



MEMOIRE

Pour l'obtention de Diplôme de Master en informatique
Option : Réseaux mobilité et systèmes embarqués

Intitulé :

Routage hiérarchique dans les RCSF

(Comparaison entre variantes de LEACH)

Présenté par : *LAOUARI Rahma et BESLINE Nassima*

Dirigé par : M^{er} TALBI SAID

Promotion 2012/2013

Remerciements

Nous tenons à remercier en premier lieu le Bon Dieu tout Puissant qui nous a accordé la volonté et le courage pour la réalisation de ce projet.

Nous tenons à exprimer nos vifs remerciements aussi à tous ceux qui nous ont aidés à aboutir dans notre travail et à tous ceux qui ont contribué d'une façon ou d'une autre à réaliser ce projet, en premier lieu notre promoteur

Mer TALBI Saïd

qui nous a beaucoup aidés avec ses conseils, sa disponibilité, sa gentillesse et sa sympathie,

Nos plus vifs remerciements vont également aux membres du jury pour nous avoir fait l'honneur de juger ce travail.



Dédicace Rahma



Je tiens très respectueusement à dédier ce modeste travail à mon très cher mari Smail pour son soutien, son encouragement et sa compréhension.

A la plus belle créature sur terre mon petit bébé Imad pour tout le bon dérangement qu'il m'a offert au cours de ce travail.

A mes parents et beaux parents, à mes frères et sœurs, à ma belle sœur Saliha et toute sa famille.

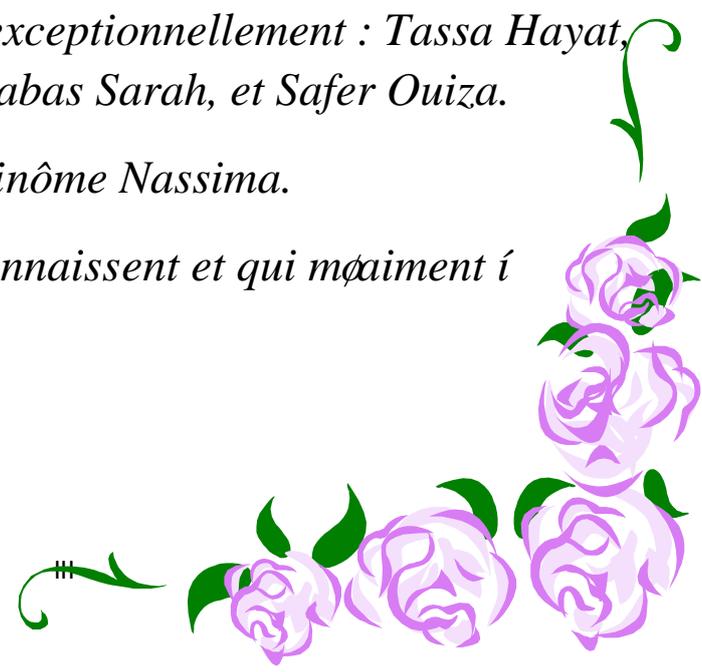
Je le dédie également à toute ma famille, du plus vieux jusqu'au plus jeune

A tous mes enseignants de l'école primaire jusqu'à l'université.

A tous mes amis et je cite exceptionnellement : Tassa Hayat, kebaily Farisa, Oubabas Sarah, et Safer Ouiza.

A mon binôme Nassima.

A tous ceux qui me connaissent et qui m'aiment





Dédicace Nassima

*Je tiens très respectueusement à dédier ce modeste travail
A mes très chers parents qui ont toujours été là pour moi,
Et qui m'ont donné un magnifique modèle de labeur et de
Persévérance.*

*A mon cher mari Sofiane pour son soutien, encouragement et sa
Compréhension*

*A mes sœurs Cilia et Anais et mes frères Rafik, Anis et surtout
mon Adorable Idir.*

*A mes grands parents maternels et paternels, à mes tantes et
Oncles.*

A ma belle famille des plus vieux aux plus jeunes.

*A tous mes cousins et cousines et à tous mes amis de près
Comme de loin*

Surtout à ma binôme Rahma et tous mes enseignants

A tous ceux qui me connaissent et m'aiment ..



