utilisation de la radio et de permettre une autoévaluation de chaque nœud dans son rôle pour réduire la surconsommation de l'énergie.

Pour faire face à ces pertes d'énergie et pour donner une longue durée de vie au réseau, notre solution se base sur les principes suivants :

Diviser symétriquement le terrain en plusieurs zones par la station de base pour permettre une mise en place facile des grappes et économiser l'énergie résultante de la communication de beaucoup de nœuds dans la formation des grappes. La station de base désignera les premiers CHs.

Éliminer les nœuds inutiles dans le fonctionnement du réseau en proposant des seuils de communication pour chaque grappe et dans chacune des zones ou clusters. En fonction des seuils, les capteurs doivent communiquer soit avec le CH, soit avec leur voisin ou s'éteindre dans le pire des cas. En effet un nœud en manque d'énergie peut transmettre des données fausses mais il peut provoquer aussi une situation telle que son voisin sera en attente de données qui n'arriveront jamais.

L'utilisation des seuils permet de diagnostiquer les nœuds et permet aussi d'économiser de l'énergie pour donner une longue vie au réseau en éliminant les nœuds susceptibles de nuire au bon fonctionnement du réseau.

Permettre aux CHs initialement désignés par la station de base de nommer leur successeur à ce rôle en se basant sur la quantité d'énergie envoyée par les capteurs du cluster.

Les CHs auront à jauger leur capacité par rapport à la moyenne des quantités des capteurs participant aux envois des données avant de décider de poursuivre leur fonction. On peut se limiter aux seuls capteurs qui participent aux envois. Les CHs perdront leur rôle au profit du nœud ayant une capacité d'énergie élevée au niveau du cluster (par rapport à la moyenne) et qui peut être proche de l'ancien CH.

Éviter les demandes de niveau d'énergie des capteurs par la station de base ou par les CHs. Chaque capteur va envoyer son état en même temps que les données captées dans le champ de captage, ce qui permettra d'utiliser la radio pour la communication dans le but de construire de nouveaux clusters ou pour désigner le changement des CHs.

## 4.4 Notre algorithme Hybride

La formation de clusters peut être effectuée d'une manière centralisée par la station de base ou d'une manière distribuée par les CHs. Pour obtenir de meilleurs résultats en terme de répartition équilibrée des nœuds entre les clusters, on a choisi une approche hybride (Centralisée et distribuée). L'approche centralisée est utilisée pour la mise en place des clusters par la station de base et l'approche distribuée après la mise en place des premiers clusters.

### 4.4.1 Les idées fondatrices

On utilise l'approche centralisée proposée dans le protocole LEACH-C à l'initialisation où chaque nœud envoie un paquet de données à la station de base (BS) contenant l'identificateur du nœud, la réserve d'énergie et sa localisation sur le réseau (en utilisant par

exemple le système de localisation GPS). La station de base va exécuter un algorithme d'optimisation afin de former les clusters.

L'algorithme de détermination du cluster optimal est un algorithme d'approximation NP complexe (NP-HARD), comme par exemple l'algorithme de recherche TABOU, ou l'algorithme de recuit simulé (simulated annealing), ou encore l'algorithme de colonie de fourmis, qui sont des métaheuristiques et ne nous donnent pas des résultats exacts mais proches de l'optimal.

Dans notre algorithme hybride, nous avons opté pour le même algorithme d'optimisation utilisé dans LEACH-C à savoir le recuit simulé. Cet algorithme a été mis au point en 1982 par S. Kirkpatrick dans lequel le choix des solutions est basé sur une loi de probabilité (distribution de Boltzmann) qui mesure la probabilité  $P(X)$  de visiter l'état  $X$  en fonction de son énergie  $E(X)$  et de la température  $T$ .  $N(T)$  est un facteur de normalisation.

$$P(X) = \frac{\exp\left(\frac{-E(X)}{KT}\right)}{N(T)} \quad (4.1)$$

$K$  : est la constante de Boltzmann.

Cette technique est utilisée depuis longtemps par les métallurgistes. On part d'une configuration connue, et on fait subir au système une modification élémentaire. Si cette modification diminue la fonction objectif, elle est acceptée sinon elle est acceptée avec la probabilité définie par l'équation 4.1.

Donc, cette méthode de formation de Clusters centralisée permet de déterminer, à partir de la position exacte des nœuds, la configuration optimale pour minimiser l'énergie dépensée. Dès que les clusters sont formés, la station de base va passer à l'élection de CH. Ces derniers sont choisis d'une manière très simple où seul le nœud qui a la plus grande réserve d'énergie est éligible pour devenir le prochain cluster head. On peut améliorer le choix du CH en ajoutant des critères de choix supplémentaires comme par exemple la position du CH par rapport aux nœuds du cluster ou de la station de base, c'est-à-dire choisir comme CH le nœud qui a la position la plus proche de tous les nœuds à partir desquels il va recevoir les données.

Après la phase d'initialisation, le choix du CH suivant est transféré au CH courant en fonction de l'énergie des nœuds participants au cluster et par rapport à sa position.

La construction de notre réseau utilise la station de base à l'initialisation. La station de base divise le champ de captage en plusieurs zones aléatoirement (algorithme 1). Puis, elle désigne les Clusters Head (CHs) de la zone entre les nœuds voisins (algorithme 1).

Par la suite, On calcule un minimum d'énergie et un maximum d'énergie à consommer par l'envoi d'un message dans chaque cluster. A cet effet, le calcul de l'énergie minimale est dérivé de la distance minimale en mètre du nœud le plus proche du CH et le calcul de l'énergie maximale est dérivé de la distance maximale du nœud le plus éloigné du CH (algorithme 2).

Nous utilisons les formules utilisées dans [ASSC02a] pour le calcul d'énergie. Cette idée permet aux nœuds de s'auto-évaluer pour la stabilité, l'économie d'énergie et pour augmenter la durée de vie du réseau.

Ces calculs sont effectués par chaque CH et sont communiqués aux nœuds pour leur évaluation (algorithme 2).

La seconde idée est de relayer, par les CH eux-mêmes, la station de base dans la désignation des CH. Cela va consister à joindre à chaque message capté la capacité d'énergie du capteur qui a capté l'information pour l'envoi au CH. Ainsi chaque CH pourra s'auto-évaluer par rapport à l'énergie moyenne des nœuds participant au cluster. En d'autres termes, en plus des messages à capter et à envoyer vers le CH, chaque nœud ajoute au message sa réserve d'énergie pour permettre au CH de mettre en place une stratégie efficace d'autoévaluation et de mise en place d'une nouvelle grappe en désignant son successeur. Les prochains CH sont choisis d'une manière très simple où seul le nœud qui a la plus grande réserve d'énergie est éligible pour devenir le prochain cluster head (algorithme 4). On peut améliorer le choix du CH en ajoutant des critères de choix supplémentaires comme par exemple la position du CH par rapport à la station de base ou par rapport au précédent CH, c'est-à-dire choisir comme CH le nœud qui a la position la plus proche par rapport à tous les nœuds à partir desquels il va recevoir les données

Quand un nœud doit envoyer des données au Cluster Head, il doit veiller à sa propre capacité énergétique pour éviter la perte de données et ne pas nuire à la stabilité du réseau (algorithme 3).

#### 4.4.2 Les algorithmes mis en œuvre

Notre algorithme s'appuie sur les principaux axes suivants : à l'initialisation, étant donné que les nœuds sont homogènes avec une même quantité d'énergie, les CHs sont désignés par la station de base.

Les CHs vont déterminer la distance minimale et maximale pour calculer l'énergie minimale et maximale pour l'envoi d'un message. A partir de ce calcul, chaque nœud peut s'auto-évaluer dans le cluster. Le capteur pourra soit atteindre son CH avec une quantité d'énergie supérieure ou égale à un seuil, ou soit orienter ses données captées vers son voisin par rapport à un autre seuil. Il pourra aussi éteindre son fonctionnement en dessous d'un seuil pour permettre au réseau une bonne poursuite de travail.

On effectue une compression des données au niveau de chaque nœud. En plus des messages à capter et à envoyer vers le CH, chaque nœud ajoute au message sa capacité d'énergie pour permettre au CH de mettre en place une stratégie efficace d'autoévaluation et de mise en place d'un nouveau cluster en désignant le prochain CH qui lui est proche avec une quantité suffisante et supérieure à la moyenne.

Les CHs qui reçoivent la capacité d'énergie des autres peuvent s'auto évaluer en comparant leur énergie à la moyenne des réserves d'énergie des nœuds de leur cluster.

#### Variables

Soit  $N$ , l'ensemble des noeuds,

Soit  $K$ , l'ensemble des zones du partitionnement,

Soit  $n$ , le nombre de noeuds,

soit  $k$ , le nombre de zones,

Soit  $n_i$ , le nombre de noeuds de la zone  $i$ ,

Soit  $N_i$ , l'ensemble des noeuds de la zone  $i$ ,

Soit  $Node_i$ , le noeud  $i$ ,

Soit SB la position de la station de base,  
 Soit CH la position du cluster head,  
 Soit ch le cluster head  
 Soit  $E_i$ , l'énergie du nœud i  
 Soit  $E_{ch}$ , l'énergie du cluster-head

### Algorithme exécuté par la station de Base

Pour faire face aux pertes d'énergie, notre solution divise le terrain en plusieurs zones par rapport à la taille du terrain (partage symétrique) ou par rapport à la concentration des nœuds. Les algorithmes probabilistes consomment beaucoup d'énergie dans la communication pour le choix du CH. Tous les capteurs du réseau s'échangent des messages concernant leur capacité d'énergie pour évaluer la force ou la faiblesse de chaque capteur dans le rôle du CH. En moyenne, on peut estimer  $n * (n - 1)$  communication [HCB00]. En effet, en considérant n nœuds, on aura n diffusions d'où n transmissions et n-1 réceptions. Ce qui est énorme en énergie pour la mise en place du réseau. La division en zones par la station de base réduit le nombre de communication par CH et par nœud. Ce qui permet un gain d'énergie en limitant la communication des capteurs à leurs zones ou à leur fonction de captage.

Nous commençons par attribuer un Id à chaque Nœud (AttribId()) puis nous créons les zones (Init(K)). La fonction  $voisin_i$  donne le nœud le plus proche du  $node_i$ .

---

#### Algorithm 1 : algorithme exécuté par la station de Base

---

```

AttribId(Node)
Init(K)
i=0;
Tant que  $i \leq k$  faire
  CH=distMin(SB, $N_i$ )
  Pour  $j = 1$  à  $n_i$  faire
    Envoi(ch, $Node_j$ , $ADV_{CH}$ )
  fin Pour
fin Tant que

```

---

Cet algorithme permet de trouver un nombre fixe de zones pour lesquelles la répartition par zone est efficace et augmente l'efficacité avec un CH par zone.

### Algorithme exécuté par chaque Cluster Head

Un nœud capteur utilise son énergie pour réaliser trois actions principales : l'acquisition, le traitement des données et la communication. La consommation d'énergie est fonction de la distance du capteur à son voisin et généralement au CH. Plus un nœud est distant du CH, plus il consomme l'énergie de la batterie. Les CHs vont déterminer la

distance minimale et maximale pour calculer l'énergie minimale et maximale pour l'envoi d'un message. Pour transmettre un message de  $k$  bits sur une distance de  $d$  mètres, l'émetteur consomme :

$$E^{trans}(\lambda) = \lambda \times (E_0^{trans} + \epsilon \times d^2) \quad (4.2)$$

$$E^{recv}(\lambda) = \lambda \times E_0^{recv} \quad (4.3)$$

$$E^{trans}(\lambda) + E^{recv}(\lambda) = \lambda \times (E_0^{trans} + \epsilon \times d^2) + \lambda \times E_0^{recv} \quad (4.4)$$

$E_0^{trans}$  : énergie de transmission réception électronique ;  $\lambda$  : taille d'un message ;  $d$  : distance entre l'émetteur et le récepteur ;  $E_0^{recv}$  : énergie d'amplification ;  $\epsilon$  : facteur d'amplification.

Le principe de RSSI [XLL<sup>+</sup>10] se base sur la relation entre la puissance émise et la puissance reçue des signaux pour estimer la distance entre les nœuds. Cette relation est représentée par l'équation 4.5.  $P_r$  est la puissance reçue des signaux sans fil,  $P_t$  est la puissance transmise de signal sans fil,  $d$  est la distance estimée en mètre entre les nœuds d'envoi et les nœuds de réception,  $n$  est le facteur de transmission dont la valeur dépend de l'environnement de propagation. [PBDT03] décrit en détail l'algorithme RSA pour estimer une distance

---

**Algorithm 2** Algorithme exécuté par CH

---

$$P_r = P_t * (1/d)^n \quad (4.5)$$

$$Nmin = disNmin(N_i, CH)$$

$$Nmax = disNmax(N_i, CH)$$

$$E_{min}(\lambda, Nmin) = \lambda \times (E_0^{trans} + \epsilon \times Nmin^2 + \times E_0^{recv}) \quad (4.6)$$

$$E_{max}(k, Nmax) = \lambda \times (E_0^{trans} + \epsilon \times Nmax^2 + \times E_0^{recv}) \quad (4.7)$$

$$\text{envoi}(ch, N_i, E_{min})$$

$$\text{envoi}(ch, N_i, E_{max})$$


---

Les deux seuils  $E_{min}$  et  $E_{max}$  sont calculés en fonction des équations 4.2, 4.3, 4.4. Dans le but d'éviter les collisions entre les capteurs de même cluster, nous utiliserons la technique de TDMA (Time Division Multiple Access) comme méthode d'accès au médium. Le capteur va utiliser la totalité de la bande passante à son slot de temps. Chaque cluster s'occupe de son modèle de communication qu'il crée et communique aux autres en même temps que les seuils. Ainsi les nœuds pourront passer à l'étape "endormi" durant les slots inactifs. On pourra, plus tard, envisager que la distance maximale permettant de calculer l'un des seuils dépende de la puissance de transmission. Dans notre situation, notre seuil est inférieur ou égal à la puissance de transmission. Le CH va envoyer ses mesures à chaque capteur du cluster pour s'auto-évaluer. Ainsi les autres capteurs auront à leur niveau la possibilité de mesurer leur rentabilité au cours du fonctionnement du réseau des capteurs.

**Algorithme exécuté par chaque nœud**

Pendant le fonctionnement, chaque capteur doit s'auto-évaluer pour éviter une utilisation inadéquate de son énergie et de celle des autres. Si leurs énergies ne sont pas totalement épuisées, ils deviennent une source de consommation d'énergie. En effet, ils captent des données et ne peuvent pas les envoyer. Ils alertent les nœuds voisins ou CH qui restent éveillés pour la réception de ses données.

**Algorithm 3** Algorithme exécuté par chaque nœud

---

```

Pour tout nœud  $u \in N_L$  faire
Si  $E_i < E_{min}$  alors
éteindre ( $node_i$ )
sinon
    Si ( $E_i \geq E_{max}$ ) alors
        envoi ( $node_i$ , CH,message)
    sinon
        envoi ( $node_i$ ,  $voisin_j$ , message)
    fin si
fin si
finpour

```

---

Etant donné que les CHs reçoivent les quantités d'énergie des capteurs de leur cluster, les CHs s'évaluent avec la moyenne des quantités d'énergie des nœuds du cluster. Ils pourront désigner leur successeur dès que leurs quantités d'énergie deviendront inférieures à la moyenne des quantités d'énergie des nœuds participant au cluster.

**Algorithm 4** Algorithme de changement de CH exécuté par chaque CH

$$E_M = \frac{\sum_{1 \leq i \leq n} (E_i)}{n} \quad (4.8)$$

```

Si  $E_{ch} < E_M$  alors
    Designer ch;
    envoi(ch,  $N_i$ , CH);
fin si

```

---

Cette technique est intéressante dans la mesure où elle tient compte de l'évolution du nombre de capteurs dans le cluster. Plus les capteurs disparaissent, plus le nombre de nœuds participant au cluster diminue permettant une mesure efficace par les CHs.

**4.5 Évaluation de l'algorithme hybride proposé**

Dans cette section, nous introduisons le dispositif utilisé pour l'évaluation de nos propositions et nous évaluons les performances de notre algorithme de clustering et de diagnostic.



### 4.5.1 Présentation de la plateforme matérielle

Nous avons effectué plusieurs simulations de notre algorithme en plaçant  $N$  nœuds capteurs dans une grille de 400x400 m. Nous avons mesuré certaines propriétés comme le nombre de zones créées et le nombre de CHs de l'algorithme hybride en utilisant les coordonnées radiales et rectangulaires dans le calcul des distances.

Notre protocole a été évalué au cours de plusieurs tests, qui nous ont permis d'observer comment il se comporte dans différentes conditions. On a imaginé quatre situations pour la position de la station de base : au bord du champ de captage et à des distances variables de la zone de captage [figure 4.1] . Ces tests ont été réalisés au sein du laboratoire DISC de l'Université de Franche-Comté à l'aide du simulateur de réseau SMAC (thèse Elgaber, A. (2014), compromis Sécurité/Énergie dans les réseaux de capteurs sans fil.). En utilisant ce

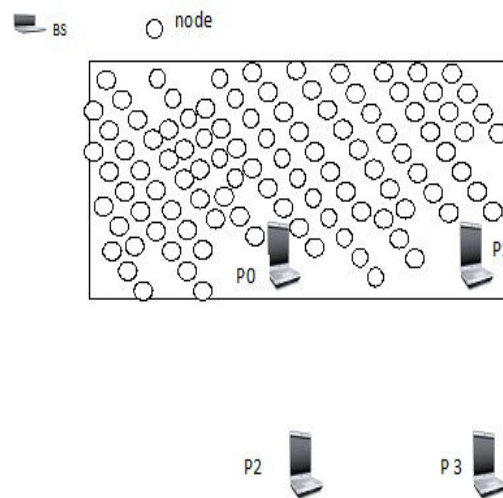


FIGURE 4.1 Illustration de la position de la station

simulateur, nous avons réalisé un certain nombre de simulations dans le but d'observer les performances de notre protocole sous différentes conditions. On place la station de base successivement aux points P0, P1, P2 et P3. Les paramètres utilisés lors de ces simulations ont été déterminés grâce à la littérature et à l'expérience du laboratoire dans les réseaux de capteurs. Dans notre simulation de l'algorithme, nous avons simulé l'algorithme de clustering LEACH, HEAD en utilisant le simulateur SMAC. Les capteurs sont déployés aléatoirement sur le terrain qui peut être segmenté en plusieurs zones. Dans l'objectif de réaliser l'essai de chaque hypothèse, des réseaux générés de façon aléatoire de tailles différentes sont utilisés. Tous les nœuds sont placés aléatoirement dans une zone carrée de 400 x 400 avec plusieurs positions pour la station de base. Les informations recueillies par les capteurs au sein de n'importe quel segment sont communiquées à la station de base (BS) par les Cluster Head. La station de base a une alimentation continue sur le secteur. Les paramètres de la simulation sont résumés dans le tableau 4.2 :

Nous avons considéré que la mesure de performance du réseau dans notre simulation est le nombre de messages transmis à la station de base pendant une durée de vie, qui est l'intervalle de temps depuis le début de l'opération sur le réseau de capteurs jusqu'à la

Notation	Description
$N = [100, 200, 300, 400]$	nombre de nœuds
$E_o = 0.5$ joule	énergie initiale d'un nœud
$E_0^{trans} = E_0^{recv} = 50$ nano-joule / bit	énergie de communication
$\epsilon = 100$ petajoule/bit/m <sup>2</sup>	énergie d'émission
$E_A = 5$ nano-joule / bit	énergie d'agrégation
$\lambda = 2000$ bits	taille d'un message
champ	400 X 400
Position du puits	P0, P1, P2 et P3

FIGURE 4.2 Les paramètres de simulation

mort du premier nœud dans le réseau. Dans les scénariis, la station de base est positionnée à P0 (x=200, y=0), P1 (x=400, y=0), P2 (x=200, y=-100) et P3 (x=400, y=-100), notre simulation nous a montré après plusieurs tentatives que notre algorithme est efficace pour trois zones avec le mode rectangulaire et huit zones avec le mode radial. Mais nous le détaillons un peu plus bas.

### 4.5.2 Métriques

Pour analyser les performances de notre protocole, nous avons opté pour les métriques suivantes :

Le nombre de clusters formés : cette métrique permet de vérifier notre hypothèse sur l'utilité d'utiliser des zones pour réduire le nombre des clusters générés, et déduire son impact sur la consommation énergétique.

Le nombre de nœuds participant au Clustering : plus le nombre de nœuds est élevé, meilleure est la technique utilisée.

Le nombre des paquets reçus par la station de base : cela permet d'avoir une idée sur la performance de notre algorithme pour organiser le réseau en zones.

### 4.5.3 Résultats

Dans cette partie, nous allons présenter les résultats de nos simulations. Nous avons utilisé des divisions rectangulaires ou radiales du terrain par rapport à la station de base comme le montre la figure 4.3. Chaque partie de la division constitue un cluster.

**Nombre de zones efficaces en P0 en utilisant la méthode radiale** La figure 4.4 montre les performances de l'algorithme de clustering en fonction du nombre de zones et en fonction de la variation du nombre de capteurs. Les différentes courbes atteignent leurs points culminants (0,54 message) lorsque le nombre de zones est égal à huit (8). Donc en utilisant la méthode radiale, l'efficacité recherchée consiste à utiliser un nombre de zones constant qui est égal à huit pour cette position.

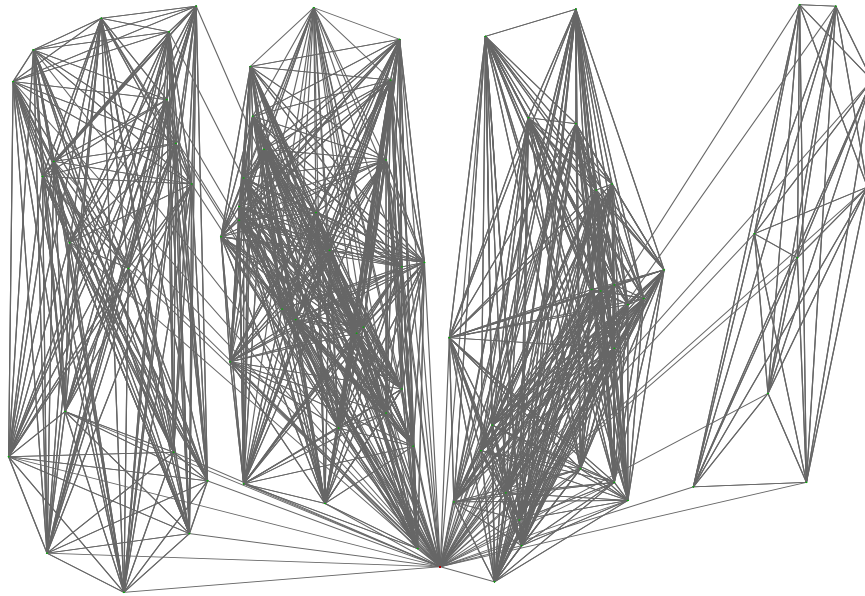


FIGURE 4.3 Exemple de division par zone (4 zones) à la position P1

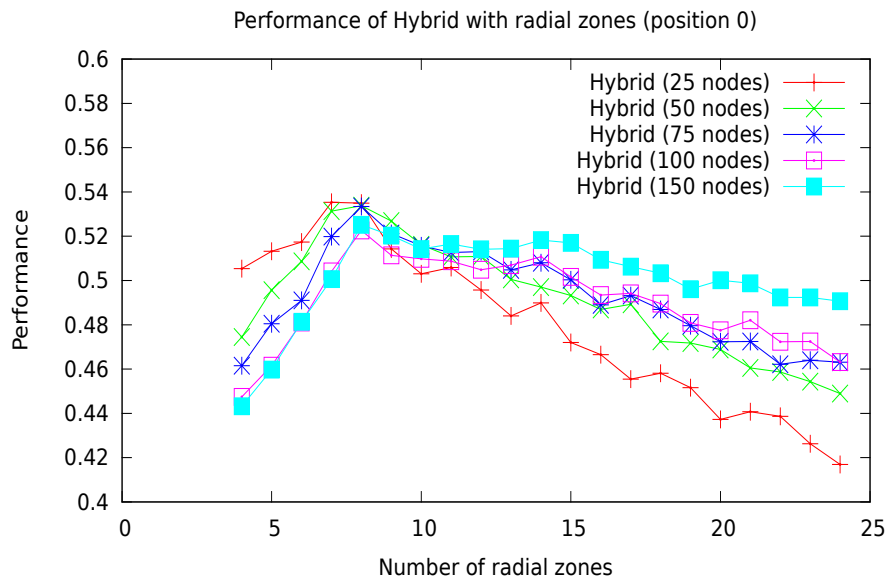


FIGURE 4.4 Performance par zone à la position P0

**Nombre de zones efficaces en P1 en utilisant la méthode radiale** Le schéma de la figure 4.5 montre les performances de notre algorithme de clustering à partir du nombre de zones et en fonction du nombre des capteurs. Après plusieurs simulations, ces courbes sont très variables du point de vue de la taille des zones et du nombre des capteurs. Mais, elles montrent un optimum vers huit (8) zones pour une meilleure rentabilité.

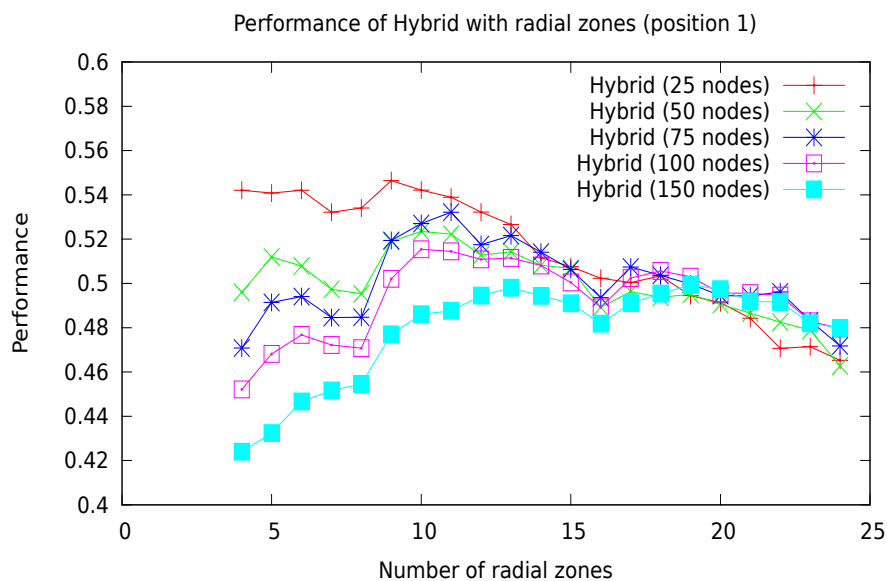


FIGURE 4.5 Performance par zone à la position P1

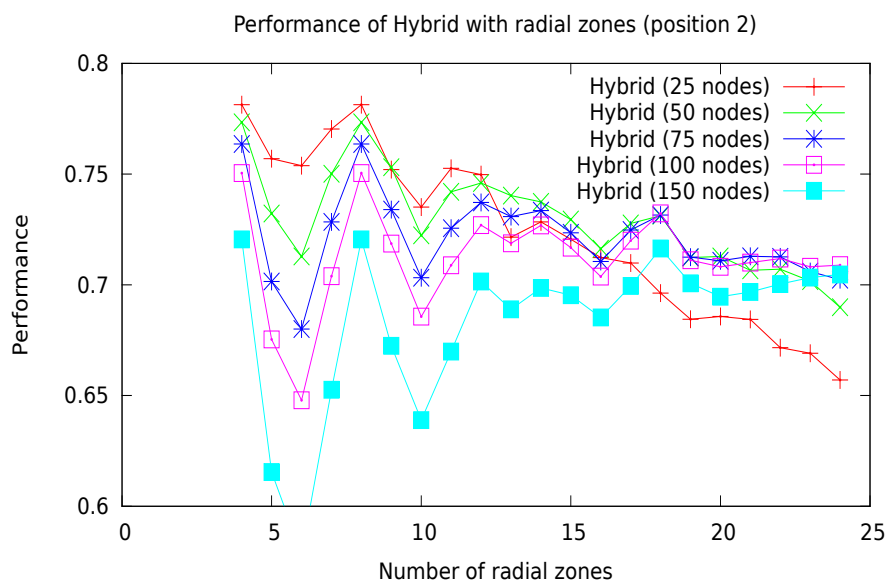


FIGURE 4.6 Performance par zone à la position P2

**Nombre de zones efficaces en P2 en utilisant la méthode radiale** La figure 4.6 n'est pas aussi pertinente que les précédentes. Les courbes diverses montrent également un point optimum. Ce qui signifie un meilleur rendement obtenu avec un nombre de zones égal à huit (8).

**Nombre de zones efficaces en P3 en utilisant la méthode radiale** La dernière figure 4.7 du mode radial place la station de base un peu à l'écart de la zone de captage.

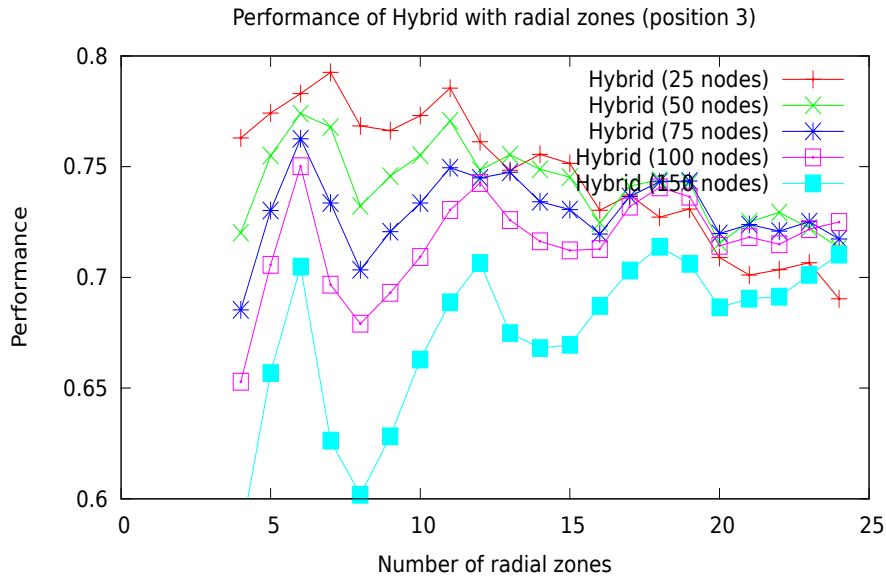


FIGURE 4.7 Performance par zone à la position P3

Les courbes de cette figure diffèrent par rapport aux autres courbes présentées ci dessus. Ces courbes présentent plusieurs points optimums. La performance est presque égale aux autres positions sauf pour la position P3 avec un nombre de nœuds à grande échelle.

En comparant les figures 4.4, 4.5, 4.6 et 4.7, nous pouvons dire que l'optimum des courbes se situe à partir du nombre de zones égal à huit.

Comme avec la méthode radiale, nous allons simuler avec le mode rectangulaire pour le calcul des distances en plusieurs positions. Nous présenterons dans la suite les résultats apportés par ce changement.

**Nombre de zones efficaces en P0 en utilisant la méthode rectangulaire** La figure 4.8 présente la performance de notre algorithme en terme de construction des zones et en fonction du nombre des capteurs déployés. La station de base est située aux coordonnées (200,0). Ces courbes différentes atteignent une valeur maximale quelque soit la taille du nombre de capteurs. C'est lorsqu'on a trois zones, que les capteurs donnent le meilleur rendement.

**Nombre de zones efficaces en P1 en utilisant la méthode rectangulaire** Dans une seconde expérimentation figure 4.9, nous plaçons la station de base au point (400,0) du champ de captage. Cette position a vu une rentabilité déséquilibrée selon la taille du réseau de capteurs. Ces courbes présentent une performance décroissante au fur et à mesure que le nombre de zones croît.

**Nombre de zones efficaces en P2 en utilisant la méthode rectangulaire** La simulation de la figure 4.10 montre une dégradation nette des performances au delà de

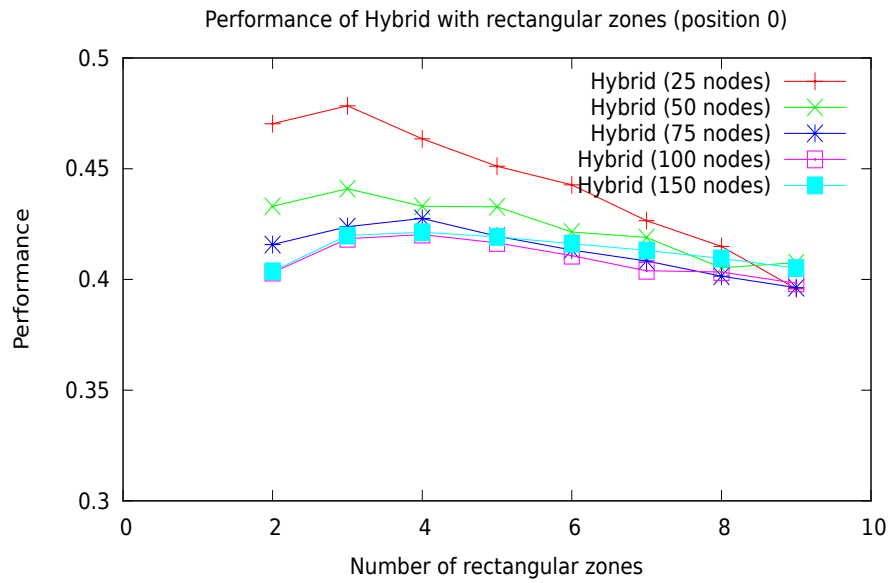


FIGURE 4.8 Performance par zone en mode rectangulaire à la position P0

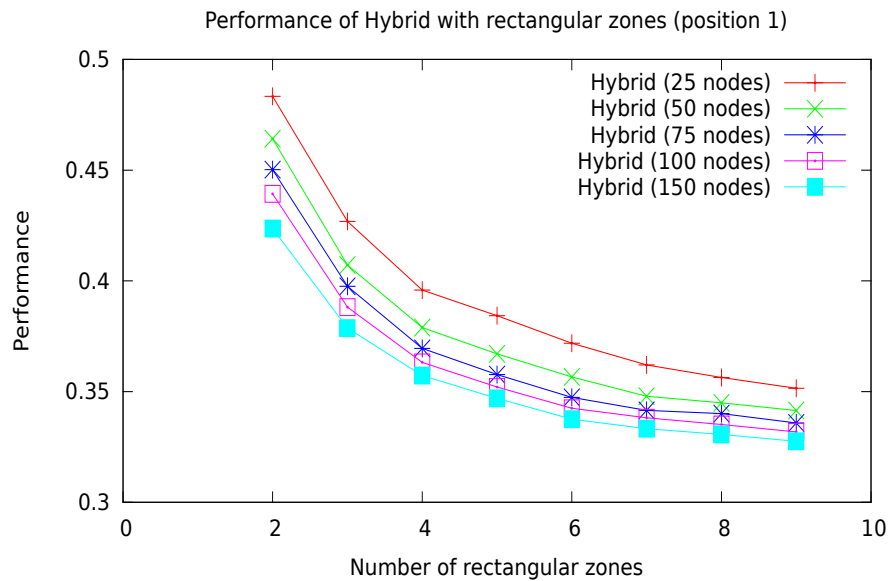


FIGURE 4.9 Performance par zone en mode rectangulaire à la position P1

trois zones. Ce qui nous permet de positionner le meilleur rendement de cette simulation à trois zones.

**Nombre de zones efficaces en P3 en utilisant la méthode rectangulaire** Dans une quatrième expérimentation, nous plaçons la station de base à la position (400, -100) par rapport au champ de captage. Les résultats affichés dans la figure 4.11 montrent une dégringolade à partir de trois zones et n'atteignent pas un rendement de 0.7.

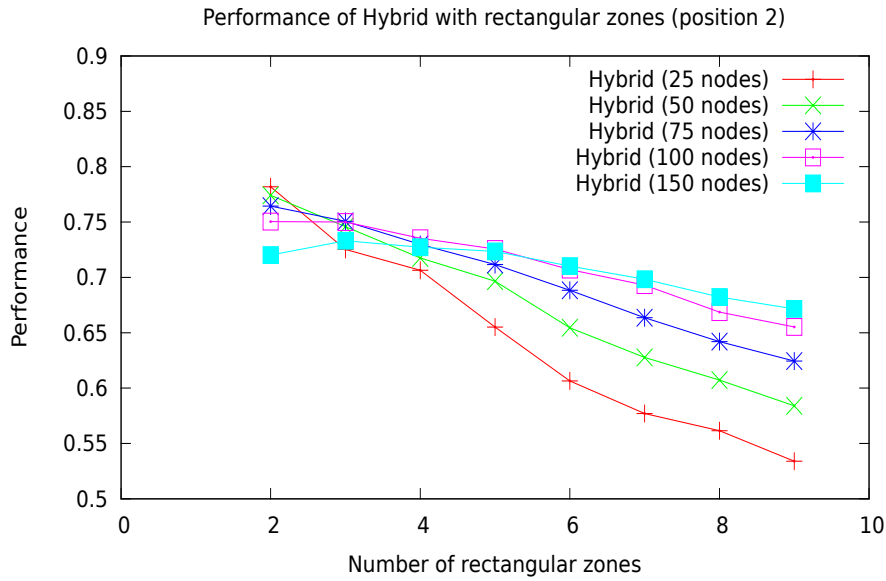


FIGURE 4.10 Performance par zone en mode rectangulaire à la position P2

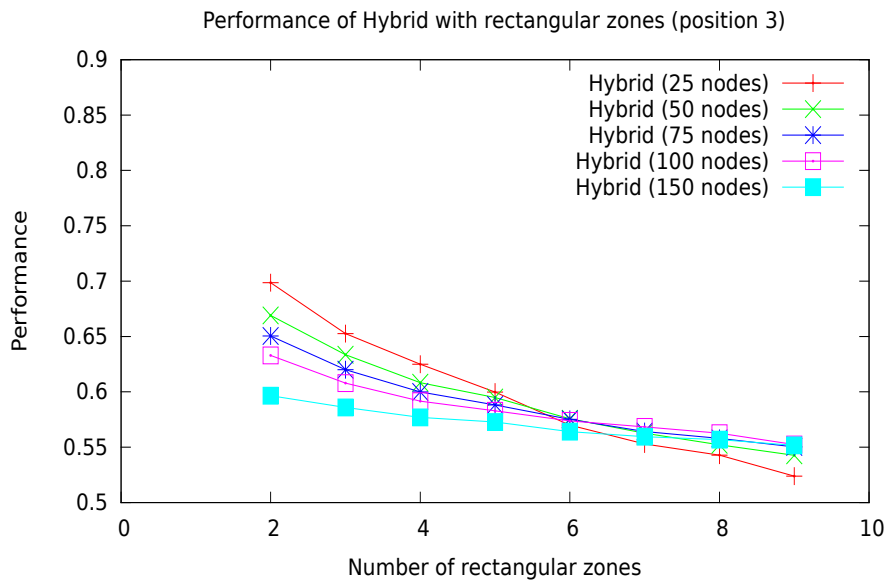


FIGURE 4.11 Performance par zone en mode rectangulaire à la position P3

Par conséquent, en comparant les courbes des figures 4.8, 4.9, 4.10 et 4.11, on retiendra un rendement meilleur avec le mode rectangulaire lorsque le nombre de zones est égal à trois. Pour le mode radial, le rendement s'est avéré intéressant pour un nombre de zones égal à huit.

Cette partie de simulation nous a permis de distinguer certains paramètres clés pour la suite de nos travaux. Nous avons montré l'efficacité du mode rectangulaire à partir de trois zones.

Nous allons retenir ces paramètres pour pouvoir comparer les performances de notre algorithme avec d'autres algorithmes comme LEACH et HEED.

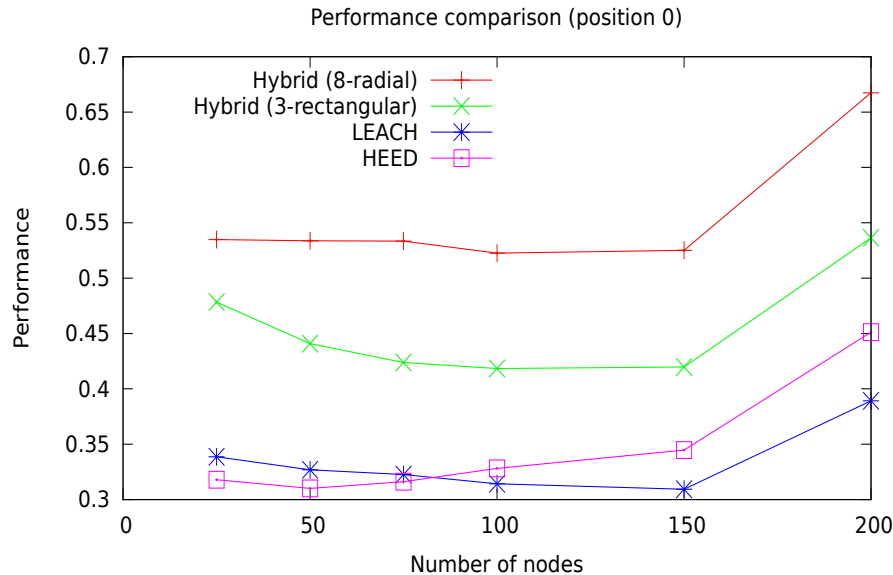


FIGURE 4.12 Performance de l'algorithme hybride à P0

**Comparaison en P0 avec LEACH et HEED** Les résultats de la figure 4.12 montrent l'efficacité des différents algorithmes hybride, LEACH et HEED avec la station de base située à ( $x=200$  et  $y=0$ ). Ces courbes montrent que HEED arrive à augmenter la vie du réseau de capteur par rapport à LEACH. Sur le schéma, nous observons également qu'à cette position de la station de base, notre algorithme hybride est nettement meilleur que HEED.

Nous constatons que plus le nombre de nœuds augmente, plus les performances de nos algorithmes évoluent positivement. Ce qui montre l'efficacité du passage à l'échelle.

**Comparaison en P1 avec LEACH et HEED** En poursuivant toujours dans nos simulations, nous avons repris les mêmes données mais cette fois ci en plaçant la station de base à la position P1 ( $x=400$ ,  $y=0$ ) comme mentionné sur la figure 4.1. Les courbes 4.13 issues de cette simulation montrent une baisse des performances au fur et à mesure que le nombre de nœud augmente.

Ces courbes montrent également l'efficacité de notre algorithme par rapport à LEACH et HEED.

**Comparaison en P2 avec LEACH et HEED** Les courbes de la figure 4.14 montrent les mêmes avantages en terme de performance de notre algorithme par rapport à LEACH et HEED. Et quand le nombre de capteurs est égal à cent, la performance en radiale est égale à la performance en rectangulaire. A cette position, dès que le nombre de nœuds



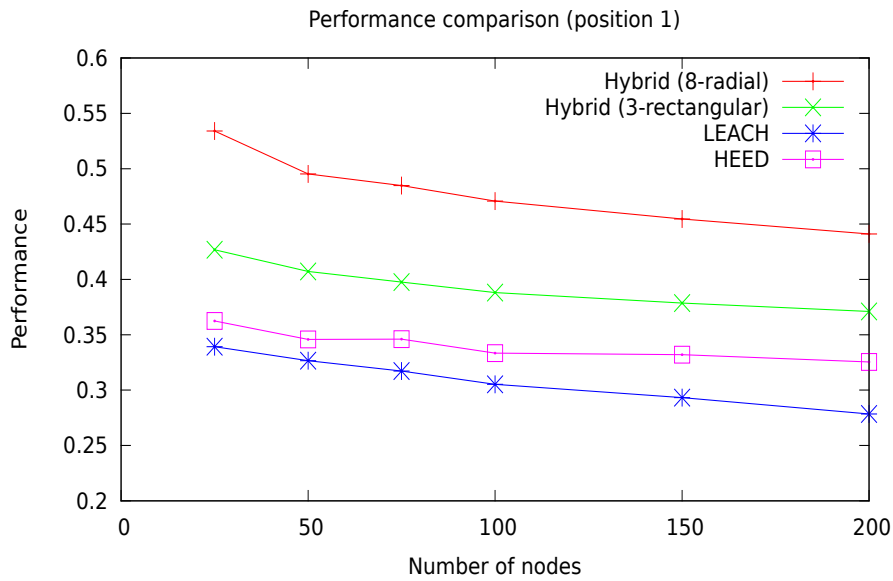


FIGURE 4.13 Performance de l'algorithme hybride à P1

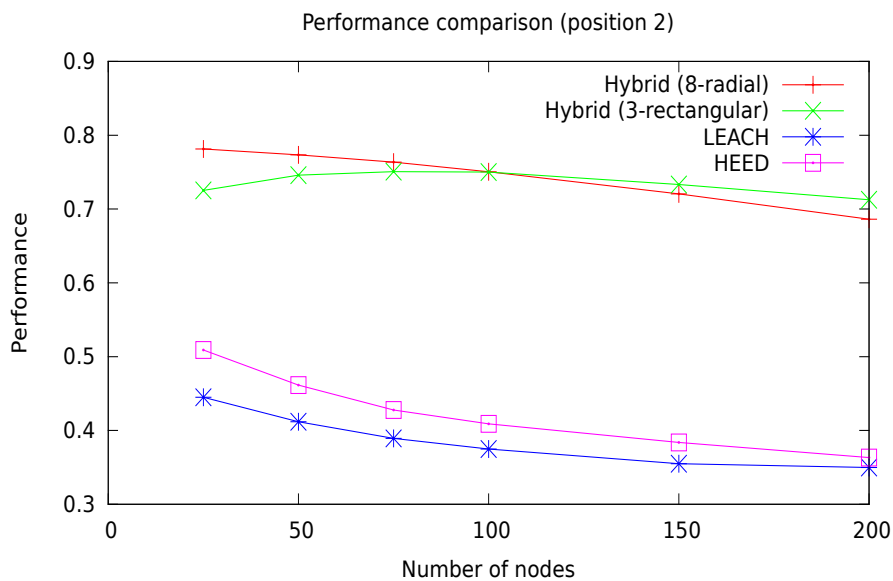


FIGURE 4.14 Performance de l'algorithme hybride à P2

dépasse la centaine, un fossé se creuse entre notre algorithme hybride et les autres. D'où l'efficacité de notre algorithme à grande échelle.

**Comparaison en P3 avec LEACH et HEED** La figure 4.15 nous donne des courbes décroissantes en fonction de l'augmentation du nombre de capteurs. Elles présentent les performances de l'algorithme hybride quelque soit le mode radial ou rectangulaire utilisé par rapport à LEACH et HEED.