Résumé :

Les capteurs sont des composants électroniques de petites tailles alimentés par batterie. Ils sont déployés en grand nombre dans des environnements généralement inaccessibles à l'homme afin de récolter des informations liées à ces environnements. Les données perçues par un capteur sont acheminées de voisin en voisin jusqu'à atteindre une station de base ou puits. C'est la raison pour laquelle nous parlons des réseaux de capteurs sans fil. Le cheminement des données à travers les différents nœuds (capteurs) du réseau obéit à un protocole de routage. La qualité de ce protocole a une incidence directe sur la durée de vie de la batterie du capteur. Nous décrivons dans ce mémoire le protocole de routage LEACH et ses variantes DSLEACH et CVLEACH, on va les implémenter sous OMNeT++ pour tirer les résultats concernant la consommation énergétique et la durée de vie de réseaux.

Mots clé : réseaux de capteur sans fil, protocoles de routage LEACH, DSLEACH et CVLEACH, Conservation d'énergie, Techniques de clustering.

Sommaire

Liste des figures	IX
Introduction générale :	XIV
1. Généralités sur les réseaux de capteur sans fil.....	1
1.1 Introduction.....	2
1.2 L'environnement sans fil	2
1.2.1 Définition	2
1.2.2 Catégories.....	2
1.2.3 Classification selon l'infrastructure.....	3
1.3 Réseaux de capteurs	4
1.3.1 Définition d'un capteur.....	4
1.3.2 Anatomie d'un capteur.....	5
1.3.3 Définition d'un réseau de capteur	6
1.3.4 Caractéristiques des réseaux de capteurs.....	7
1.3.5 Architecture d'un réseau de capteurs.....	8
1.3.6 Architecture protocolaire des réseaux de capteurs	9
1.3.7 Domaines d'applications des RCSF	11
1.4 Différentes problématiques dans les réseaux de capteurs.....	12
1.5 Conclusion	13
2. Routage dans les RCSF	15
2.1 Introduction.....	16
2.2 Facteurs de conception des RCSF.....	16
2.3 Classification des protocoles de routage.....	18
2.3.1 Protocoles basés sur la structure du réseau	18
2.3.2 Les protocoles de routage en fonction de leurs modes opératoires	20
2.3.3 Routage en fonction d'établissement des chemins.....	21
2.3.4 Routage en fonction de paradigme de communication.....	22
2.4 Exemples de protocoles de routage dans les WSN	23
2.4.1 Protocoles plats.....	23
2.4.2 Protocoles hiérarchiques.....	29
2.5 Conclusion	35
3. Protocoles LEACH, et ses variantes.....	36

3.1. Introduction.....	37
3.2 Le protocole de base LEACH :(<i>Low Energy Adaptive Clustering Hierarchy</i>).....	38
3.2.1 Phase d'initialisation	39
3.2.2 Phase de transmission	42
3.2.3 Avantages et limites.....	43
3.3. Le protocole DSLEACH (<i>Density of Sensor based LEACH</i>).....	43
3.3.1 Principe de fonctionnement.....	44
3.3.1 Avantages et limites.....	46
3.4 Le protocole CVLEACH (<i>Coverage based energy efficient LEACH</i>)	46
3.4.1 Le principe du fonctionnement.....	46
3.4.2 Les principales améliorations.....	47
3.4.3 Avantages et limites.....	47
3.5 Conclusion	47
4. Tests et comparaison	48
4.1 Introduction.....	49
4.2 Présentation de quelques simulateurs existants.....	49
4.2.1 NS 2	49
4.2.2 J-Sim	50
4.2.3 TinyOs.....	50
4.2.4 OMNeT++.....	50
4.3 Le choix de notre simulateur	51
4.3.1 Les principaux éléments d'OMNeT	51
4.3.1. Principe de fonctionnement d'OMNeT++	55
4.3.2 L'architecture d'OMNet.....	55
4.4 Les différents modèles implémentés	56
4.4.1 Le module simple « Physical Process ».....	56
4.4.2 Le module simple « Wireless Chanel »	57
4.4.3. Le module composé « Node » :.....	58
4.4.4 Le modèle énergétique.....	62
4.4.6 Implémentation des différentes fonctionnalités.....	63
4.5 Simulation	63
4.5.1 Les métriques d'évaluation des performances.....	63



PDF Complete

Your complimentary use period has ended. Thank you for using PDF Complete.