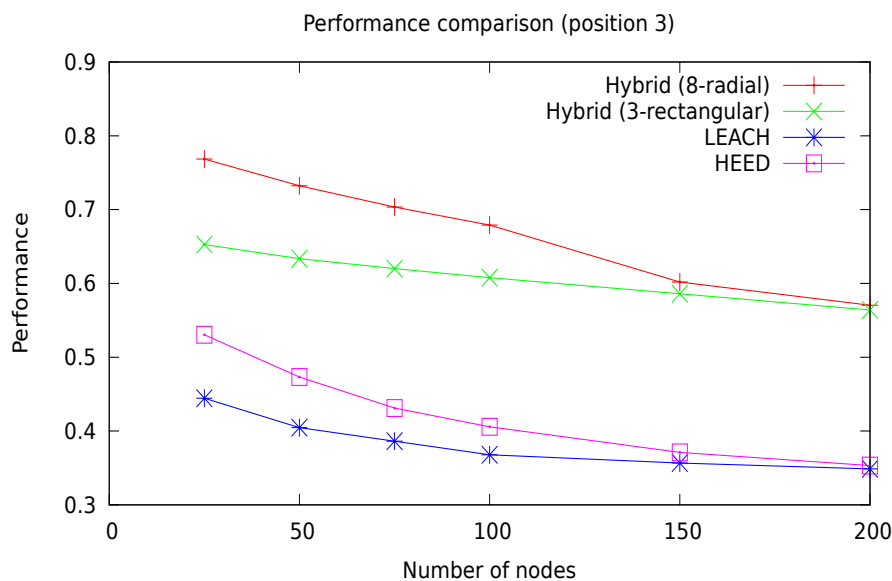


FIGURE 4.15 Performance de l'algorithme hybride à P3

Cette efficacité est plus marquante quand le nombre de nœuds dépasse la centaine.

#### 4.5.4 Précisions et observations diverses

Dans nos expériences, nous montrons que notre algorithme est plus efficace que LEACH et HEED dans certaines configurations. Le mode rectangulaire comme le mode radial donne plus de 30% d'amélioration par rapport à LEACH et 25% d'amélioration comparé à HEED. L'efficacité de notre algorithme semble être significative lorsque nous sommes dans une application à grande échelle.

## 4.6 Expérimentation avec la plateforme SensLAB

Cette partie présente la plateforme sensLAB et l'expérimentation effectuée. En effet, nous avons voulu évaluer notre algorithme sur une vraie plateforme de capteurs.

### 4.6.1 Introduction

Afin d'analyser le comportement de notre algorithme dans un environnement réel, il est possible de s'appuyer sur la plate-forme SensLAB. Nos résultats de recherche ont été évalués uniquement en simulation, et les quelques expérimentations réelles menées à petite échelle (une dizaine de nœuds) ne permettent pas forcément de valider les algorithmes. C'est pourquoi, nous avons voulu utiliser la plateforme SensLAB. Composée de plus de 1000 nœuds capteurs, la plateforme est répartie sur 4 sites (voir figure 4.17) géographiques inter connectés en France (Grenoble 4.21, Lille 4.19, Rennes et Strasbourg). Chaque site possède 256 nœuds et présente des spécificités aux niveaux matériel et environnemental.

Site	Grenoble	Strasbourg	Euratech	Rennes
Serveur url (pour ssh)	grenoble.senslab.info	strasbourg.senslab.info	euratech.senslab.info	renne.senslab.info
Radio chip	CC1101 800 MHz	CC1101 800 MHz	CC2420 2.4 GHz	CC2420 2.4 GHz
Nœud mobiles	non	oui	oui	non

FIGURE 4.16 Caractéristiques des sites sensLAB.  
[dRCD+11b]

La plateforme est dédiée à l'expérimentation de réseaux de capteurs à grande échelle. Elle fournit des outils pour déployer rapidement et facilement des expériences, évaluer et analyser les résultats obtenus à la fois par l'expérience et par le banc d'essai. Ainsi, SensLAB propose des nœuds fixes ou mobiles (voir tableau 4.16), intérieurs ou extérieurs, déployés en topologie en grille 2D ou 3D [DVMSR10]. Pour éviter de manipuler les capteurs un par un, SensLAB permet de charger automatiquement une application sur l'ensemble des nœuds sélectionnés.

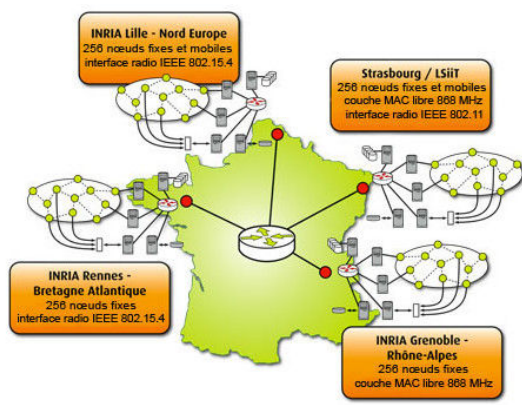


FIGURE 4.17 les quatre (4) sites SensLAB  
[dRCD+11b]

## 4.6.2 Nœud SensLAB

Un nœud SensLAB (Figure 4.18) se compose de trois composants matériels : le nœud ouvert, le nœud de contrôle et la passerelle.

**Le nœud ouvert :** Le nœud ouvert est la partie dédiée à l'utilisateur pour son expérimentation. Il s'agit d'un capteur classique totalement programmable.

**Le nœud de contrôle :** Le nœud de contrôle est la partie permettant la mise en route du nœud ouvert et sa programmation. Il apporte les fonctionnalités de mesure et d'injection durant les expérimentations. Il présente un grand avantage, l'utilisateur se concentre uniquement sur le code qu'il doit évaluer via les nœuds capteurs sans tenir compte de leur consommation d'énergie.

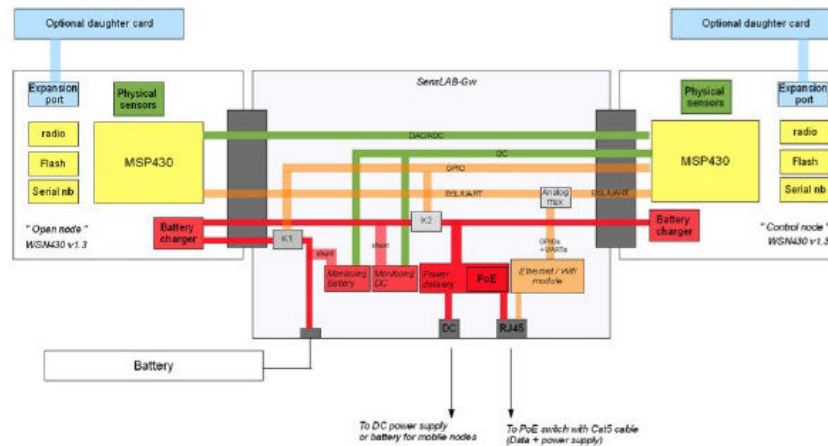


FIGURE 4.18 Architecture d'un nœud senslab

La passerelle : Elle assure l'alimentation des nœuds ouvert et de contrôle, la communication entre ces deux entités, ainsi que la communication entre le nœud SensLab et la plateforme.

Lorsqu'ils sont fixes, les nœuds SensLAB sont reliés à la plateforme par Ethernet. Les nœuds mobiles et extérieurs sont eux reliés via Wifi



FIGURE 4.19 site senslab de Lille

### Le nœud ouvert WSN430

Comme le montre la figure 4.20, un nœud ouvert WSN430 est composé des éléments suivants :

Micro-contrôleur MSP430F1611 : Construit autour d'un CPU 16 bits, il a été conçu pour des applications embarquées à basse consommation et à faible coût. Il est par-

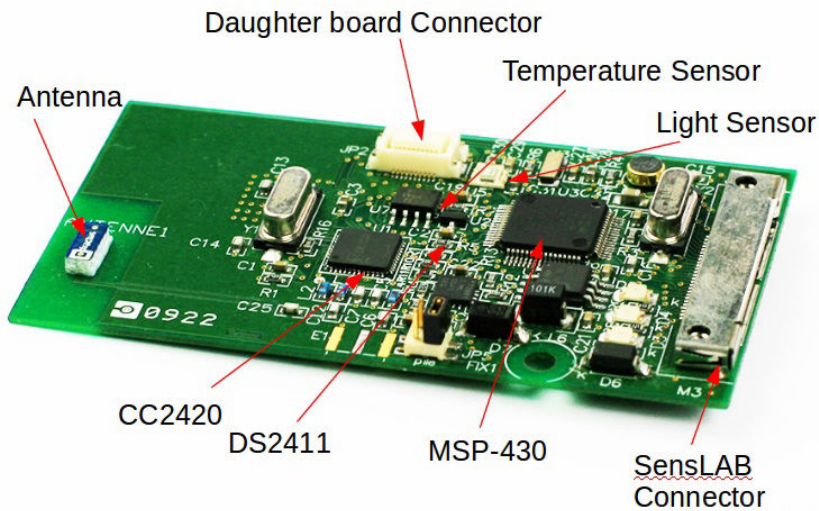


FIGURE 4.20 Noeud WSN430  
[dRCD<sup>+</sup>11a]

ticulièrement adapté aux applications sans-fil fonctionnant sur batteries. Il intègre une mémoire Flash de 48ko et une mémoire RAM de 10ko.

Le chip radio : Le chip radio (CC1100 ou CC2420 de Texas Instrument) est un émetteur/récepteur radio pour des fréquences respectives de 800MHz (CC1100) et 2.4GHz (CC2420) bon marché. Il est connu pour sa faible consommation.

Capteur de température DS1722.

Capteur de luminosité TSL2550.

Le chip DS2411 : Il fournit une identité absolument unique. Ce numéro de série peut être déterminé avec une interface électronique minimale. L'immatriculation du chip est codée sur 64 bits. Elle inclut un numéro de série de 48 bits, un CRC de 8 bits ainsi que 8 autres bits correspondant à sa famille.

Batterie de 3,7 volt avec une puissance de 830mAh.

## Contiki OS

Contiki OS est un des systèmes d'exploitation utilisés avec les réseaux de capteurs. On peut également utiliser Tinyos, OpenWSN, Riot ou FreeRTOS. Utilisé pour l'internet des objets aussi, Contiki se sert du langage C pour la programmation des tâches. A Contiki s'associe un simulateur Cooja en java pour la visualisation des expérimentations. Contiki offre deux types de pile protocolaire :

la couche Rime qui permet un dialogue vers les capteurs voisins ainsi que le routage. La pile de Rime prend en charge un seul saut et primitives de communication multi-hop. Les primitives multi-hop ne précisent pas comment les paquets sont acheminés à travers le réseau.

la couche uIP(micro IP) , orientée Internet qui offre les services essentiels du protocole IP mais nécessite plus de ressources que Rime.

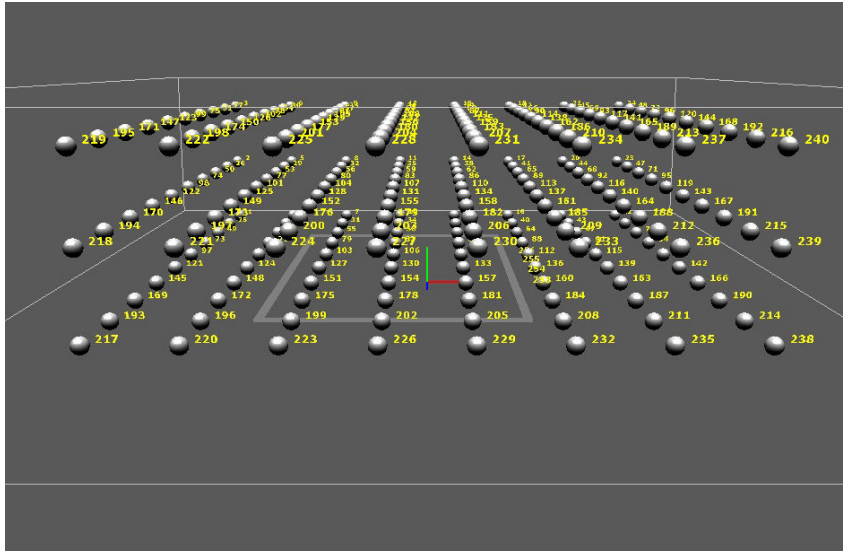


FIGURE 4.21 site senslab de Grenoble

Une fois que le programme est installé sur l'une des plateformes citées précédemment, la codification du code en hexadécimal est exploitable par Senslab.

### 4.6.3 Expérimentation

Nous avons choisi dix noeuds sur le site de Grenoble qui forment une topologie en grille.

Chaque nœud du cluster formé de ces 10 noeuds envoie les données récoltées (ici on a considéré la température) au Cluster Head qui a son tour agrège les données pour la station de base. A la fin de l'expérimentation, des fichiers de résultat sont générés automatiquement selon les spécifications de l'utilisateur. Ces fichiers sont de type CSV (Comma Separated Values). Les mesures possibles sont : le courant, la tension, la puissance, la température, la luminosité.

Chaque ligne de toute mesure contient trois champs : le numéro du noeud sur lequel la mesure a été prise, le temps de mesure (référéncé au début de l'expérience), et la valeur de mesure : Volt pour la tension, Ampère pour le courant, Watt pour la puissance, lux pour la luminosité, degré Celsius pour la température.

La figure 4.22 représente la consommation de l'énergie du réseau au cours du temps. Cette énergie a été calculée sur la base de la formule  $E = P * t$ . L'expérience s'est déroulée sur dix (10) capteurs et a duré un certain temps avec un prélèvement de la puissance tous les 500 ms. Ce qui nous a valu plusieurs données pour chaque capteur. Nous avons répété l'expérience une cinquantaine de fois.

La courbe de la figure 4.22 montre l'évolution de la consommation d'énergie du réseau de l'algorithme SHE sur senslab. Cette consommation est, sans surprise, croissante dans le temps et nous donne de bonnes perspectives pour la suite des expérimentations à une large échelle.

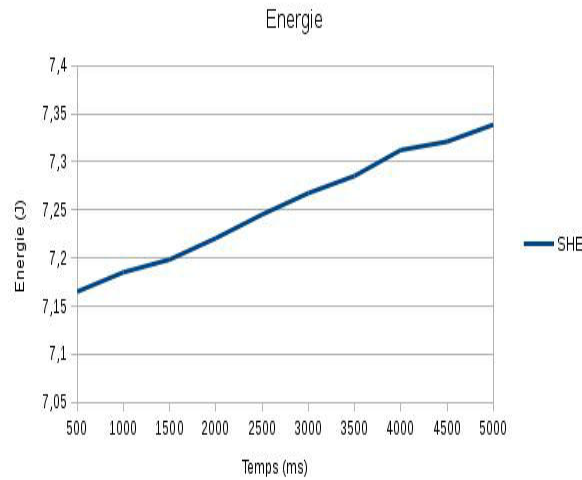


FIGURE 4.22 Evolution de la consommation de l'énergie du réseau en fonction du temps

Vu les limites de la plateforme, notamment au niveau du positionnement géographique, notre expérimentation s'est focalisée sur la consommation d'énergie des nœuds simples, des nœuds CH et de la consommation globale du réseau. Avec plusieurs séries d'expérimentation effectuées à plusieurs reprises, on aboutit à une consommation d'un nœud simple cc1101 800MHz représentée par la figure 4.23.

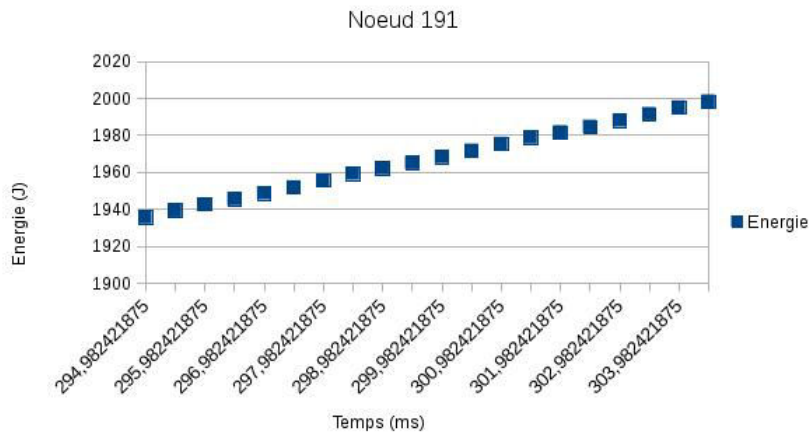


FIGURE 4.23 Energie consommée par un noeud

Cette consommation est obtenue après des expériences de six minutes sur cinq nœuds avec un nœud représentant le cluster head. La courbe suivante montre la consommation d'énergie du Cluster-Head pour la collecte d'information auprès des autres nœuds. Cette courbe montre l'évolution de la consommation d'énergie du cluster head pour le temps de collecte d'information et aussi une très faible sinon insignifiante au temps de pause ou d'inactivité du cluster head. Nous avons poursuivi notre expérience sur 10 nœuds toujours avec un nœud agissant comme cluster-head. Nous obtenons les mêmes allures avec

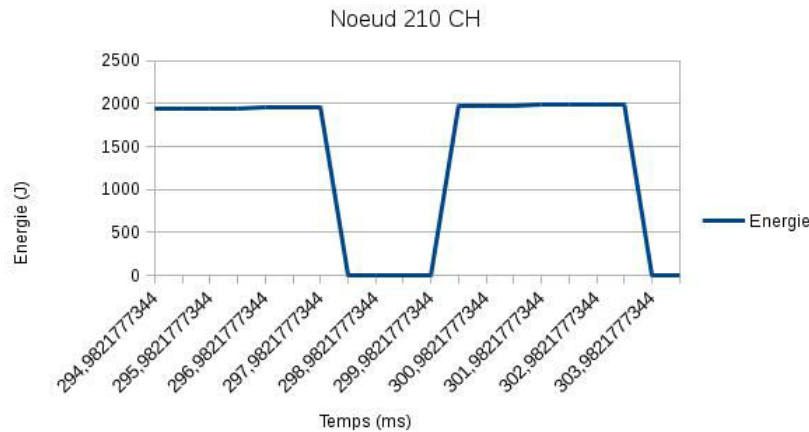


FIGURE 4.24 Energie consommée par un noeud CH

les courbes des nœuds simples et du cluster-head. La courbe de la figure 4.25 présente l'évolution de la consommation d'énergie du cluster-head. On voit cette consommation diminuer puis disparaître lorsque le CH se repose. A l'activation de son rôle de CH, la consommation croît et retrouve son niveau de départ. Plus on a de nœuds, moins le cluster-head se repose.

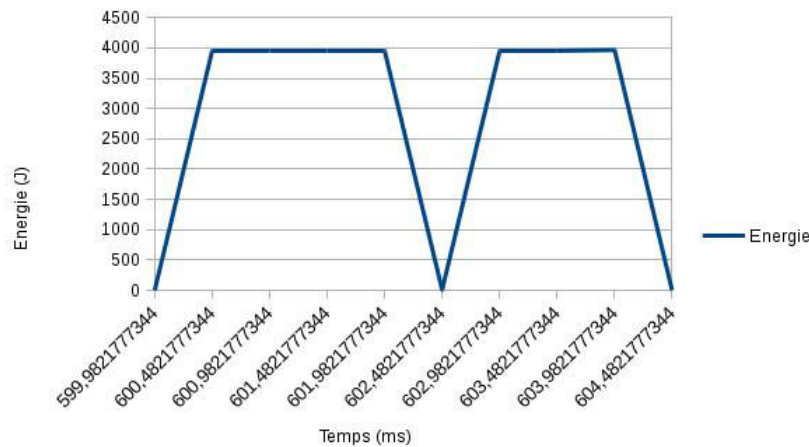


FIGURE 4.25 Energie consommée par un noeud CH

Cette expérience nous a permis de déployer concrètement l'algorithme sur une plateforme réelle afin d'éprouver notre algorithme. La plate-forme ne nous a cependant pas permis de déployer intégralement ces algorithmes. La position des capteurs et leur état d'énergie manquent pour pouvoir aller au bout de notre démarche. Cependant, ce premier pas sur une plate-forme réelle nous a montré que l'algorithme pouvait être appliqué sur le terrain et nous a permis, également, de conforter notre vision des capteurs. Nos simulations ont donc bien été effectuées avec des paramètres concrets et réels sur les capteurs.



## **4.7 Conclusion**

Cette partie nous a permis de présenter notre algorithme et de montrer l'efficacité de celui-ci par rapport à LEACH et HEED.

On peut retenir que cette nouvelle approche dans la formation des clusters, dans l'évaluation des capteurs en tant que nœud simple ou Cluster Head permet d'économiser l'énergie des capteurs pour augmenter la vie du réseau de capteurs. Avec cet algorithme, on arrive à économiser l'énergie dépensée dans la formation des clusters, dans la désignation des CHs, et diminuer l'énergie perdue dans l'attente de transmission d'un message par un nœud en manque d'énergie.

Nous avons également testé cet algorithme en utilisant une plate-forme de capteurs réels. Cette expérimentation avec SensLAB nous laisse entrevoir des résultats encourageants. Notre algorithme étant fonctionnel sur un réseau de capteurs fixes, nous développons dans la suite du document une autre contribution se basant sur le précédent travail pour économiser l'énergie et maintenir la connectivité dans des réseaux de capteurs mobiles.

# Chapitre 5

## Adaptation de SHE aux réseaux de capteurs mobiles : utilisation de l'algorithme des chauves-souris

### Sommaire

---

<b>5.1</b>	<b>Topologie dynamique et mobilité de capteurs</b>	<b>92</b>
5.1.1	Exemples d'application	92
5.1.2	Problématique	94
<b>5.2</b>	<b>Echolocalisation</b>	<b>96</b>
5.2.1	Quelques techniques de localisation	96
5.2.2	Le principe de l'écho-localisation	96
5.2.3	Algorithme d'optimisation par essaims de particules	98
5.2.4	Algorithme des chauves souris	98
<b>5.3</b>	<b>SHEM : Sensor Hybrid Energy Mobility</b>	<b>99</b>
5.3.1	Description générale de la méthode	99
5.3.2	Notre algorithme détaillé	101
<b>5.4</b>	<b>Simulation</b>	<b>104</b>
5.4.1	Déplacement	104
5.4.2	Paramètres de simulation	105
5.4.3	Mesures de performance	105
5.4.4	Premier scénario	106
5.4.5	Deuxième scénario	110
5.4.6	Troisième scénario	116
5.4.7	Bilan de la simulation	116
<b>5.5</b>	<b>Conclusion</b>	<b>118</b>

---

Dans la première contribution, nous nous sommes intéressés à la gestion de l'énergie dans des réseaux de capteurs statiques. Cette deuxième contribution doit nous permettre de prendre en compte les situations suivantes :

Un capteur est défaillant. En effet certains nœuds peuvent être défaillants ou inhibés à cause d'un manque d'énergie, d'un dommage physique ou d'une interférence. Ceci ne doit pas affecter la globalité de la tâche du réseau de capteurs. En cas de défaillance, de nouveaux liens et routes doivent être établis pour assurer la collecte des données.

Les capteurs sont mobiles. La mobilité des nœuds modifie considérablement la topologie. Nous pouvons prendre l'exemple d'une foule en mouvement dans laquelle certaines personnes seraient dotées d'un capteur. Nous aborderons ce problème dans la partie 5.2.1 parlant de l'évacuation des foules. Nous nous intéresserons à deux situations différentes selon que les capteurs puissent ou non avoir une influence sur leurs déplacements.

Un capteur peut être rajouté au réseau comme dans l'exemple de suivi des déplacements des pompiers en cas de catastrophe.

Il faut donc gérer avec précision la maintenance de cette topologie au moment du déploiement, du post-déploiement et du redéploiement des nœuds capteurs.

Nous présentons donc dans cette partie le problème de topologie dynamique, liée aux situations évoquées plus haut dans les réseaux de capteurs. Par la suite, nous revenons sur l'état de l'art lié à ce problème. Puis, nous présentons notre contribution avec un algorithme de maintenance automatique de la topologie.

## 5.1 Topologie dynamique et mobilité de capteurs

Les capteurs utilisent leur énergie principalement à des fins de calcul et de transmission de données. La durée de vie d'un capteur dépend essentiellement de celle de sa batterie. Dans un WSN, chaque nœud joue le rôle d'émetteur et de routeur. Une défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et imposer une réorganisation de ce dernier.

De la même manière, la topologie peut évoluer du fait de la mobilité des capteurs ou de l'arrivée de nouveaux nœuds.

Afin de bien clarifier cette problématique, nous débutons par une présentation de quelques exemples représentatifs. Nous détaillons ensuite cette problématique avant de rappeler quelques pistes tirées de l'état de l'art. Puis, nous présentons notre contribution basée sur l'écho-localisation. Enfin nous évaluons notre méthode avec des simulations expérimentales.

### 5.1.1 Exemples d'application

Dans cette partie, nous présentons des exemples de situations qui mettent en exergue la topologie dynamique et qui peuvent influencer notre contribution présentée plus loin sur l'écho-localisation et notre algorithme de déplacement des chauves souris.

#### Mouvement de foule

La modélisation des mouvements de foule revêt un intérêt particulier pour l'architecture, la planification urbaine et la gestion du territoire, comme par exemple dans le cadre

des transports urbains, des manifestations sportives, de l'accès aux sites religieux, des rassemblements politiques et à l'accès aux sorties de secours. Dans ce contexte, la gestion du trafic piétonnier vise à optimiser les temps de parcours et éviter les accidents dus à des situations de panique à travers une meilleure planification des infrastructures.

La plupart des modèles disponibles actuellement sont des modèles microscopiques qui décrivent la trajectoire de chaque individu en tenant compte des interactions entre individus. Les chercheurs de l'équipe OPALÉ du centre INRIA Sophia Antipolis Méditerranée, en collaboration avec des scientifiques de l'Université de Brescia et du ICM (Université de Varsovie, Pologne), développent une approche macroscopique qui vise à décrire le comportement global d'une foule en utilisant des équations dérivées de la dynamique des fluides [CDGS07]. Les modèles doivent être conçus de façon à reproduire le comportement des piétons. Notamment, les piétons visent généralement le chemin le plus rapide (qui n'est pas forcément le plus court) et préfèrent éviter les endroits bondés et les parois. En revanche, en situation d'urgence, les gens sont pris de panique et leur comportement devient irrationnel, ce qui peut provoquer de graves accidents. Dans ce cas, la tendance est celle de suivre le groupe qui se précipite vers la sortie la plus proche, en délaissant les éventuelles sorties secondaires qui permettraient une évacuation plus rapide. Grâce à l'emploi d'un nombre réduit de variables et paramètres, les modèles macroscopiques sont bien adaptés à la solution de problèmes de gestion et contrôle optimal.

Ces caractéristiques peuvent être reproduites en couplant une loi de conservation (celle du nombre de piétons) avec une équation écho-localisation semblable aux mouvements de chauves-souris, qui donne la direction du «plus court chemin» tout en tenant compte de la répartition des piétons dans l'espace : il peut en effet être préférable de faire un détour par une zone vide, dans laquelle on peut marcher rapidement, plutôt que de passer en ligne droite par un endroit où il y a beaucoup de monde. Par exemple, on peut déterminer la position et la forme optimale d'une sortie, afin de faciliter l'écoulement et minimiser les temps d'évacuation. En particulier et de façon contre-intuitive, un ou plusieurs obstacles positionnés devant une sortie peuvent diminuer la pression entre les individus, et ainsi faciliter l'évacuation d'une salle, tout en prévenant et minimisant le risque d'accident. Dans cette situation, on pourrait doter certains piétons en mouvement de capteurs, pour localiser les issues de sortie, et les diriger vers les issues libres en tenant compte de l'effet de foule.

### **Suivi des déplacements des pompiers dans une catastrophe [LTGLH07]**

Les secouristes, de par la nature de leurs missions, interviennent en environnement à risque et mettent parfois leur vie en danger. Les opérations d'information, de surveillance et de suivi des personnes évoluant dans un environnement sinistré (les pompiers) nécessitent des améliorations. Un pompier est une personne entraînée à combattre le feu et à offrir une gamme de secours de diverses natures (inondations, accidents, sauvetages, secours et assistance à personnes en danger, etc.).

Afin de faciliter le travail des pompiers lors de leurs interventions, des chercheurs du Fraunhofer Institute for Applied Information Technology (Sankt Augustin, Allemagne) ont développé un ensemble d'outils de terrain et de simulations informatiques, appelés respectivement LifeNet et FireSim. Les deux systèmes ont pour vocation d'améliorer la

localisation des hommes, ainsi que la navigation, la gestion des informations et des communications dans des environnements hostiles où les options et la liberté de mouvement sont souvent limitées. LifeNet regroupe différents capteurs et modules placés directement sur la tenue du pompier. Un éjecteur automatique de balises, placé dans le dos, permet de créer un réseau sans fil sur l'ensemble de la zone sinistrée. Des capteurs supplémentaires, situés dans les bottes donne la possibilité à chacun de repérer la position de ses équipiers. De même les pompiers à New York le 11 septembre ne se seraient pas tous engouffrés dans des escaliers des tours jumelles si ils avaient eu des informations plus précises sur le sinistre dans les étages supérieurs

## Déplacement groupé des drones

Les drones présentent de réels avantages dans le domaine civil par exemple pour les opérations de sauvetage et dans le domaine militaire pour la reconnaissance du terrain. Dans ce cadre un groupe de petits drones adéquatement équipés de capteurs pourrait constituer un réseau de capteurs mobiles. Ces petits engins pourront certainement trouver de nombreux usages pour mettre en place très rapidement des signalisations ou transmettre de l'information d'urgence et/ou collecter des données en cas d'accident ou catastrophe dans le domaine civil à moindre coût. Leur déplacement groupé pourra se faire par l'algorithme SHEM que nous détaillons plus loin dans ce chapitre.

Un système de transmission de données entre le drone et la station de base, achemine à la fois les ordres venant du sol (en temps réel) et les informations envoyées par le drone (en temps réel ou différé, le plus souvent par intermittence). Cette transmission s'effectue par télécommunication, soit en portée optique (ligne directe) sur de courtes distances jusqu'à 50 km ; soit en utilisant un relais, ce dernier pouvant être un autre drone proche de la station de base. Ce qui sera l'application de l'algorithme SHE.

### 5.1.2 Problématique

Les problèmes qui découlent de la mobilité des nœuds et de la défaillance des nœuds sont [YZLM14, MRS14, SSB14] :

architecture réseau : la plupart des architectures réseau reposent sur des capteurs statiques. Pourtant, la mobilité des stations de base et/ou des capteurs est parfois nécessaire dans de nombreuses applications. La transmission de messages en provenance, ou vers, un nœud mobile est un autre défi. La capture peut être aussi bien statique que dynamique, suivant l'application.

Connectivité : la plupart des WSN possèdent une densité importante de capteurs, excluant ainsi l'isolement de nœuds. Pourtant, le déploiement, la mobilité et les défaillances font varier la topologie du réseau. Ainsi, la connexité n'est pas toujours assurée [YMG08, WXZ<sup>+</sup>03, RM04].

Gestion de la mémoire : elle est saturée en attente du nœud mobile pour la collecte de l'information. Les auteurs de [SRS04, GBE<sup>+</sup>05] proposent des méthodes de planification de la circulation des éléments mobiles dans un réseau de capteurs où les nœuds de capteurs prélèvent des échantillons à des taux différents. Dans ce cas,

certains nœuds doivent être visités plus fréquemment que d'autres, afin d'éviter les pertes dues à un débordement de tampon de données ;

Retard dans la transmission à cause d'un nœud collecteur dans la collecte de données. [WQD<sup>+</sup>14] essaye d'apporter une solution à ce problème ;

Localisation du nœud relais ou de transmission ayant disparu du réseau par manque d'énergie ou de vol. La localisation d'un nœud est nécessaire dans un réseau de capteurs pour associer une position aux valeurs mesurées par le ou les capteurs et en cas d'utilisation d'un algorithme de routage géographique. De nombreuses techniques de localisation ont été développées. Mais la plupart considère un emplacement aléatoire du nœud [DEG<sup>+</sup>01, HT05, Yan10].

Qualité de service : dans diverses applications, la donnée doit être transmise dans une certaine plage de temps, après quoi, elle devient caduque. Pourtant, dans la plupart des applications, la durée de vie du réseau est favorisée au détriment de la qualité d'émission des données. Les protocoles de routage assurant une qualité de service et prenant en compte la gestion de l'énergie, représentent un défi nouveau et stimulant [AKK04].

Dans les travaux [YZLM14, MRS14, SSB14], la topologie peut être décrite de quatre manières :

1. un ou plusieurs nœuds capteurs sont mobiles et un puits fixe situé dans la cellule de communication de chaque nœud capteur. Les nœuds mobiles font la collecte et les paquets sont acheminés en un saut à la station de base [BSM<sup>+</sup>07].
2. plusieurs nœuds-capteurs mobiles et un puits fixe, les nœuds mobiles étant en position relative fixe et assumant pour certains la fonction de capteur émetteur, pour d'autres la fonction de capteur émetteur et de relais et pour d'autres la seule fonction de relais. Le réseau comprend alors une partie embarquée avec une topologie spécifique. La liaison mobile fixe est réalisée entre un des nœuds relais mobiles (celui-ci n'étant pas toujours le même au cours du temps) et le puits [CBD02, SJ04, LW07]. Le contrôle de la topologie, obtenu en modifiant la puissance d'émission, sera utilisé notamment pour s'affranchir de certaines contraintes physiques ne permettant pas de placer les nœuds aux meilleurs endroits.
3. plusieurs relais dans la nature (infrastructures) afin que la liaison mobile fixe soit réalisée de manière optimale entre chaque nœud capteur et un des nœuds relais. Celui-ci n'étant pas toujours le même au cours du temps, et un ensemble de relais permet l'acheminement des paquets jusqu'au puits [GDPV03].
4. Une rotation de nœud mobile collecteur par les nœuds voisins [LH05]

En effet, les liens de communications sans fil sont sujets aux pannes, provoquant des partitionnements dans le réseau et des changements dynamiques dans sa topologie. Les messages peuvent être altérés ou perdus à cause de la nature de la communication erronée.

De plus, le bruit, les interférences, les collisions et la volatilité du voisinage conduisent à une chute importante des performances.

Aussi, dans les protocoles de contrôle de topologie où chaque capteur doit ajuster sa puissance de transmission pour minimiser sa consommation énergétique, les algorithmes ont le plus souvent besoin d'information sur la position de leurs voisins [DPS08]. Cette

approche consiste à mettre à profit la mobilité des capteurs pour rétablir une certaine forme de connectivité entre les sous réseaux de capteurs ou entre des clusters. En effet il arrive qu'un ensemble de capteurs soit isolé du reste du réseau [CPP09, JSB<sup>+</sup>06].

Nous avons dans une précédente contribution proposé un algorithme permettant d'augmenter la durée de vie du réseau de capteurs en mettant en place des clusters et la gestion de ses clusters.

Nous allons proposer une solution élaborée à partir de l'écho-localisation dont nous détaillerons les principes dans les parties suivantes.

## 5.2 Echolocation

La localisation d'un nœud mobile est une fonctionnalité importante dans des réseaux très divers et pour des terminaux comme les smartphones, les véhicules connectés en VANET, les nœuds de réseaux de capteurs [YZLM14].

La localisation est généralement gérée à l'extérieur par le système GPS (Global Positioning System), car il est relativement précis et peu cher. Son coût énergétique est en revanche élevé, à cause de l'utilisation des satellites et de la synchronisation de l'horloge avec les satellites les plus proches. Ceci entraîne donc la diminution de la durée de vie effective de l'ensemble du réseau .

La localisation à l'intérieur d'un RCSF ne peut pas s'appuyer sur le GPS, à cause du manque de précision. Ainsi, il ne répond pas à diverses contraintes de réseaux de capteurs, car il est un peu lent, coûteux et repose sur une infrastructure par satellite. Diverses techniques ont été développées.

### 5.2.1 Quelques techniques de localisation

Plusieurs techniques de localisation existent parmi lesquelles, nous pouvons citer les techniques Range-based et les techniques Range-free.

1. Les techniques Range-based [BONL08]. Elles sont fondées sur la mesure de la distance ou des angles entre les directions à des nœuds de référence appelés ancres. Elle évalue la distance entre un émetteur et un récepteur radio.
2. Les techniques Range-free [HHB<sup>+</sup>03]. Elles sont basées sur l'hypothèse qu'un nœud situé dans un environnement stable vis-à-vis des conditions de propagation des signaux est perçu de manière totalement dépendante de sa position par l'ensemble des nœuds fixes qui captent son signal.

Ces techniques courantes ne renseignent pas sur la vitesse des nœuds émetteurs ou récepteurs.

Cette deuxième contribution s'appuie sur le principe de l'écho-localisation que nous détaillerons à la suite de cette partie.

### 5.2.2 Le principe de l'écho-localisation

L'écho-localisation consiste à envoyer des sons et à écouter leur écho pour localiser, et dans une moindre mesure identifier, les éléments d'un environnement. Elle est utilisée par

certains animaux, notamment des chauves souris et des cétacés, et artificiellement avec le sonar.

Il permet à ces animaux de localiser les éléments de leur environnement (obstacles, parois de grottes ou autres cavités) et repérer leur nourriture (exemple : fleurs ou feuilles de plantes réfléchissant l'écho des ultrasons de chauves-souris) ou leurs proies des milieux où la vue est inefficace à cause du manque de lumière (nuit, grotte, profondeur marine, turbidité de l'eau).

Certains aveugles utilisent l'écho-localisation afin de localiser des obstacles. L'utilisation de l'ouïe par les aveugles est notée par Denis Diderot dans sa Lettre sur les aveugles à l'usage de ceux qui voient en 1749. Les premières expériences scientifiques sur le sujet débutent vraiment en 1944 avec les travaux de Michaels Supa et de son équipe qui confirme que c'est bien l'écho des sons qu'ils émettent qui permet à des aveugles de déterminer la distance de certains obstacles.

L'écho-localisation permet de déterminer la distance de l'obstacle, par la durée écoulée entre l'émission du son et la perception de l'écho. L'émetteur muni de deux oreilles mesure l'écart entre les deux réceptions et en déduit la direction de la cible.

L'écho-localisation renseigne sur sa taille, par l'intensité de l'écho (plus la cible est petite, moins elle réfléchit de son) et la durée de l'écho (une grande cible ne produit pas un écho bien net, mais un écho plus long au fur et à mesure de la réception en provenance des parties de plus en plus éloignées de la cible). Par mesure du décalage Doppler, elle renseigne aussi sur la vitesse relative de la cible par rapport à l'émetteur. Enfin, chaque type de cible déforme l'écho de façon caractéristique, ce qui permet à l'émetteur d'en déterminer la nature ; les battements d'ailes des insectes, notamment, signalent leur présence dans l'écho. Selon l'utilisation, ce ne sont pas les mêmes types de cris qui sont utilisés, et notamment pas les mêmes fréquences.

Le processus d'écho-localisation est très complexe et il a été étudié en détail par de nombreux chercheurs [FJFY13]. Le comportement peut être modélisé en utilisant trois règles générales :

1. Toutes les chauves-souris utilisent l'écho-localisation pour détecter la distance et elles devinent aussi la différence entre alimentaire / proie et également le fond d'obstacles d'une manière inexplicable.
2. Les chauves-souris volent au hasard et elles fixent une fréquence minimale, la longueur d'onde et l'intensité variable pour rechercher une proie. Elles peuvent ajuster automatiquement la fréquence des impulsions émises en fonction de la proximité de leur cible.
3. Bien que l'intensité peut varier de plusieurs façons, nous devons supposer que l'intensité varie d'un sommet à une valeur minimale constante.

Nous allons par la suite expérimenter au niveau du réseau de capteurs avec des antennes pour chaque capteur. Cette partie montrera l'importance de la connectivité dans le gain d'énergie pour gérer la topologie dynamique du réseau de capteurs mobiles. Nous allons utiliser l'écho-localisation dans le cadre de notre contribution en se basant sur l'algorithme de déplacement des chauves souris similaire aux algorithmes d'optimisation par essaims de particules présentés ci dessous.



### 5.2.3 Algorithme d'optimisation par essais de particules

Proposé par [Cle05], l'optimisation par essaims particulaires (OEP en abrégé), est une technique bio-inspirée qui exploite une population d'individus pour la recherche de régions prometteuses au sein d'un espace de solutions. Chaque individu (appelé particule) se déplace d'une position à une autre avec une vitesse adéquate en ajustant la direction de sa trajectoire selon trois paramètres : sa meilleure position, la meilleure position au sein de ses proches voisins ainsi que la meilleure position dans le groupe. Le principe de l'algorithme OEP est décrit comme suit :

- dans un espace  $R^n$  , au moment  $t$ , chaque particule  $i$  a une position  $\vec{x}_i^t$  . Elle se déplace avec une vitesse  $\vec{v}_i^t \in R^n$  selon certains paramètres comme sa perception de l'environnement, l'alignement et la direction des voisins.
- la vitesse et la position de la particule  $i$ , au moment  $t$ , sont mises à jour conformément aux règles suivantes :

$$\vec{v}_i^{t+1} = c_i \vec{v}_i^t + c_p (p_i^t - \vec{x}_i^t) + c_g (g^t - \vec{x}_i^t) \quad (5.1)$$

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1} \quad (5.2)$$

avec  $p_i$  et  $g$  respectivement la position du meilleur voisin de  $i$  et la position de la meilleure particule du groupe.  $c_i$  ,  $c_p$  and  $c_g$  : des nombres aléatoires (coefficients de confiance) distribués uniformément dans  $[0,1]$ . Ils dénotent des coefficients d'accélération de la particule  $i$ , ses voisins et de la meilleure particule du groupe.

### 5.2.4 Algorithme des chauves souris

L'algorithme des chauves-souris, dont l'appellation d'origine est "Bat Algorithm" en anglais, est une méta-heuristique très récente. Le premier article la concernant a été proposé en 2010 par Xin-She Yang [Yan10]. Elle est basée sur le comportement d'écholocation des chauves-souris. Spécifiquement, l'espèce des micro-chiroptères, qui se dirigent en vol par écholocation, peuvent trouver, discriminer les différents types d'insectes et éviter les obstacles.

L'optimisation par la technique des chauvesouris (BAT) est une approche inspirée du comportement de chasse des chauve souris. Au cours de leur vol et, afin d'éviter les obstacles et cibler leur proies, chaque individu (bat) émet un bisonar [Yan10] à travers son environnement ; le retour de l'écho permet d'identifier les divers objets de son entourage. Les études développées dans ce domaine ont montré que la résonance de l'onde émise varie d'une valeur élevée au cours d'un vol de prospection à une valeur assez faible avec une augmentation de fréquence quand la chauve souris détecte et oblique en direction d'une proie. Il est supposé par intuition que ces animaux sont en mesure de différencier leurs proies des autres obstacles proches, notamment les chauves-souris avoisinants.

Similaire à l'OEP, l'algorithme BAT peut être implémenté pour la résolution de problèmes d'optimisation continue où les solutions possibles peuvent être représentées par les positions géographiques des bats.

Dans l'algorithme "BAT Algorithm", les individus sont indépendants les uns des autres dans leur mouvement. Son inconvénient est le manque de communication entre les individus, les obligeant ainsi à chercher le meilleur chemin, Chacun à sa manière sans toutefois bénéficier de l'expérience des voisins et par conséquent, consomme plus de ressources inutiles.

## 5.3 SHEM : Sensor Hybrid Energy Mobility

Nous proposons l'algorithme SHEM dont l'étude est naturellement menée dans le cadre de quelques hypothèses réalistes qui permettent d'en limiter le périmètre.

1. Chacun de nos capteurs est mobile et possède deux antennes. Une antenne pour la capture des données, une autre antenne de fréquence permettant d'estimer la position et la vitesse du capteur pour permettre de positionner le capteur ou les autres capteurs pour une bonne connectivité.
2. Tous les capteurs ont le même rayon de communication et même sensibilité en réception donc les liaisons radio sont symétriques (il est impossible de trouver pratiquement des nœuds qui ont cette caractéristique donc c'est une hypothèse de travail).
3. Le modèle de couche physique est idéal. Un nœud destinataire  $C_i$  reçoit toujours sans erreur un message d'un nœud émetteur  $C_j$  quand il se trouve à sa portée, c'est à dire si la distance  $d(C_i, C_j) \leq R_c$ , avec  $R_c$  : rayon de communication, valeur comprise entre 1 et 100 mètres avec les technologies considérées.

Nous allons utiliser l'algorithme "Bat" tout en ajoutant la construction de notre topologie du réseau par l'algorithme du chapitre 4. Le calcul des distances effectué par l'algorithme de chaque cluster servira à orienter les capteurs lorsqu'ils s'écartent ou deviennent proches pour maintenir la connectivité.

### 5.3.1 Description générale de la méthode

Notre première contribution a consisté à la mise en place des clusters et la désignation des clusters Head par la station de base. Il a permis à chaque capteur de s'auto-évaluer en énergie pour continuer à résider dans le cluster et au cluster head de permettre une transition de son rôle à un autre avec moins d'énergie possible à perdre dans la construction du clustering et dans le choix du cluster head. Notre seconde contribution utilise la mise en place de ces clusters pour permettre aux capteurs de se déplacer aléatoirement et de façon imprévisible pour minimiser le délai d'acheminement, le taux de pertes, la consommation d'énergie et la connectivité entre les capteurs du réseau. Un nœud peut envoyer une information au cluster auquel il appartient en utilisant la distance maximale définie dans notre première contribution. Ceci est possible lorsque le capteur se rend compte que sa distance parcourue s'approche de cette distance maximale qui est signe de l'éloignement du capteurs des autres capteurs. Le capteur émetteur pourra le faire à un taux d'amplification élevé. Ce message permettra aux autres capteurs du clustering d'adapter leur vitesse et direction de déplacement pour rester en contact permanent et assurer la connectivité du

réseau avec le capteur émetteur. Lorsqu'il découvre une zone d'intérêt ou veut transmettre des données, il utilise une amplification faible. S'il y a un changement, il envoie un message à tous les clusters head concernant sa nouvelle position pour une mise à jour de leurs données. Notre algorithme est résumé par la figure 5.1.

En fait, nous nous basons sur l'écho-localisation au niveau de la deuxième antenne qui est un système de repérage utilisé par certains animaux comme les chauves-souris pour permettre de garder la connectivité dans nos clusters.

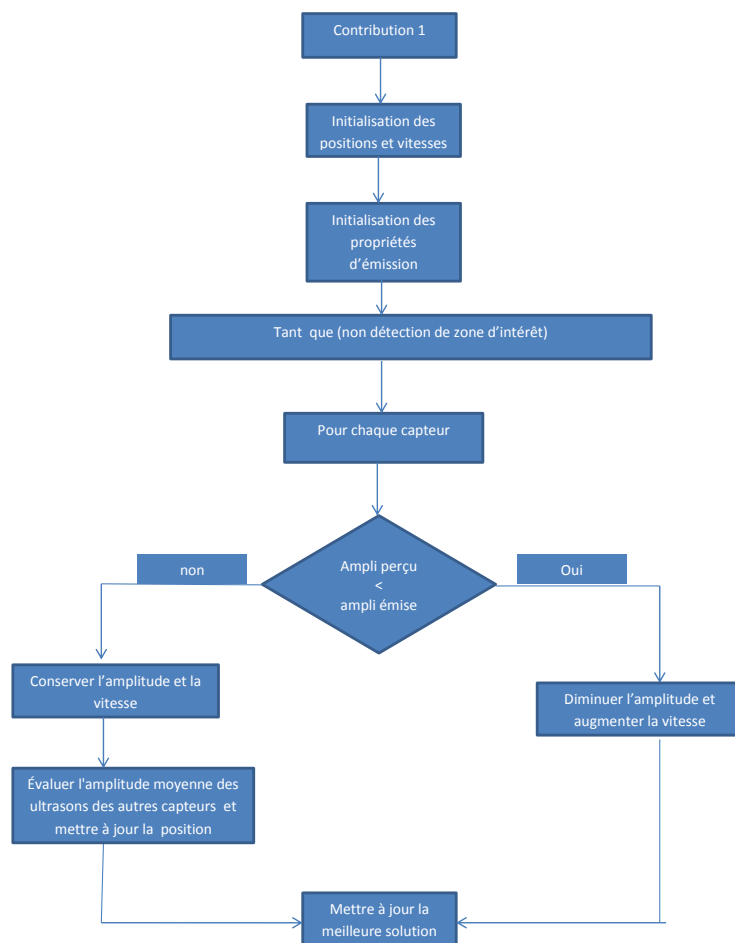


FIGURE 5.1 Algorithme assurant la topologie dynamique

### 5.3.2 Notre algorithme détaillé

Notre algorithme part du principe que le capteur peut suivre son déplacement naturel et surtout demander un déplacement. Le déplacement de chaque capteur est libre ou peut être guidé par d'autres capteurs. Il se déplace selon son mouvement actuel ou change brusquement de direction comme le déplacement d'une chauve-souris guidé par sa proie et les sons émis par d'autres chauves souris. Ce déplacement peut entraîner le mouvement des autres capteurs (les voisins, les CH) en fonction de l'intérêt des données captées, la distance parcourue et le flux des données. A chaque instant  $t$ , chacun des  $N$  capteurs du cluster possède une position  $\vec{x}_i$  et une vitesse  $\vec{v}_i$ . Au cours de ses déplacements, chaque capteur émet des ultrasons avec une puissance  $P_i \in [P_{min} : P_{Lmax}]$  à une fréquence  $f_i \in [f_{min} : f_{max}]$ . Les émissions d'ultrasons s'effectuent en rafale selon le taux d'impulsions  $T_i \in [0 : 1]$ . Lorsque le nœud détecte une zone d'intérêt, il émet plus fréquemment des ultrasons avec une puissance faible ( $T_i$  grand et  $P_i$  faible). Inversement, lorsque le capteur estime sa distance loin des autres sans détection d'une zone d'intérêt, les émissions sont moins fréquentes mais plus puissantes pour permettre d'appréhender les positions du CH et de son voisin (pour permettre une jonction par le déplacement des autres nœuds ou demandés son repositionnement par rapport aux autres nœuds).

#### Variables

$P_{min}$  : puissance d'émission minimale

$P_{max}$  : puissance d'émission maximale

$P_i$  : puissance d'émission d'un capteur  $i$

$T_i$  : taux d'impulsion pour un capteur  $i$

$f_i$  : fréquence d'émission d'un capteur  $i$

$\vec{V}_i$  : vitesse du capteur  $i$

$\vec{X}_i$  : position du capteur  $i$

$D$  : direction en angle  $\alpha$  : facteur de réduction de la puissance d'émission

$\gamma$  : facteur de réajustement du taux d'impulsion

$U(|1 : N|)$  : fonction permettant de choisir une valeur dans l'intervalle 1 et  $N$ .

#### Initialisation

A l'initialisation, les capteurs sont par défaut répartis uniformément dans l'espace de recherche (champ de captage). C'est-à-dire la construction des clusters et désignation des CH par la station de base. La vitesse initiale est en général nulle. Dans la plupart des implémentations, la puissance d'émission est choisie dans l'intervalle  $[0 : 1]$ . C'est-à-dire  $P_{min} = 0$  et  $P_{max} = 1$ . La valeur initiale de la puissance d'émission et du taux d'impulsion est souvent fixée à une valeur proche de 0.5. Dans ce cas, un capteur a 50% de chance de choisir un déplacement aléatoire initialement et en cas d'amélioration de la meilleure solution, de mettre à jour ses propriétés d'émissions des ultrasons. Il est également possible d'initialiser  $P_i$  et  $T_i$  de façon aléatoire et différente pour chaque nœud.

## Déplacement des capteurs

Comme mentionné plus haut, le déplacement d'un capteur obéit à son mouvement actuel ou change de direction comme le mouvement d'une chauve souris. Pour poursuivre son mouvement actuel, on utilise un principe similaire à celui de l'optimisation par essaim. Dans cette optimisation, chaque chauve souris est donc positionnée (aléatoirement ou non) dans l'espace de recherche du problème. Chaque itération fait bouger les chauves souris en fonction de 3 composantes :

sa vitesse actuelle,

Sa meilleure solution  $P_i$ ,

La meilleure solution obtenue dans son voisinage  $P_g$ .

Cela donne l'équation de mouvement suivante [Cle05] :

$$\vec{V}_{k+1} = w * \vec{V}_k + b_1 * (P_i - \vec{X}_k) + b_2 * (P_g - \vec{X}_k) \quad (5.3)$$

$$\vec{X}_{k+1} = \vec{X}_k + \vec{V}_{k+1} \quad (5.4)$$

Avec : w inertie

$b_1$  tiré aléatoirement dans  $[0, \varphi 1]$

$b_2$  tiré aléatoirement dans  $[0, \varphi 2]$ .

Dans notre situation, la nouvelle vitesse est obtenue en sommant la vitesse actuelle et un vecteur de vitesse directionnelle externe. Dans le cas de la forme de base de l'algorithme, cette vitesse directionnelle externe est en général obtenue en multipliant la fréquence d'émission  $f_i$  et la direction entre la position actuelle et la position de la meilleure solution (obtenue en mettant le cluster en place). La fréquence d'émission est générée uniformément dans l'intervalle  $[f_{min} : f_{max}]$  et permet de contrôler le rythme du déplacement. Le déplacement s'effectue en ajoutant à la position courante la nouvelle vitesse représentée par l'équation (5.9). Pour changer de direction, la position est obtenue à partir de la position actuelle d'un capteur choisi aléatoirement (le cluster head ou le capteur voisin). Cette opération est représentée par l'équation (5.5). Cette position subit une perturbation aléatoire proportionnelle à la puissance moyenne des ultrasons émis par tous les capteurs. Les équations (5.5) et (5.6) résument ces formules. Pour contrôler le choix de telle ou telle stratégie de mouvement, le taux d'impulsion  $T_i$  est utilisé. Plus ce taux est élevé, plus le capteur exploite sa vitesse actuelle ainsi que l'information externe pour définir sa nouvelle position. Plus le taux est faible, plus les mouvements aléatoires sont autorisés. Ce taux joue un rôle similaire à celui de la température dans le recuit simulé et détermine un équilibre entre l'exploitation des positions actuelles et l'exploration de l'espace. C'est ce qui est exprimé par les équations (5.7),(5.8) et (5.9) . La phase de déplacement est résumée par l'algorithme suivant :

## Mise à jour des propriétés d'émission des ultrasons

Si la position solution est meilleure que la meilleure solution connue actuellement alors la mise à jour des propriétés d'émission des ultrasons est effectuée en fonction de la probabilité de la puissance  $P_i$ . Dans ce cas, la puissance d'émission est réduite par un facteur  $\alpha \in ]0 : 1[$  et le taux d'impulsion  $T_i$  est augmenté et calculé selon la formule

---

**Algorithm 1** algorithme de déplacement des capteurs

---

Pour i allant de 1 à N faire

Si  $U([0 : 1]) > t_i$  alors

$$K = U(|1 : N|) \quad (5.5)$$

$$\vec{x}_i = \vec{x}_k + \frac{\sum_{j=1}^N P_j(t-1)}{N} * U(|-1 : 1|)^D \quad (5.6)$$

sinon

$$f_i = U(|f_{min} : f_{max}|) \quad (5.7)$$

$$\vec{v}_i = \vec{v}_i + (\vec{x}_i - x^*) * f_i \quad (5.8)$$

$$\vec{x}_i = \vec{x}_i + \vec{v}_i \quad (5.9)$$

Fin Si

Fin Pour

---

(5.11) avec t le numéro de l'itération et comme  $\gamma \in ]0 : 1[$ . La diminution progressive de  $P_i$  provoque une diminution de la probabilité d'augmenter  $T_i$ . L'augmentation de  $T_i$  diminue la probabilité de recourir à un déplacement aléatoire des capteurs et permet d'augmenter progressivement l'exploitation des solutions aux dépens de l'exploration de l'espace. Les paramètres  $\gamma$  et  $\alpha$  sont en général choisis égaux et ont pour valeur 0.9. L'étape de la mise à jour est donnée par l'algorithme suivant.

---

**Algorithm 2** Mise à jour des propriétés d'émission des ultrasons

---

Pour i allant de 1 à N faire

Si  $(U[0 : 1]) < P_i$  et  $f(x_i) < f(x^*)$  alors

$$P_i = \alpha * P_i \quad (5.10)$$

$$T_i = T_{max}(1 - e^{-\gamma * t}) \quad (5.11)$$

fin si

Finpour

---

### Mise à jour de la meilleure solution

La mise à jour de la meilleure solution qui optimise la connectivité s'effectue classiquement en conservant la meilleure position des capteurs. A ce niveau, les capteurs étant en mouvement, les capteurs peuvent changer de voisin immédiat nécessaire en cas de chute d'énergie. Donc le rôle de cette partie sera de calculer la position, la mise à jour des coordonnées du voisin immédiat, du CH et en essayant de conserver la topologie d'initialisation. Nous allons expérimenter l'algorithme SHEM et l'algorithme Bat.

**Algorithm 3** Mise à jour de la meilleure solution

---

 Pour  $i$  allant de 1 à  $N$  faire

 Si  $f(x_i) < f(x^*)$  alors

 $x^* = x_i$ 

Finsi

 Finpour
 

---

## 5.4 Simulation

Pour simuler notre algorithme, nous avons utilisé le simulateur des systèmes multi agents NetLogo. Nous nous sommes basés sur l'exemple Flocking de NetLogo pour effectuer la programmation de notre algorithme. On suppose que chaque capteur est considéré comme un agent dans notre simulation. La figure 5.2 présente notre interface de simula-

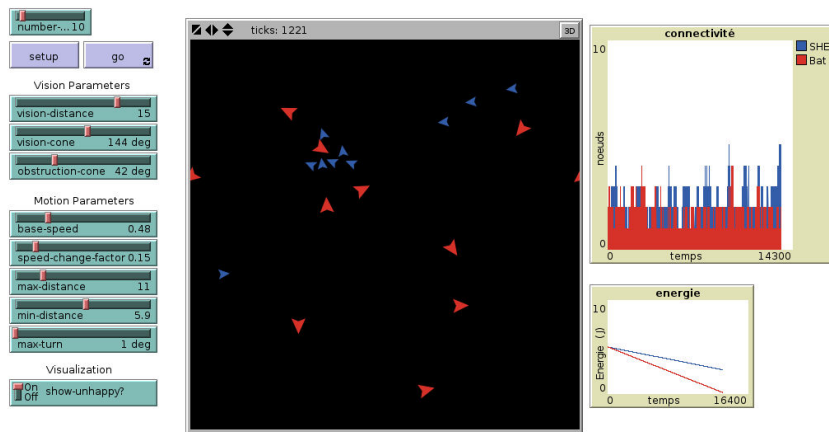


FIGURE 5.2 Interface du simulateur

tion.

### 5.4.1 Déplacement

Les chauves-souris émettent des sons qui sont renvoyés par les objets qui les entourent sous forme d'un écho. A partir de cet écho qu'elles entendent, les chauves-souris peuvent créer une image mentale de leur environnement tout comme nous le faisons à l'aide de nos yeux qui reçoivent la lumière réfléchiée par les objets qui nous entourent.

Les sons utilisés par les chauves-souris sont des ultra-sons, c'est-à-dire qu'ils sont trop aigus pour que l'oreille humaine les entendent.

L'écholocation nécessite d'autre part la faculté d'émettre des ultrasons très puissants de manière très précise ce qui est très coûteux en énergie. En vol de chasse les chauves-souris émettent des ultrasons de façon synchronisée avec leurs battements d'ailes à intervalle de temps régulier, ce qui permet d'économiser de l'énergie ; elles émettent donc . Lorsqu'elles ont repéré un insecte, plus rapidement et en direction de la proie ce qui leur permet de l'appréhender de façon plus précise. A l'inverse, lorsqu'elles parcourent un terrain connu, par exemple à la sortie de leur gîte, elles n'émettent pas à chaque battement

d'ailes mais plutôt une fois tous les deux ou trois battements. Dans notre simulation, les nœuds auront à un temps donné  $t$  une vitesse initiale. Pour matérialiser la communication entre les capteurs, nous avons suggéré une période où chaque capteur émet un message et les autres capteurs adaptent leur déplacement et direction. Il s'agit d'une synchronisation du temps permettant à chaque capteur d'avoir la possibilité de communiquer sa position, ses difficultés et la tâche qu'il est entrain d'accomplir.

Nous avons comparé l'algorithme SHEM à l'algorithme original "Bat" de déplacement des chauves souris.

### 5.4.2 Paramètres de simulation

Au cours de nos simulations, il a fallu déterminer des ensembles de valeurs pour les paramètres suivants :

- le nombre de nœuds ou agents doit varier dans le réseau en supposant qu'en mobilité et en écho-localisation, beaucoup de nœuds entraînent une consommation élevée d'énergie. On limitera le nombre de nœuds à trente avec des nœuds fixes, mobiles et injection d'autres nœuds.

- la vitesse des nœuds doit varier et être aussi constante pour voir l'effet sur la consommation d'énergie par exemple.

- la distance entre les nœuds doit varier en estimant qu'en écho-localisation, la distance maximale est de cinq (5) mètres. La distance entre les nœuds varie entre un et cinq mètres.

- la direction du déplacement est celle du nœud émetteur.

Il a fallu aussi combiner ces paramètres pour avoir des expérimentations représentatives.

### 5.4.3 Mesures de performance

Les indicateurs choisis pour évaluer l'algorithme proposé peuvent être :

- Nombre de nœuds déconnectés ;

- Le taux de connexion en se référant au pourcentage de nœuds connectés ;

- La distance moyenne entre les nœuds ;

- Le nombre moyen de voisins ;

- La consommation d'énergie du réseau dans son ensemble.

En mesurant le nombre de nœuds déconnectés au fil du temps et le taux de connexion, il est possible d'évaluer directement la qualité de la proposition et de répondre à son objectif principal, à savoir maintenir la connectivité.

Dans les mesures de connectivité, un capteur mesure les capteurs qui sont dans sa portée de transmission. En d'autres termes, soit un capteur se trouve dans la zone de couverture d'un capteur donné, ou soit il est en dehors de sa portée de transmission.

Les deux indicateurs de taux et de distance donnent un aperçu sur la façon dont les nœuds sont répartis et comment l'algorithme proposé influence le mouvement des nœuds. De bons résultats pourraient être obtenus en utilisant des distances pas trop grandes entre les nœuds pour que leur connectivité ne puisse pas être cassée. Cela signifie qu'en utilisant la solution proposée les nœuds peuvent s'écarter les uns des autres, mais pas trop. Cependant, les nœuds ne doivent pas garder une trop petite distance entre eux, ce



qui impliquerait une concentration élevée de nœuds, qui n'est pas souhaitable non plus. Le même raisonnement est valable pour le quatrième indicateur, dans lequel un très petit nombre de voisins, ou pas, signifierait que les nœuds soient très clairsemés puis très enclins à se déconnecter. Cependant, un nombre trop important de voisins signifierait que les nœuds soient trop concentrés, ce qui n'est pas souhaitable.

Les capteurs sont distribués au hasard sur la zone au début de la simulation et mis en zone comme au chapitre 4. Une fois que la simulation est lancée, les nœuds commencent à bouger comme proposé dans l'algorithme.

Dans chaque scénario, on présente quatre courbes. Les deux premières comparent "Bat" et "SHEM" sans influence sur le déplacement. Les deux autres comparent "Bat" et "SHEM" avec influence sur le déplacement. On appelle ce dernier SHEM-D.

#### 5.4.4 Premier scénario

Dans ce premier scénario, nous avons considéré dix (10) nœuds qui vont être mobiles avec la proposition de déplacement des chauves-souris. Ce scénario doit nous permettre de mesurer le nombre de nœuds déconnectés et le taux de connectivité.

##### A vitesse fixe et distance fixe

Après plusieurs simulations d'environ 250 ms et à 250 reprises, la figure 5.3 a été obtenue avec une vitesse constante pour tous les nœuds. Elle montre le pourcentage de noeuds

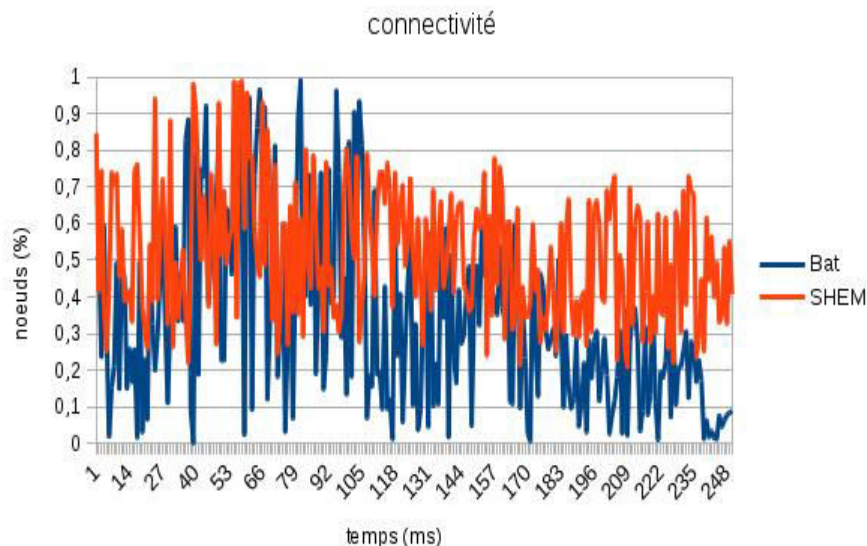


FIGURE 5.3 Schéma de connectivité

connectés entre l'algorithme "Bat" et l'algorithme "SHEM" au cours de l'expérience. Les différents creux s'expliquent par l'écartement des noeuds. La courbe "Bat" présente entre 15 et 22 ms, entre 98 et 100 ms un pourcentage presque nul exprimant l'isolement de tous

les noeuds du réseau. Entre 239 et 250 ms, cette courbe donne un très faible taux s'expliquant par la mort des noeuds du réseau. Cette perte de noeuds est exprimée par toutes les courbes en présentant une perte progressive du taux de connexion dans l'ensemble. Par contre, on constate une forte connectivité au fil du temps par la courbe SHEMA dont le plus faible taux dépasse 10%.

La courbe 5.4 présente l'évolution de l'énergie dans la globalité. On constate que l'algorithme "SHEMA" consomme moins que l'algorithme "Bat" au fil du temps. Cela s'explique par la réduction de la distance d'envoi et aussi le maintien de la connectivité. 5.4. Cette courbe est régressive et explique la diminution de l'énergie au cours de la

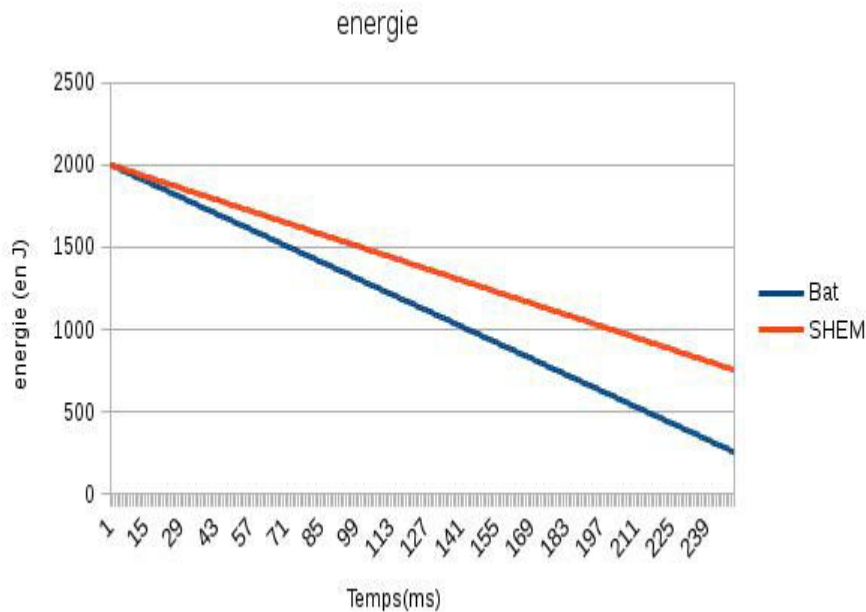


FIGURE 5.4 Schéma de la consommation d'énergie

vie du réseau. Elle montre une baisse considérable pour "Bat". Cela peut s'expliquer par l'isolement des noeuds et une communication à distance considérable. Cette courbe montre un déplacement aléatoire qui consomme plus d'énergie que l'algorithme SHEMA.

En prenant en compte le changement de direction, on obtient les figures 5.5, 5.6 représentant la connectivité et l'évolution de l'énergie au cours de la vie du réseau. La prise en compte de la direction montre une augmentation de la connectivité et aussi un léger gain en énergie qui passe de 870 (SHEMA) à 905 (SHEMA-D).

Cela s'explique par un nombre de communication élevée et aussi avec la diminution de la distance de communication.

### A vitesse progressive et distance variable

Nous avons poussé le premier scénario plus loin en modifiant la vitesse de chaque capteur en fonction de la vitesse du capteur émetteur de message. A l'ancienne vitesse des

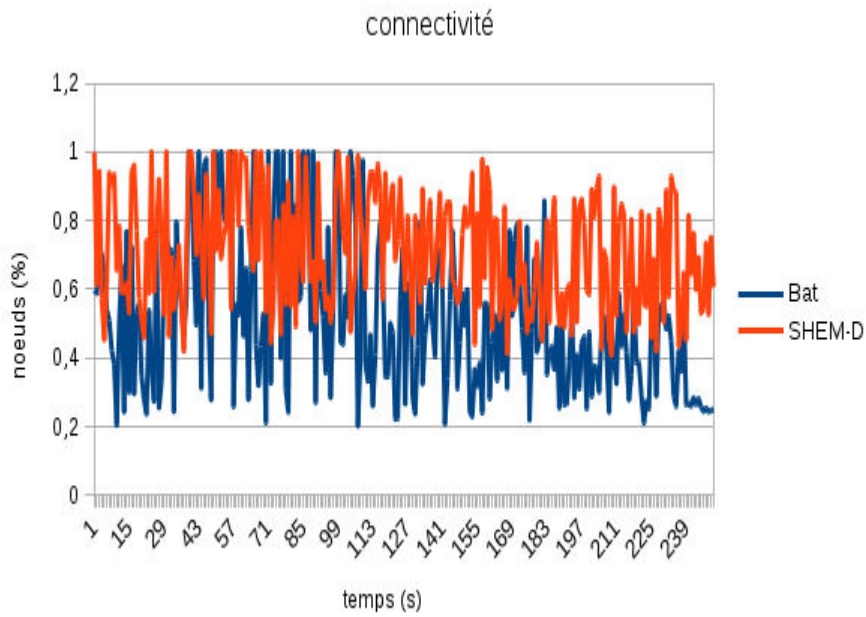


FIGURE 5.5 schéma de connectivité

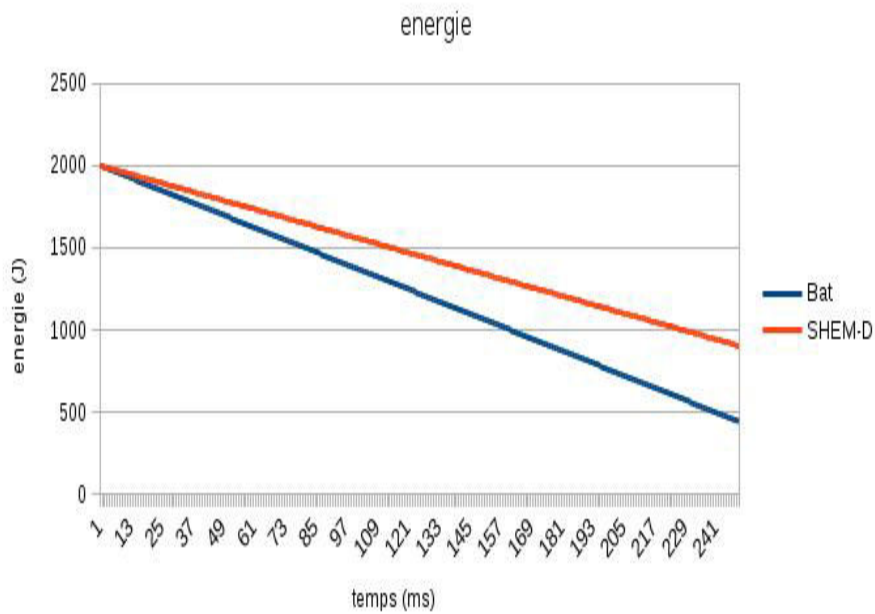


FIGURE 5.6 schéma d'énergie

capteurs s'ajoute la vitesse du capteur émetteur. Ce qui a permis d'avoir une vitesse dynamique avec une limitation de vitesse maximale. Avec cette donne, le taux de connectivité a augmenté. La figure 5.7 illustre cette connectivité. Les creux de la courbe s'expliquent

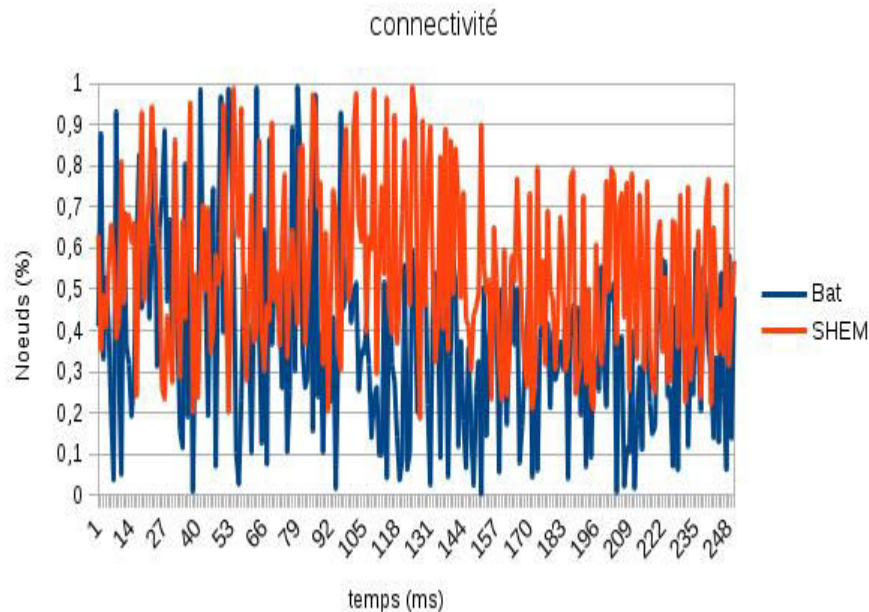


FIGURE 5.7 Schéma de connectivité à vitesse et distance variable

par l'écartement de certains noeuds au cours de la vie du réseau. Ces noeuds s'éloignent grâce à leur mouvement individuel. On constate une augmentation du taux minimal de connectivité par rapport à la première expérience. Ce qui signifie que la vitesse joue un rôle considérable dans la connectivité. Le schéma présente une nette amélioration de la connectivité avec "SHEM" par rapport à "Bat". La gestion de la vitesse prouve par ce schéma que la connectivité peut être maintenue en jouant sur ces paramètres. Ce qui entraîne une diminution de la consommation d'énergie exprimée par le schéma de la [figure 5.8]. On constate une perte de connectivité exprimée par quelques pics entre 27 et 40 ms et aussi entre 84 et 90 ms.

Cette modification a beaucoup joué sur la consommation d'énergie exprimée par la figure 5.8 qui a diminué par rapport à l'hypothèse ci dessus où la vitesse est constante. Cela s'explique par une réduction rapide de la distance d'envoi des messages.

En prenant en compte la direction, on obtient les figures 5.9, 5.10 qui donnent presque les mêmes allures de connectivité et de consommation que précédemment. Avec cette configuration, la connectivité s'est encore améliorée avec une faible augmentation de gain d'énergie avec 1002 pour SHEM et 1060 pour SHEM-D.

En somme pour le premier scénario, la modification de la vitesse et de la direction des capteurs est un facteur de réduction de la consommation d'énergie en réduisant la distance d'envoi. Elle augmente la connectivité des noeuds.

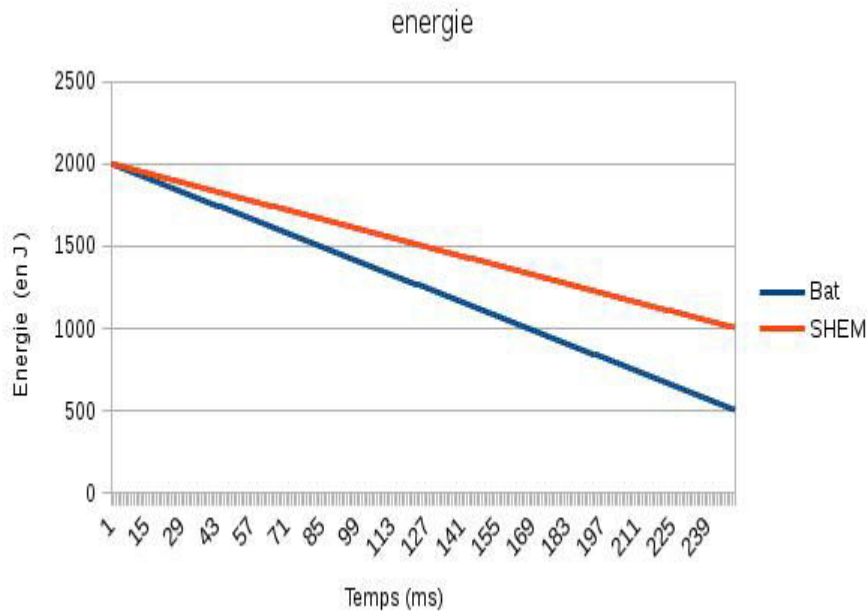


FIGURE 5.8 Schéma de consommation d'énergie

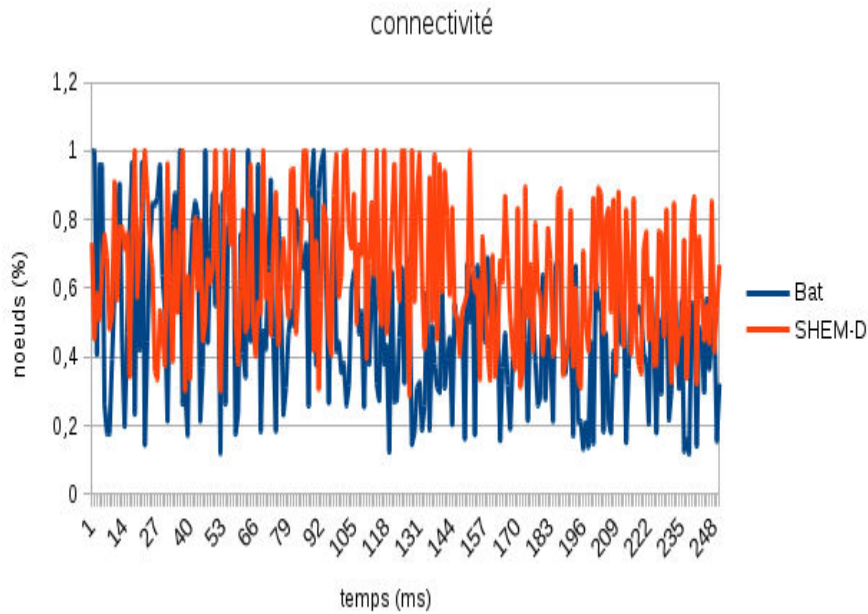


FIGURE 5.9 schéma de connectivité

### 5.4.5 Deuxième scénario

#### A vitesse fixe et distance fixe

Dans le deuxième scénario, nous avons considéré 20 nœuds et rendu 10 nœuds mobiles. En utilisant le même temps de simulation, on obtient la figure 5.11 avec un degré

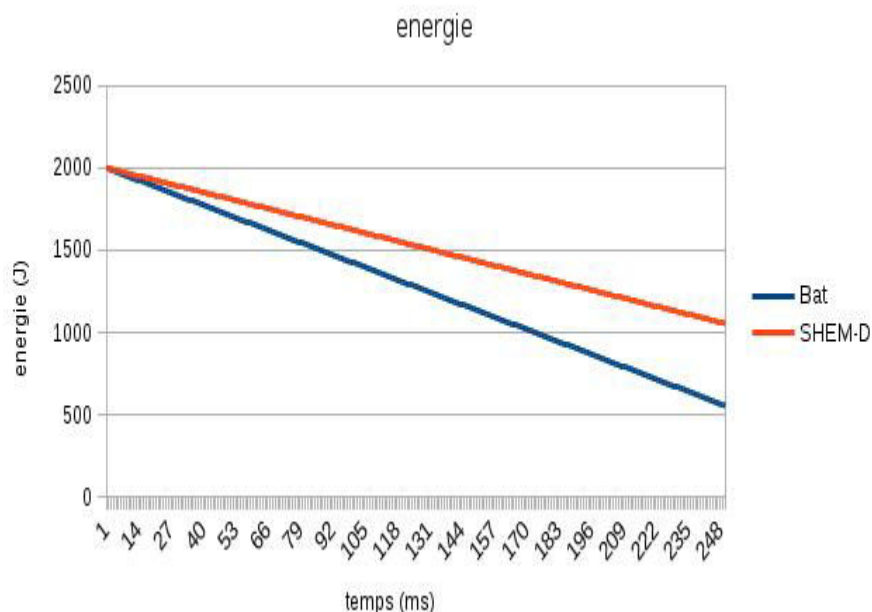


FIGURE 5.10 Schéma de consommation d'énergie

de connectivité plus important que le premier scénario. Cela s'explique par l'influence des nœuds fixes qui assurent une certaine connectivité avec les nœuds isolés. Les nœuds isolés se rapprochent des autres nœuds grâce aux nœuds fixes qui assurent le rôle de relais entre les nœuds isolés et les nœuds mobiles. On remarque également des creux expliquant l'isolement de certains nœuds dans le réseau. Au niveau de cette figure, nous remarquons une augmentation du taux de connectivité minimal et un taux maximal avoisinant les 100% pour "SHEMA". "Bat" présente toujours quelques pics de pertes de connexion entre tous les nœuds.

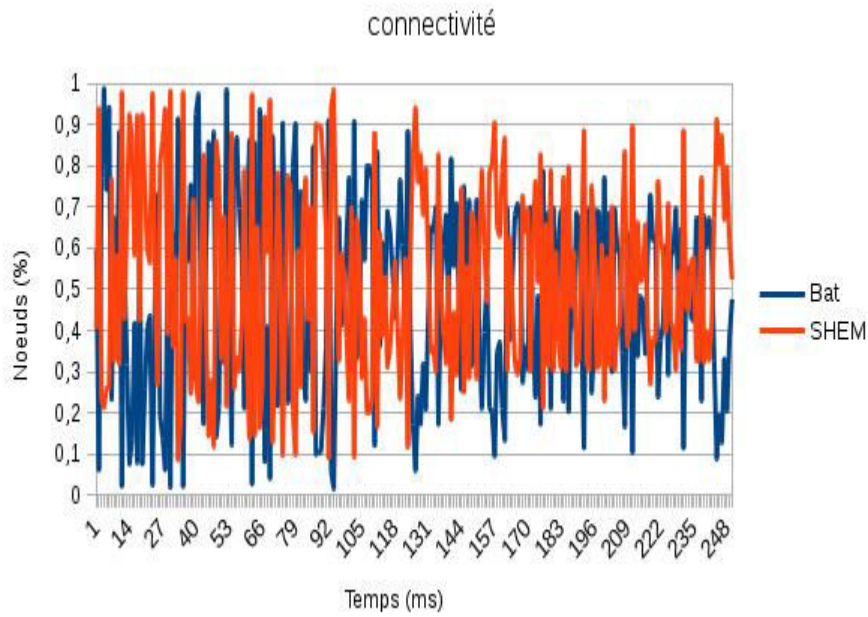
En ce qui concerne la consommation, la figure 5.12 nous montre une baisse de la consommation d'énergie pour les deux algorithmes. Plus le nombre de capteurs est important, plus le nombre de communications est élevé pour assurer la connectivité. L'expérience s'est poursuivie avec la prise en compte de la direction et qui aboutit aux schémas des figures 5.13, 5.14. Ces courbes montrent l'intérêt de la direction sur la consommation et la connectivité du réseau. Le taux de connectivité a augmenté et la consommation d'énergie a baissé. SHEMA est de 1004 et SHEMA-D est de 1078.

Les capteurs fixes arrivent à assurer la connexion mais aussi raccourcissent les distances de communication pour diminuer la consommation d'énergie. Comme précédemment, on constate que "SHEMA-D" économise l'énergie mieux que "SHEMA" et "Bat".

### A vitesse progressive et distance variable

Nous avons poursuivi cette expérimentation en faisant varier la vitesse des capteurs comme dans le premier scénario.

Ce qui nous a permis d'avoir la figure 5.15 mentionnant un taux de connectivité im-



!h]

FIGURE 5.11 Schéma de connectivité

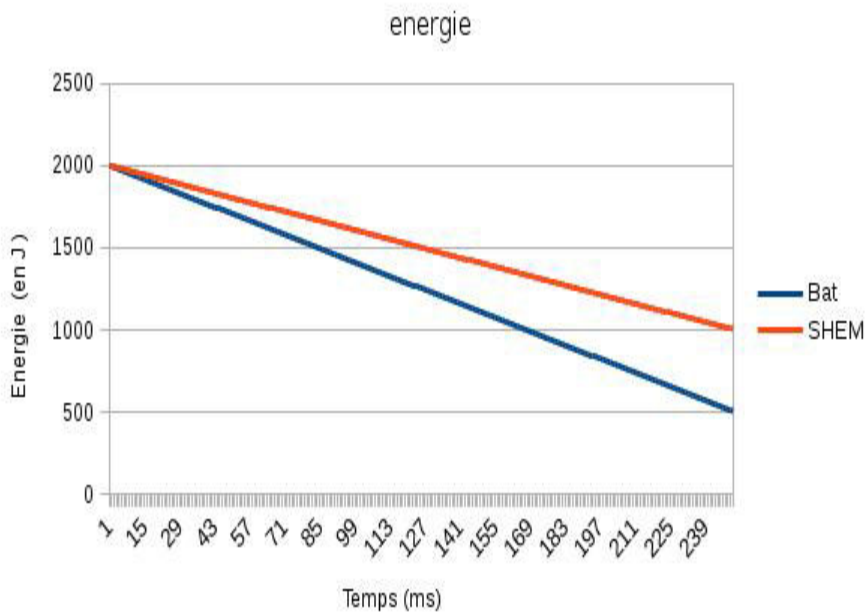


FIGURE 5.12 Schéma de consommation d'énergie

portant dans ce scénario. Ce taux s'explique par la variation de la vitesse qui permet de rapprocher les noeuds. Plus les capteurs sont proches, plus la consommation diminue du point de vue de la distance d'émission. Mais le nombre de communication augmente



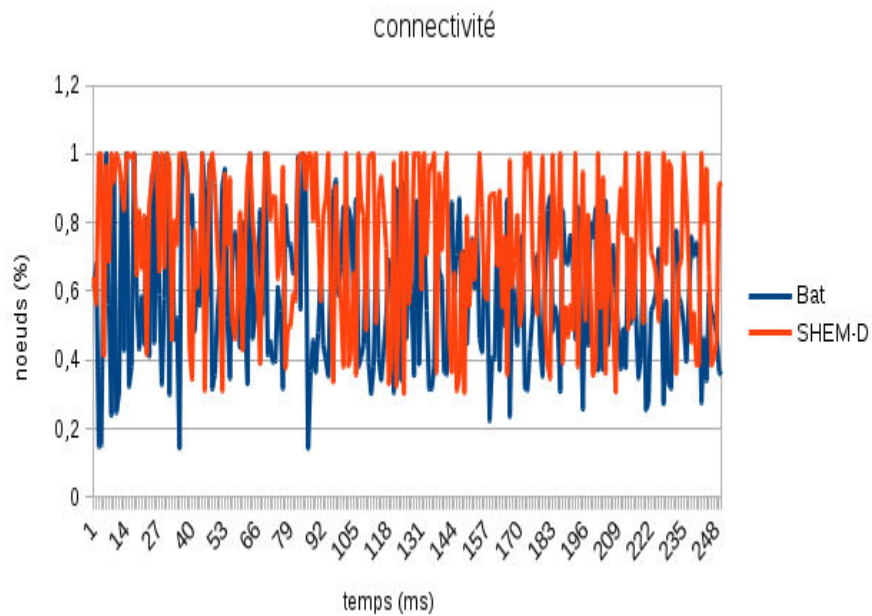


FIGURE 5.13 Schéma de connectivité

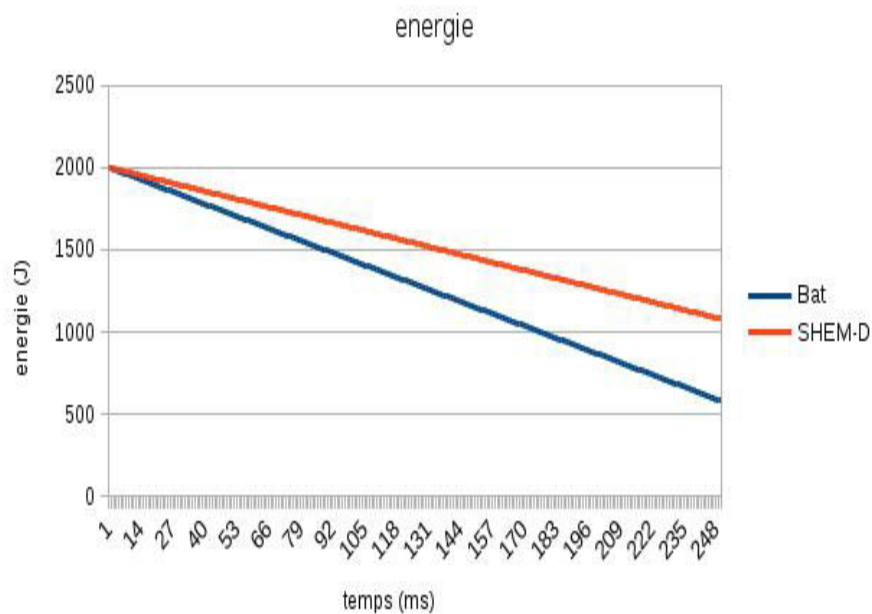


FIGURE 5.14 Schéma de consommation d'énergie

pour maintenir cette connectivité. Ce qui augmente la consommation globale des noeuds (figure 5.16).

La figure suivante 5.16 montre aussi une baisse de la consommation d'énergie dans le



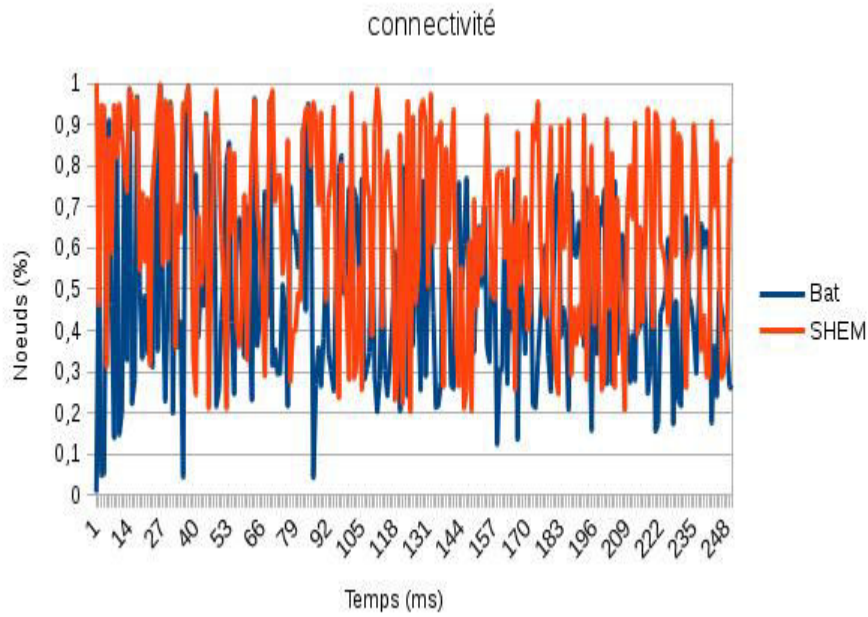


FIGURE 5.15 Schéma de connectivité à vitesse progressive et distance variable

réseau de capteurs à vitesse variable.

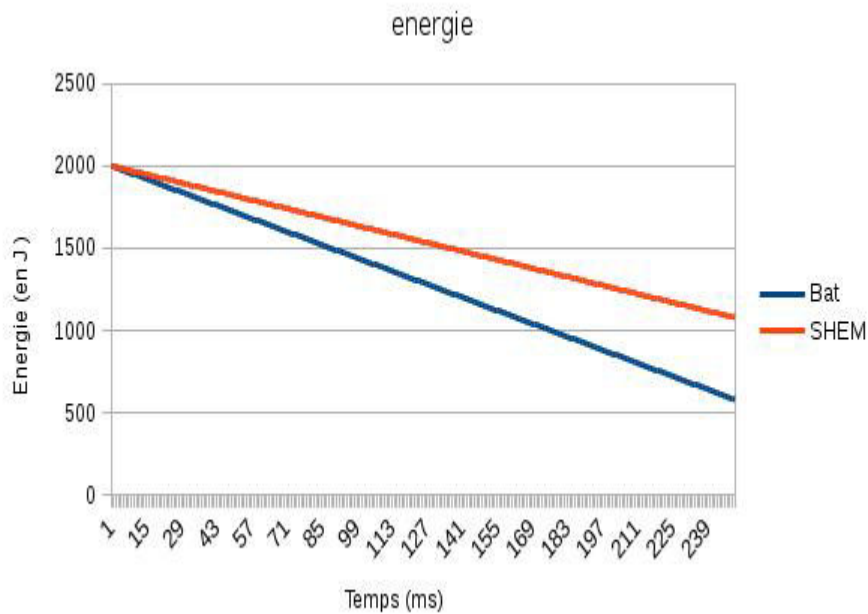


FIGURE 5.16 Consommation d'énergie à vitesse progressive et distance variable

Les figures 5.17, 5.18 prennent en compte la redirection des capteurs en mouvement.

Elles montrent une amélioration de la connectivité et de la consommation qui passe de 1005 pour SHEM à 1103 pour SHEM-D.

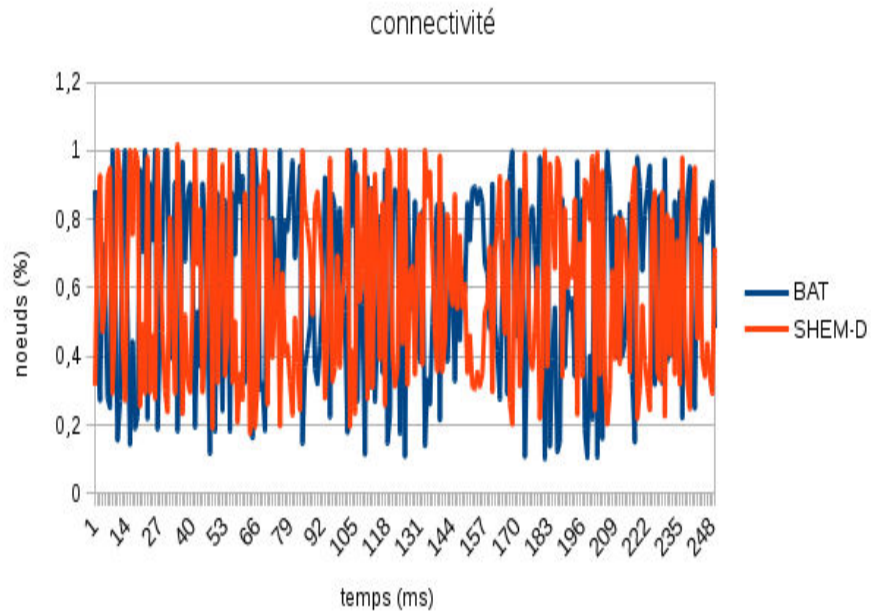


FIGURE 5.17 Schéma de connectivité

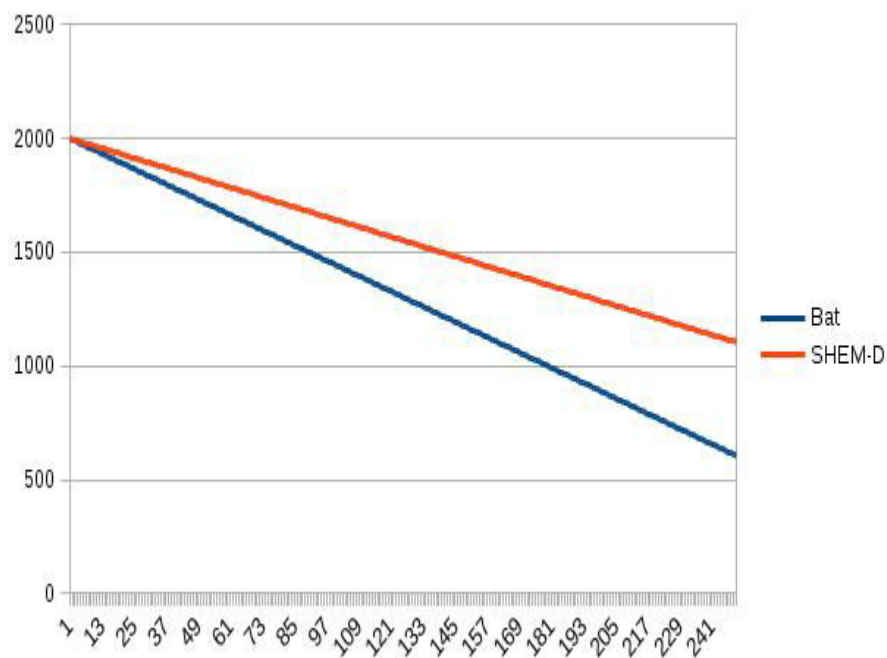


FIGURE 5.18 Consommation d'énergie

Nous constatons dans le deuxième scénario, que les noeuds fixes qui jouent le rôle de

relais ont amélioré le taux de connectivité et la consommation d'énergie par rapport au premier scénario

### 5.4.6 Troisième scénario

Dans ce scénario, nous avons décidé d'augmenter le nombre de capteurs. Dans les

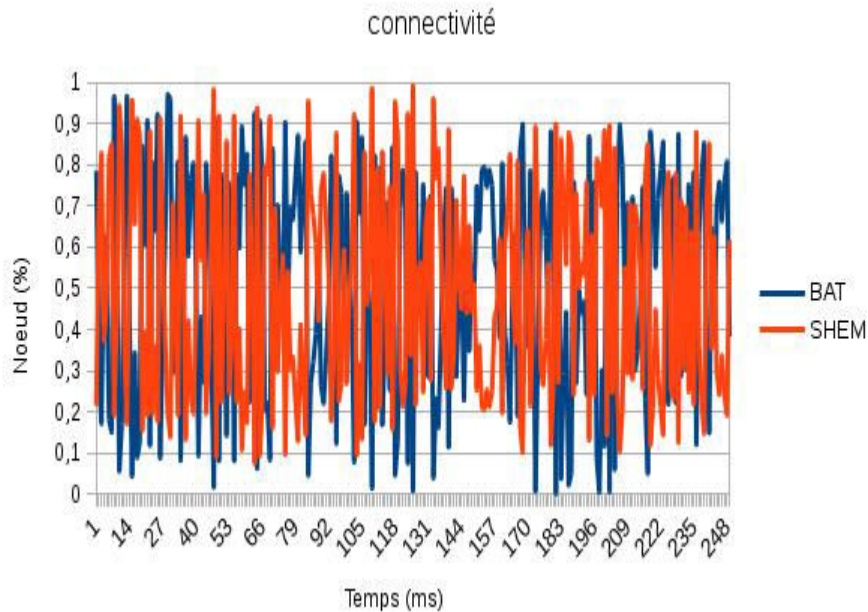


FIGURE 5.19 Schéma de connectivité après injection de nouveaux nœuds

mêmes conditions que les deux précédents scénarios, l'insertion de nouveaux capteurs au nombre de dix maintient et augmente davantage la connectivité du réseau. La figure 5.19 montre cette augmentation du taux de connectivité. Comme les autres scénarii, on constate l'isolement de certains nœuds du réseau par les creux des courbes. Malgré l'insertion des nœuds, on constate la perte de connectivité pour l'algorithme "Bat" par exemple entre 109 et 130 ms.

En outre, la figure 5.20 montre aussi une baisse régressive de l'énergie mais avec un gain d'énergie par rapport aux autres scénarii. Une augmentation est constatée vers les 170 ms et est due à l'insertion de nouveaux nœuds.

Les courbes des figures 5.21 et 5.22 sont obtenues dans les mêmes conditions en jouant sur la direction. Ce qui montrent davantage le rôle de la redirection sur la connectivité et la consommation d'énergie. Les valeurs de ces données ont augmenté de 1006 à 1130.

### 5.4.7 Bilan de la simulation

L'algorithme assure une bonne connectivité pour la transmission de l'information et des évènements mais consomme de l'énergie pour assurer cette connectivité. Il faut noter

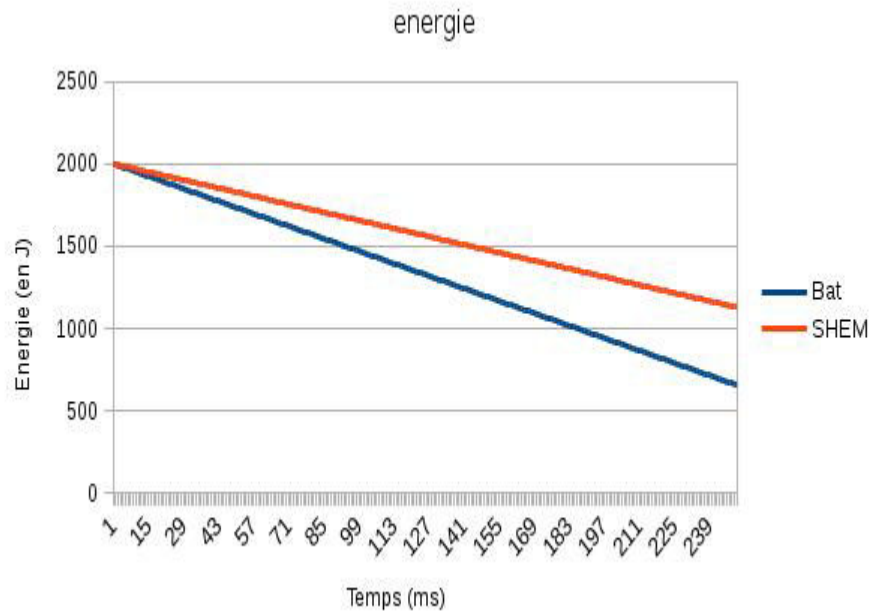


FIGURE 5.20 Schéma de consommation d'énergie par injection de nouveaux noeuds

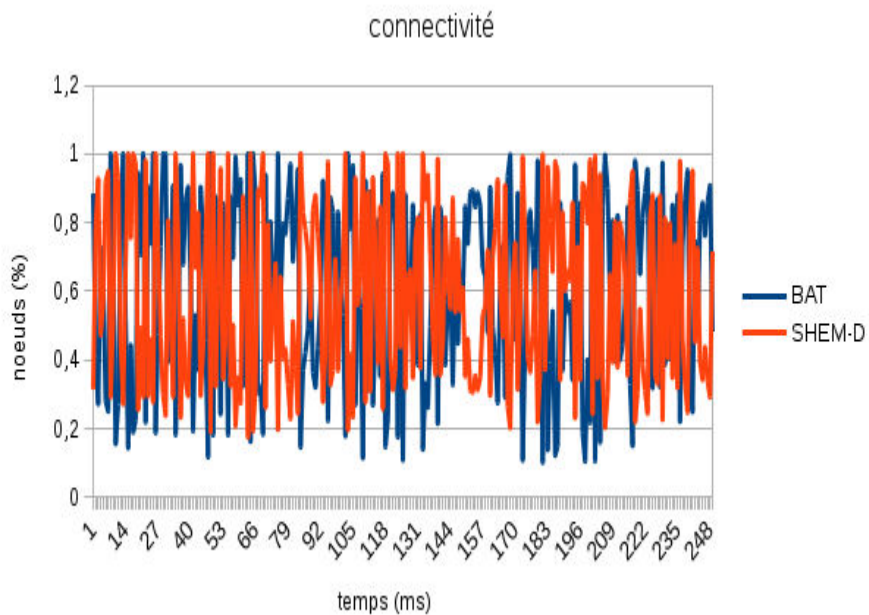


FIGURE 5.21 Schéma de connectivité

qu'une mobilité avec une augmentation de vitesse permet de réduire cette consommation d'énergie en réduisant la distance d'envoi. Et elle permet de réduire la consommation d'énergie des capteurs. Les relais fixes ont une importance pour la connectivité et réduisent

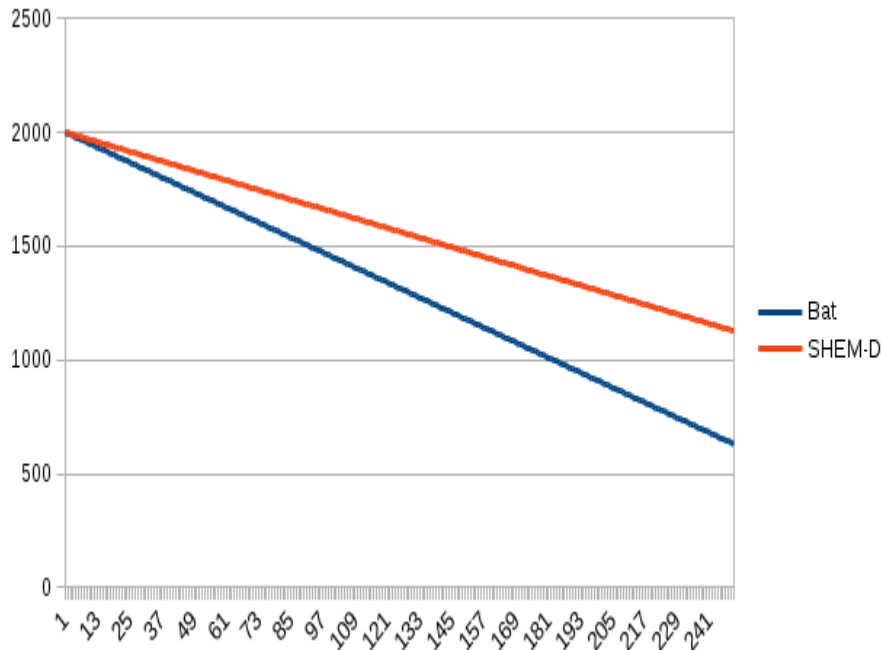


FIGURE 5.22 Schéma de consommation

le mouvement des noeuds mobiles en diminuant la distance de transmission des noeuds. Le renouvellement des noeuds du réseau est aussi capital pour le maintien de la connectivité et la réduction du taux de connectivité.

La solution avec SHEMA et SHEMA-D arrive à maintenir la connectivité et à réduire la consommation d'énergie du réseau plus que "Bat". Ce qui montre que cette solution est envisageable lorsque le maintien de la connectivité et la consommation d'énergie sont primordiales.

## 5.5 Conclusion

Dans le cadre de cette étude, nous nous sommes intéressés à la mobilité et à la connectivité en se basant sur le mode de vie des chauves souris. L'implémentation de notre algorithme sous le simulateur Netlogo nous a permis d'extraire quelques résultats par simulation et évaluer l'efficacité de notre approche d'écho localisation. Ces résultats montrent l'avantage du déplacement à l'aide de l'écho-localisation pour assurer la tolérance aux pannes, la connectivité dans un réseau de capteurs sans fil. L'exploitation de la mobilité de groupe des chauves souris permet d'étendre la durée de vie du réseau. Ce traitement se fait d'une manière distribuée. Ce but est atteint par la mise en œuvre d'un mécanisme de dynamique de la topologie, et le déplacement de quelques noeuds mobiles est indispensable. Pour conclure, nous pouvons noter, grâce à ces simulations, que les résultats obtenus montrent le bon comportement de notre algorithme et nous encouragent à orienter nos travaux de recherche futurs en favorisant l'utilisation des déplacements des chauves souris et la communication par écho localisation.

# Conclusion et Perspectives

Les réseaux de capteurs sont composés d'un grand nombre de dispositifs de communication autonomes, avec des ressources de calcul et d'énergie limitées. Ils font partie des technologies qui s'intègrent de plus en plus dans les applications du quotidien.

Cependant, les réseaux de capteurs sans fil rencontrent plusieurs problèmes qui affectent leur bon fonctionnement. Nous pensons ainsi à la limitation de la capacité de la batterie, au type de communication, aux environnements hostiles où sont déployés les capteurs, ... Par ailleurs, ces réseaux peuvent être affectés par des pannes parmi leurs nœuds, pannes qui peuvent causer un dysfonctionnement du réseau en entier.

Parmi les enjeux liés à ces réseaux de capteurs, nous nous sommes intéressés, classiquement, à la gestion de l'énergie, point sensible des réseaux de capteurs sans fil. Dans ce cadre, il faut également maintenir la connectivité, pour que le réseau remplisse son office. Il s'agit donc de maintenir un équilibre entre économie d'énergie et maintien d'une connectivité suffisante. Mais nous nous sommes penchés sur un autre problème, celui de la mobilité des capteurs. Il ne faut effectivement pas se limiter aux réseaux de capteurs statiques, ils ne correspondent pas toujours à des cas concrets. Nous pensons que la mobilité est un aspect très important permettant l'extension du domaine d'application des réseaux de capteurs sans fil. Ainsi, les capteurs peuvent être mobiles, naturellement car ils sont tributaires d'objets mouvants mais on peut imaginer qu'ils puissent influencer sur la position de leur porteur pour améliorer la connectivité et augmenter la durée de vie du réseau.

Dans cette thèse, notre travail a, ainsi, consisté à trouver des solutions pour augmenter la durée de vie du réseau en maintenant sa connectivité pour une meilleure rentabilité.

Pour nous permettre de proposer des solutions, nous avons étudié les réseaux de capteurs dans leur globalité, dans le chapitre 1. Nous avons prolongé l'état de l'art en nous focalisant sur les solutions existantes dans la littérature pour réduire la consommation d'énergie dans les réseaux de capteurs statiques et mobiles.

En analysant les problèmes liés à la consommation énergétique des réseaux des capteurs, nous avons ainsi proposé deux algorithmes dans les chapitres 4 et 5. Ils permettent de surmonter les limites des systèmes actuels. Le premier algorithme permet de construire des clusters en utilisant des zones de déploiement et en permettant aux capteurs de communiquer sur leur capacité en énergie. Le déploiement des capteurs impacte fortement les performances des applications implémentées, le choix de la stratégie de déploiement utilisée doit donc être en fonction des objectifs à atteindre et des contraintes spécifiques des capteurs déployés. Pour la mise en place de nos clusters, cet algorithme nous a permis de déterminer un nombre optimal de zones de déploiement. Il propose également une

désignation des cluster heads et permet à ceux-ci de désigner leur successeur après épuisement de leur énergie. Cette technique permet alors à ces cluster heads et aux capteurs de s'auto-évaluer pour le bon fonctionnement du réseau. Elle a permis d'économiser l'énergie dépensée dans la formation des clusters et de maintenir la connectivité des nœuds participant au réseau.

Nous avons appliqué notre algorithme proposé sur un réseau simulé, mais aussi sur une plate-forme réelle de capteurs (senslab). Dans la simulation et en comparaison avec d'autres algorithmes comme LEACH ou HEED, nous avons montré une amélioration de la consommation d'énergie par rapport aux techniques cités ci dessus.

Le deuxième algorithme s'est donc focalisé sur les capteurs en mouvement. Il s'appuie sur le premier algorithme mais il l'améliore en se basant sur le déplacement des chauves souris. Le déplacement par écho-localisation adapté à nos capteurs assure la tolérance au panne et la connectivité. Ces capteurs ont donc une capacité de déplacement leur permettant d'optimiser cette gestion du réseau. Nous avons testé cet algorithme par simulation. Les résultats ont montré son efficacité dans le maintien de la connectivité.

Les travaux effectués dans cette thèse ouvrent de nombreuses perspectives de recherche. En premier lieu, de nombreux paramètres de nos méthodes peuvent être étudiés plus en détail. Tester la faisabilité des solutions proposées sur des réseaux de capteurs semble être une étape évidente. En effet, les études par écholocalisation menées n'ont pour l'instant été validées que par simulations. L'application de notre algorithme sur des plateformes réelles pourrait être un facteur décisif pour l'évolution de ces dernières. Dans cette thèse, nos expériences se sont focalisées sur la réception et l'envoi des messages. Il faudra songer à poursuivre l'expérimentation de nos solutions en prenant en compte la consommation interne des capteurs, les collisions et les interférences. Une autre piste de recherche possible est le fait de provoquer la création de plusieurs chemins de routage topologiquement adaptés à chaque étape. En effet, les ordres de déplacement sont soit joints au paquet à router, soit inclus dans un beacon. Ces informations de déplacement sont reçues par la totalité du voisinage du nœud courant.

Comme cité ci dessus, les RCSFs connaissent une large utilisation dans plusieurs types d'applications et domaines dont ceux qui exigent une grande sécurité. En revanche, les caractéristiques et les spécificités (transmission radio en milieu ouvert, topologie dynamique, coopération des nœuds) des RCSFs constituent un véritable challenge pour assurer leur sécurité. Dans cette optique, nous proposons d'étudier cette problématique, et de proposer des solutions maximisant les performances de sécurités tout en réduisant la consommation d'énergie. Il pourrait être intéressant, également, d'explorer des méthodes permettant de combiner des systèmes de mobilité qui pourraient être antagonistes. Ainsi, des capteurs positionnés sur des drones pourraient avoir besoin d'une forme de mobilité incompatible avec celle des drones dans leur mission de couverture du terrain. Une piste de recherche intéressante serait donc de trouver une méthode permettant de concilier ces deux mobilités.

# Bibliographie

- [AALK12] Moshaddique Al Ameen, Jingwei Liu, and Kyungsup Kwak. Security and privacy issues in wireless sensor networks for healthcare applications. *Journal of medical systems*, 36(1) :93–101, 2012.
- [ACDFP09] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks : A survey. *Ad Hoc Networks*, 7(3) :537–568, 2009.
- [AKK04] Jamal N Al-Karaki and Ahmed E Kamal. Routing techniques in wireless sensor networks : a survey. *Wireless communications, IEEE*, 11(6) :6–28, 2004.
- [All04] ZigBee Alliance. Zigbee specification version 1.0, online at <http://www.zigbee.org>, 2004.
- [ASFI13] Aznita Abdul Aziz, Y Ahmet Sekercioglu, Paul Fitzpatrick, and Milosh Ivanovich. A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks. *Communications Surveys & Tutorials, IEEE*, 15(1) :121–144, 2013.
- [ASSC02a] Ian F Akyildiz, W Su, Y Sankarasubramaniam, and E Cayirci. Sensor networks. *Y, et al. A survey on IEEE Communications Magazine*, 40(8) :102–114, 2002.
- [ASSC02b] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *Communications magazine, IEEE*, 40(8) :102–114, 2002.
- [AV10] Ian F Akyildiz and Mehmet Can Vuran. *Wireless sensor networks*, volume 4. John Wiley & Sons, 2010.
- [AY05] Kemal Akkaya and Mohamed Younis. A survey on routing protocols for wireless sensor networks. *Ad hoc networks*, 3(3) :325–349, 2005.
- [BCD<sup>+</sup>05] Shah Bhatti, James Carlson, Hui Dai, Jing Deng, Jeff Rose, Anmol Sheth, Brian Shucker, Charles Gruenwald, Adam Torgerson, and Richard Han. Mantis os : An embedded multithreaded operating system for wireless micro sensor platforms. *Mobile Networks and Applications*, 10(4) :563–579, 2005.
- [BCM<sup>+</sup>08] Stefano Basagni, Alessio Carosi, Emanuel Melachrinoudis, Chiara Petrioli, and Z Maria Wang. Controlled sink mobility for prolonging wireless sensor networks lifetime. *Wireless Networks*, 14(6) :831–858, 2008.



- [BCSV06] Alvisé Bonivento, Luca P Carloni, and Alberto Sangiovanni-Vincentelli. Platform-based design of wireless sensor networks for industrial applications. In *Proceedings of the conference on Design, automation and test in Europe : Proceedings*, pages 1103–1107. European Design and Automation Association, 2006.
- [BFH<sup>+</sup>13] Mandicou Ba, Olivier Flauzac, Bachar Salim Hagggar, Rafik Makhloufi, Florent Nolot, and Ibrahima Niang. Energy-aware self-stabilizing distributed clustering protocol for ad hoc networks. *KSII Transactions on Internet and Information Systems (TIIS)*, 7(11) :2577–2596, 2013.
- [BM02] Suman Banerjee and Archan Misra. Minimum energy paths for reliable communication in multi-hop wireless networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 146–156. ACM, 2002.
- [BMJ<sup>+</sup>98] Josh Broch, David A Maltz, David B Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 85–97. ACM, 1998.
- [BMTM04] Sean M Brennan, Angela M Mielke, David C Torney, and Arthur B MacCabe. Radiation detection with distributed sensor networks. *Computer*, 37(8) :57–59, 2004.
- [BONL08] Azzedine Boukerche, Horacio ABF Oliveira, Eduardo F Nakamura, and Antonio AF Loureiro. Vehicular ad hoc networks : A new challenge for localization-based systems. *Computer communications*, 31(12) :2838–2849, 2008.
- [BR14] Maha Bouaziz and Abderrezak Rachedi. A survey on mobility management protocols in wireless sensor networks based on 6lowpan technology. *Computer Communications*, 2014.
- [BSH03] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. Important : A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 825–835. IEEE, 2003.
- [BSM<sup>+</sup>07] Yanzhong Bi, Limin Sun, Jian Ma, Na Li, Imran Ali Khan, and Canfeng Chen. Hums : an autonomous moving strategy for mobile sinks in data-gathering sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2007, 2007.
- [CBD02] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5) :483–502, 2002.
- [CBR04] Ian D Chakeres and Elizabeth M Belding-Royer. Aodv routing protocol implementation design. In *Distributed Computing Systems Workshops*,

- 
2004. *Proceedings. 24th International Conference on*, pages 698–703. IEEE, 2004.
- [CCGF09] Eugène Pamba Capo-Chichi, Hervé Guyennet, and J-M Friedt. K-rle : a new data compression algorithm for wireless sensor network. In *Sensor Technologies and Applications, 2009. SENSORCOMM'09. Third International Conference on*, pages 502–507. IEEE, 2009.
- [CCL03] Imrich Chlamtac, Marco Conti, and Jennifer J-N Liu. Mobile ad hoc networking : imperatives and challenges. *Ad Hoc Networks*, 1(1) :13–64, 2003.
- [CCV04] Sinem Coleri, Sing Yiu Cheung, and Pravin Varaiya. Sensor networks for monitoring traffic. In *Allerton conference on communication, control and computing*, pages 32–40, 2004.
- [CDGS07] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1) :5–16, 2007.
- [CG98] Tsu-Wei Chen and Mario Gerla. Global state routing : A new routing scheme for ad-hoc wireless networks. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 1, pages 171–175. IEEE, 1998.
- [CJA<sup>+</sup>03] Thomas Clausen, Philippe Jacquet, Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, Laurent Viennot, et al. Optimized link state routing protocol (olsr). 2003.
- [CK03] Chee-Yee Chong and Srikanta P Kumar. Sensor networks : evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8) :1247–1256, 2003.
- [Cle05] Maurice Clerc. *L'optimisation par essais particuliers*. Hermes Science Publications, 2005.
- [CPP09] Marco Conti, Andrea Passarella, and Luciana Pelusi. Mobile-relay forwarding in opportunistic networks. In *Chapter in Adaptive Techniques in Wireless Networks (M. Ibnkahla, Editor)*. Citeseer, 2009.
- [CRL13] João MLP Caldeira, Joel JPC Rodrigues, and Pascal Lorenz. Mac layer handover mechanism for continuous communication support in healthcare mobile wireless sensor networks. *Telecommunication Systems*, pages 1–14, 2013.
- [CS01] Kuo-Hsing Chiang and Nirmala Shenoy. A random walk mobility model for location management in wireless networks. In *Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on*, volume 2, pages E–43. IEEE, 2001.
- [CTDC06] Erdal Cayirci, Hakan Tezcan, Yasar Dogan, and Vedat Coskun. Wireless sensor networks for underwater surveillance systems. *Ad Hoc Networks*, 4(4) :431–446, 2006.

- [CW06] Mihaela Cardei and Jie Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Computer communications*, 29(4) :413–420, 2006.
- [DEG<sup>+</sup>01] Lance Doherty, Laurent El Ghaoui, et al. Convex position estimation in wireless sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1655–1663. IEEE, 2001.
- [DHM04] Jing Deng, Richard Han, and Shivakant Mishra. Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In *Dependable Systems and Networks, 2004 International Conference on*, pages 637–646. IEEE, 2004.
- [DLH<sup>+</sup>14] Wei Dong, Yunhao Liu, Yuan He, Tong Zhu, and Chun Chen. Measurement and analysis on the packet delivery performance in a large-scale sensor network. *Networking, IEEE/ACM Transactions on*, 22(6) :1952–1963, 2014.
- [DPS08] Gianluca Dini, Marco Pelagatti, and Ida Maria Savino. An algorithm for reconnecting wireless sensor network partitions. In *Wireless Sensor Networks*, pages 253–267. Springer, 2008.
- [DQW14] Tao Du, Shouning Qu, and Qin Wang. An energy aware ladder diffusion routing algorithm for wsns. *Lecture Notes on Information Theory Vol*, 2(1), 2014.
- [dRCD<sup>+</sup>11a] Clément Burin des Rozières, Guillaume Chelius, Tony Ducrocq, Eric Fleury, Antoine Fraboulet, Antoine Gallais, Nathalie Mitton, Thomas Noël, Erkan Valentin, and Julien Vandaële. Two demos using senslab : Very large scale open wsn testbed. In *Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on*, pages 1–2. IEEE, 2011.
- [dRCD<sup>+</sup>11b] Clément Burin des Rozières, Guillaume Chelius, Tony Ducrocq, Eric Fleury, Antoine Fraboulet, Antoine Gallais, Nathalie Mitton, Thomas Noël, and Julien Vandaële. Using senslab as a first class scientific tool for large scale wireless sensor network experiments. In *NETWORKING 2011*, pages 147–159. Springer, 2011.
- [DRS<sup>+</sup>05] Karthik Dantu, Mohammad Rahimi, Hardik Shah, Sandeep Babel, Amit Dhariwal, and Gaurav S Sukhatme. Robomote : enabling mobility in sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 55. IEEE Press, 2005.
- [DVMSR10] Tony Ducrocq, Julien Vandaële, Nathalie Mitton, and David Simplot-Ryl. Large scale geolocalization and routing experimentation with the senslab testbed. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on*, pages 751–753. IEEE, 2010.

- 
- [Ega05] David Egan. The emergence of zigbee in building automation and industrial controls. *Computing and Control Engineering*, 16(2) :14–19, 2005.
- [EGHK99] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges : Scalable coordination in sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270. ACM, 1999.
- [EH12] Samina Ehsan and Bechir Hamdaoui. A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks. *Communications Surveys & Tutorials, IEEE*, 14(2) :265–278, 2012.
- [FJFY13] Iztok Fister Jr, Dušan Fister, and Xin-She Yang. A hybrid bat algorithm. *arXiv preprint arXiv :1303.6310*, 2013.
- [FJWW13] Chunyao Fu, Zhifang Jiang, WEI Wei, and Ang WEI. An energy balanced algorithm of leach protocol in wsn. *IJCSI International Journal of Computer Science Issues*, 10(1) :354–359, 2013.
- [FLN03] Ronald Fagin, Amnon Lotem, and Moni Naor. Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66(4) :614–656, 2003.
- [FRL05] Chien-Liang Fok, Gruiia-Catalin Roman, and Chenyang Lu. Rapid development and flexible deployment of adaptive wireless sensor network applications. In *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, pages 653–662. IEEE, 2005.
- [GBE<sup>+</sup>05] Yaoyao Gu, Doruk Bozdog, Eylem Ekici, Füsün Özgüner, and Chang-Gun Lee. Partitioning based mobile element scheduling in wireless sensor networks. In *SECON*, pages 386–395. Citeseer, 2005.
- [GD90] Shahram Ghandeharizadeh and David J DeWitt. Hybrid-range partitioning strategy : A new declustering strategy for multiprocessor database machines. In *Proceedings of the 16th International Conference on Very Large Data Bases*, pages 481–492. Morgan Kaufmann Publishers Inc., 1990.
- [GDPV03] Shashidhar Rao Gandham, Milind Dawande, Ravi Prakash, and Subbarayan Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Global telecommunications conference, 2003. GLOBECOM'03. IEEE*, volume 1, pages 377–381. IEEE, 2003.
- [GVSOTCBA09] Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Alicia Triviño Cabrera, and Claudia Jacy Barenco Abbas. Routing protocols in wireless sensor networks. *Sensors*, 9(11) :8399–8421, 2009.
- [HBT<sup>+</sup>03] Julian Hsu, Sameer Bhatia, Mineo Takai, Rajive Bagrodia, and Michael J Acriche. Performance of mobile ad hoc networking routing protocols in realistic scenarios. In *Military Communications Conference*,

2003. *MILCOM'03. 2003 IEEE*, volume 2, pages 1268–1273. IEEE, 2003.
- [HCB00] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2000.
- [HCB02] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4) :660–670, 2002.
- [HGPC99] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60. ACM, 1999.
- [HHB<sup>+</sup>03] Tian He, Chengdu Huang, Brian M Blum, John A Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 81–95. ACM, 2003.
- [HN15] Mohammad Hammoudeh and Robert Newman. Adaptive routing in wireless sensor networks : Qos optimisation for enhanced application performance. *Information Fusion*, 22 :3–15, 2015.
- [HPS02] Zygmunt J Haas, Marc R Pearlman, and Prince Samar. The zone routing protocol (zrp) for ad hoc networks. *draft-ietf-manet-zone-zrp-04.txt*, 2002.
- [HSA00] G Hoblos, M Staroswiecki, and A Aitouche. Optimal design of fault tolerant sensor networks. In *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*, pages 467–472. IEEE, 2000.
- [HSLA03] Tian He, John A Stankovic, Chenyang Lu, and Tarek Abdelzaher. Speed : A stateless protocol for real-time communication in sensor networks. In *Distributed Computing Systems, 2003. Proceedings. 23rd International Conference on*, pages 46–55. IEEE, 2003.
- [HT05] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4) :519–528, 2005.
- [HXS<sup>+</sup>13] Pei Huang, Li Xiao, Sima Soltani, Matt W Mutka, and Ning Xi. The evolution of mac protocols in wireless sensor networks : A survey. *Communications Surveys & Tutorials, IEEE*, 15(1) :101–120, 2013.
- [Ibn08] Mohamed Ibnkahla. *Adaptation and cross layer design in wireless networks*, volume 2. CRC press, 2008.
- [IGE<sup>+</sup>03] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *Networking, IEEE/ACM Transactions on*, 11(1) :2–16, 2003.

- 
- [IMK11] Farruh Ishmanov, Aamir Saeed Malik, and Sung Won Kim. Energy consumption balancing (ecb) issues and mechanisms in wireless sensor networks (wsns) : a comprehensive overview. *European Transactions on Telecommunications*, 22(4) :151–167, 2011.
- [ISS12] Fabio Iannello, Osvaldo Simeone, and Umberto Spagnolini. Medium access control protocols for wireless sensor networks with energy harvesting. *Communications, IEEE Transactions on*, 60(5) :1381–1389, 2012.
- [JBRAS03] Amit Jardosh, Elizabeth M Belding-Royer, Kevin C Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 217–229. ACM, 2003.
- [JD88] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [Jia99] Mingliang Jiang. Cluster based routing protocol (cbrp). *draft-ietf-manet-cbrp-spec-01.txt*, 1999.
- [JM96] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.
- [JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering : a review. *ACM computing surveys (CSUR)*, 31(3) :264–323, 1999.
- [JSB<sup>+</sup>06] Sushant Jain, Rahul C Shah, Waylon Brunette, Gaetano Borriello, and Sumit Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *Mobile Networks and Applications*, 11(3) :327–339, 2006.
- [Kac47] Mark Kac. Random walk and the theory of brownian motion. *American Mathematical Monthly*, pages 369–391, 1947.
- [KEW02] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker. The impact of data aggregation in wireless sensor networks. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 575–578. IEEE, 2002.
- [KPSV02] Farinaz Koushanfar, Miodrag Potkonjak, and A Sangiovanni-Vincentelli. Fault tolerance techniques for wireless ad hoc sensor networks. In *Sensors, 2002. Proceedings of IEEE*, volume 2, pages 1491–1496. IEEE, 2002.
- [KSJ<sup>+</sup>04] Aman Kansal, Arun A Somasundara, David D Jea, Mani B Srivastava, and Deborah Estrin. Intelligent fluid infrastructure for embedded networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 111–124. ACM, 2004.
- [Leh09] Mohamed Lehsaini. *Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique*. PhD thesis, 2009.

- [LFBG15] Lemia Louail, Violeta Felea, Julien Bernard, and Herve Guyennet. Mac-aware routing in wireless sensor networks. In *Communications and Networking (BlackSeaCom), 2015 IEEE International Black Sea Conference on*, pages 225–229. IEEE, 2015.
- [LGF07] Hung-Cuong Le, Herve Guyennet, and Violeta Felea. Obmac : an overhearing based mac protocol for wireless sensor networks. In *Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on*, pages 547–553. IEEE, 2007.
- [LGF08] Mohamed Lehsaini, Hervé Guyennet, and Mohammed Feham. A novel cluster-based self-organization algorithm for wireless sensor networks. In *Collaborative Technologies and Systems, 2008. CTS 2008. International Symposium on*, pages 19–26. IEEE, 2008.
- [LGF10] Mohamed Lehsaini, Herve Guyennet, and Mohammed Feham. An efficient cluster-based self-organisation algorithm for wireless sensor networks. *International Journal of Sensor Networks*, 7(1) :85–94, 2010.
- [LH05] Jun Luo and J-P Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies. Proceedings IEEE*, volume 3, pages 1735–1746. IEEE, 2005.
- [LLV13] Mo Li, Zhenjiang Li, and Athanasios V Vasilakos. A survey on topology control in wireless sensor networks : Taxonomy, comparative study, and open issues. *Proceedings of the IEEE*, 101(12) :2538–2557, 2013.
- [LM07] Jian Li and Prasant Mohapatra. Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive and Mobile Computing*, 3(3) :233–254, 2007.
- [LMFJ<sup>+</sup>04] Konrad Lorincz, David J Malan, Thaddeus RF Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoffrey Mainland, Matt Welsh, and Steve Moulton. Sensor networks for emergency response : challenges and opportunities. *Pervasive Computing, IEEE*, 3(4) :16–23, 2004.
- [LMP<sup>+</sup>05] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, et al. Tinyos : An operating system for sensor networks. In *Ambient intelligence*, pages 115–148. Springer, 2005.
- [LR02] Stephanie Lindsey and Cauligi S Raghavendra. Pegasus : Power-efficient gathering in sensor information systems. In *Aerospace conference proceedings, 2002. IEEE*, volume 3, pages 3–1125. IEEE, 2002.
- [LTGLH07] Paul Lukowicz, Andreas Timm-Giel, Michael Lawo, and Otthein Herzog. Wearit@ work : Toward real-world industrial wearable computing. *Pervasive Computing, IEEE*, 6(4) :8–13, 2007.
- [LW07] Fan Li and Yu Wang. Routing in vehicular ad hoc networks : A survey. *Vehicular Technology Magazine, IEEE*, 2(2) :12–22, 2007.

- 
- [MA01] Arati Manjeshwar and Dharma P Agrawal. Teen : A routing protocol for enhanced efficiency in wireless sensor networks. In *IPDPS*, volume 1, page 189, 2001.
- [MA02] Arati Manjeshwar and Dharma P Agrawal. Apteen : A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *ipdps*, volume 2, page 48, 2002.
- [MB02] Archan Misra and Suman Banerjee. Mrpc : Maximizing network lifetime for reliable routing in wireless environments. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 2, pages 800–806. IEEE, 2002.
- [MCP<sup>+</sup>02] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97. ACM, 2002.
- [MG10] David Martins and Hervé Guyennet. Wireless sensor network attacks and security mechanisms : A short survey. In *Network-Based Information Systems (NBIS), 2010 13th International Conference on*, pages 313–320. IEEE, 2010.
- [MK07] Naourez Mejri and Farouk Kamoun. Algorithme de routage hiérarchique mheed à plusieurs sauts pour les grands réseaux de capteurs. In *4th International Conference : Sciences of Electronic, Technologies of Information and Telecommunications*, pages 1–7, 2007.
- [MM08] Saoucene Mahfoudh and Pascale Minet. An energy efficient routing based on olsr in wireless ad hoc and sensor networks. In *Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1253–1259. IEEE, 2008.
- [MRS14] Dieter Mitsche, Giovanni Resta, and Paolo Santi. The random waypoint mobility model with uniform node spatial distribution. *Wireless networks*, 20(5) :1053–1066, 2014.
- [PB94] Charles E Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *ACM SIGCOMM Computer Communication Review*, volume 24, pages 234–244. ACM, 1994.
- [PBDT03] Nissanka B Priyantha, Hari Balakrishnan, Erik Demaine, and Seth Teller. Anchor-free distributed localization in sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 340–341. ACM, 2003.
- [PGC00] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing : A routing scheme for ad hoc wireless networks. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 1, pages 70–74. IEEE, 2000.
- [PK00] Gregory J Pottie and William J Kaiser. Wireless integrated network sensors. *Communications of the ACM*, 43(5) :51–58, 2000.



- [RM04] Kay Romer and Friedemann Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE*, 11(6) :54–61, 2004.
- [RSPS02] Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B Srivastava. Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2) :40–50, 2002.
- [RT99] Elizabeth M Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *Personal Communications, IEEE*, 6(2) :46–55, 1999.
- [SJ04] Amit Kumar Saha and David B Johnson. Modeling mobility for vehicular ad-hoc networks. In *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 91–92. ACM, 2004.
- [SKJ<sup>+</sup>06] Arun A Somasundara, Aman Kansal, David D Jea, Deborah Estrin, and Mani B Srivastava. Controllably mobile infrastructure for low energy embedded networks. *Mobile Computing, IEEE Transactions on*, 5(8) :958–973, 2006.
- [SR02] Rahul C Shah and Jan M Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Wireless Communications and Networking Conference, 2002. WCNC2002. 2002 IEEE*, volume 1, pages 350–355. IEEE, 2002.
- [SRS04] Arun A Somasundara, Aditya Ramamoorthy, and Mani B Srivastava. Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines. In *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, pages 296–305. IEEE, 2004.
- [SS12] R Sumathi and MG Srinivas. A survey of qos based routing protocols for wireless sensor networks. *Journal of Information Processing Systems*, 8(4) :589–602, 2012.
- [SS14] Getsy S Sara and D Sridharan. Routing in mobile wireless sensor network : a survey. *Telecommunication Systems*, 57(1) :51–79, 2014.
- [SSB14] Ricardo Silva, Jorge Sá Silva, and Fernando Boavida. Mobility in wireless sensor networks survey and proposal. *Computer Communications*, 52 :1–20, 2014.
- [SSSM64] Frank Spitzer, Frank Spitzer, Frank Spitzer, and Austria Mathematician. *Principles of random walk*. Springer, 1964.
- [Toh01] Chai-Keong Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *Communications Magazine, IEEE*, 39(6) :138–147, 2001.
- [TR04] Caimu Tang and Cauligi S Raghavendra. Compression techniques for wireless sensor networks. In *Wireless sensor networks*, pages 207–231. Springer, 2004.
- [WB10] Qinghua Wang and Ilangko Balasingham. Wireless sensor networks-an introduction. *Chapter*, 1 :1–13, 2010.

- 
- [WBMP05] Z Maria Wang, Stefano Basagni, Emanuel Melachrinoudis, and Chiara Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*, pages 287a–287a. IEEE, 2005.
- [WGG10] Klaus Wehrle, Mesut Güneş, and James Gross. *Modeling and tools for network simulation*. Springer, 2010.
- [WQD<sup>+</sup>14] Na Wang, Xiao-gang Qi, Li Duan, Hua Jiang, and Xiaofei Liu. Clustering-based routing algorithm in wireless sensor networks with mobile sink. *Journal of Networks*, 9(9) :2376–2383, 2014.
- [WWT12] You-Chiun Wang, Fang-Jing Wu, and Yu-Chee Tseng. Mobility management algorithms and applications for mobile sensor networks. *Wireless Communications and Mobile Computing*, 12(1) :7–21, 2012.
- [WXZ<sup>+</sup>03] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 28–39. ACM, 2003.
- [XHE01] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2001.
- [XLL<sup>+</sup>10] Jiuqiang Xu, Wei Liu, Fenggao Lang, Yuanyuan Zhang, and Chenglong Wang. Distance measurement model based on rssi in wsn. *Wireless Sensor Network*, 2(08) :606, 2010.
- [XW<sup>+</sup>05] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3) :645–678, 2005.
- [Yan10] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer, 2010.
- [YF04] Ossama Younis and Sonia Fahmy. Heed : a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *Mobile Computing, IEEE Transactions on*, 3(4) :366–379, 2004.
- [YGE01] Yan Yu, Ramesh Govindan, and Deborah Estrin. Geographical and energy aware routing : A recursive data dissemination protocol for wireless sensor networks. Technical report, Technical report ucla/csd-tr-01-0023, UCLA Computer Science Department, 2001.
- [YLCW05] Mao Ye, Chengfa Li, Guihai Chen, and Jie Wu. Eecs : an energy efficient clustering scheme in wireless sensor networks. In *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International*, pages 535–540. IEEE, 2005.

- [YLN03] Jungkeun Yoon, Mingyan Liu, and Brian Noble. Random waypoint considered harmful. In *INFOCOM 2003. twenty-second annual joint conference of the IEEE computer and communications. IEEE societies*, volume 2, pages 1312–1321. IEEE, 2003.
- [YMG08] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer networks*, 52(12) :2292–2330, 2008.
- [YZLM14] Sheng Yu, Baoxian Zhang, Cheng Li, and Hussein Mouftah. Routing protocols for wireless sensor networks with mobile sinks : A survey. *Communications Magazine, IEEE*, 52(7) :150–157, 2014.
- [ZK03] Marco Zuniga and Bhaskar Krishnamachari. Integrating future large-scale wireless sensor networks with the internet. *USC Computer Science, Tech. Rep*, 2003.
- [ZLGZ11] Hai-Ying Zhou, Dan-Yan Luo, Yan Gao, and De-Cheng Zuo. Modeling of node energy consumption for wireless sensor networks. *Wireless Sensor Network*, 3(01) :18, 2011.

# Résumé

Un certain nombre de travaux basés sur les réseaux de capteurs sans fil s'intéressent à la gestion de l'énergie de ces capteurs. Cette énergie est, de fait, un facteur critique dans le fonctionnement de ces réseaux. Une construction adéquate des clusters de capteurs est un très bon moyen pour minimiser la consommation de cette énergie. La problématique liée à ces réseaux réside ainsi souvent dans leur durée de vie mais aussi dans le nécessaire maintien de la connectivité entre tous les capteurs. Ces deux aspects sont étroitement liés. Dans cette thèse, nous nous sommes focalisés sur ces deux volets, dans le contexte de réseaux de capteurs statiques mais aussi celui de capteurs mobiles.

Nous proposons, dans un premier temps, un algorithme hybride pour la mise en place des clusters et la gestions de ces clusters. L'originalité de cette solution réside dans la mise en place de zones géographiques de désignation des cluster heads mais aussi dans la transmission, dans les messages échangés, de la quantité d'énergie restante sur les capteurs. Ainsi, les données sur les capteurs permettront de désigner les cluster heads et leurs successeurs qui détermineront les seuils pour les autres capteurs et pour leur fonctionnement. L'algorithme est testé à travers de nombreuses simulations.

La seconde partie du travail consiste à adapter notre premier algorithme pour les réseaux de capteurs mobiles. Nous influons sur la trajectoire des capteurs pour maintenir la connectivité et limiter la consommation d'énergie. Pour cela, nous nous inspirons de l'écho-localisation pratiquée par les chauves-souris. Nous nous sommes donc intéressés à la topologie changeante et dynamique dans les réseaux de capteurs. Nous avons analysé la perte d'énergie en fonction de la distance et de la puissance de transmission entre les nœuds et le cluster head. Nous évaluons également notre algorithme sur des capteurs qui ont un déplacement aléatoire. Nous appliquons ces algorithmes à une simulation de flotte de drones de surveillance.

**Mots-clés:** WSN, Routage, Clustering, Algorithme, Mobilité, Système multi-agents, Système complexe.

# Abstract

A number of works based on wireless sensor networks are interested in the energy management of these sensors. This energy is in fact a critical factor in the operation of these networks. Proper construction of sensor clusters is a great way to minimize the consumption of this energy. The problems related to these networks and often lies in their lifetime but also in the need to maintain connectivity between all transducers. These two aspects are closely linked. In this thesis, we focused on these two aspects in the context of static sensor networks but also of mobile sensors.

We propose, as a first step, a hybrid algorithm for setting up clusters and the management of these clusters. The uniqueness of this solution lies in the establishment of geographic areas for designation of cluster heads but also in transmission, in the exchanged messages, the amount of remaining energy on the sensors. Thus, the sensor data will designate the cluster heads and their successors will determine the thresholds for other sensors and for their operation. The algorithm is tested through many simulations.

The second part of the work is to adapt our first algorithm for mobile sensor networks. We affect the trajectory of sensors to maintain connectivity and reduce energy consumption. For this, we are guided echo-location practiced by bats. We're interested in changing and dynamic topology in sensor networks. We analyzed the loss of energy as a function of the distance and the power transmission between the nodes and the head cluster. We also evaluate our algorithm on sensors that have a random move. We apply these algorithms to a fleet of surveillance drones simulation.

**Keywords:** WSN, Routing, Clustering, Algorithmic, Mobility, Multi-agent system, complex system.