[Click Here to upgrade to Unlimited Pages and Expanded Features](#)

4.5.2 Paramètres de simulation.....	65
4.6 Discussion des résultats	66
4.6.1 Moyenne d'énergie consommée (MEC).....	66
4.6.2 Ecart type d'énergie consommée	67
4.6.3 Durée de vie du réseau.....	67
4.7 Conclusion	69
Conclusion générale :.....	70
Bibliographie.....	72

Liste des figures

Figure 1.1 : Réseau avec infrastructure

Figure 1.2: Réseau sans infrastructure.

Figure 1.3: Quelques capteurs.

Figure 1.4: Anatomie d'un nœud sense.

Figure 1.5: exemple d'un RCSF.

Figure 1.6: Architecture d'un RCSF.

Figure 1.7: pile protocolaire des RCSF

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

Figure 2.2: La topologie plate.

Figure 2.3: La topologie hiérarchique.

Figure 2.4: fonctionnement de protocole SPIN

Figure 2.5: les phases de communication de protocole DD

Figure 2.6: Agent d'événement

Figure 3.1: routage basé clustering

Figure 3.2 : un réseau avec clustering

Figure 3.3 : opération de l'étape d'initialisation de LEACH

Figure 3.4 : Interférence lors d'une communication dans LEACH

Figure 3.5 : Répartition du temps et différentes phases pour chaque round

Figure 4.1 : Les éléments principaux d'OMNeT IDE

Figure 4.2: Le mode graphique de fichier NED

Figure 4.3: Le mode texte de fichier NED

Figure 4.4: L'éditeur de fichier INI

Figure 4.5 : L'éditeur de code source INI

Figure 4.6: L'architecture en modules d'OMNeT

Figure 4.7: La structure de notre réseau

Figure 4.8: L'arborescence de module PhysicalProcess

Figure 4.9: L'arborescence de module WirelessChannel

Figure 4.10 : L'arborescence de module Node

Figure 4.11: Architecture interne d'un nœud capteur

Figure 4.12: L'architecture de notre plateforme (les modules et leurs connexions)

Figure 4.13 : Moyenne d'énergie consommée par les nœuds

Figure 4.14 : l'écart type de l'énergie consommée

Figure 4.15 : La durée de vie de réseau pour 50 nœuds

Figure 4.16: La durée de vie de réseau pour 100 nœuds.

Introduction générale :

Les progrès technologiques réalisés ces dernières années dans le domaine des réseaux sans fil, de la micro-fabrication et des microprocesseurs embarqués ont permis la production de nouveaux types de capteurs dotés de moyens de communication sans fil, peu onéreux et pouvant être configurés pour former des réseaux autonome et à infrastructures non prédéfinies. Ensemble, ils forment un réseau de capteurs sans fil capable de surveiller une région ou un phénomène d'intérêt, de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs, de les traiter pour les communiquer ensuite via le support sans fil à un ou plusieurs points de collecte.

De nombreux domaines d'application sont alors envisagés pour ces réseaux, tels que le contrôle de l'environnement, la détection et la surveillance des désastres, le domaine militaire et médical, et bien d'autres.

Cependant, du à leurs nature embarquée, autonome et miniature, les capteurs présentent certaines contraintes. Par exemple, la faible satisfaction en énergie est le point le plus crucial dans les RCSF. La batterie est un composant important d'un capteur. En général, elle n'est ni remplaçable ni rechargeable. Elle fournit donc une quantité d'énergie très limitée à l'échelle de 1 à 2 Joule par node (capteur). Elle limite ainsi la durée de vie du capteur et influe sur le fonctionnement global du réseau.

L'optimisation de la consommation au niveau d'un réseau est un problème complexe. Il nécessite une solution globale par la mise en œuvre d'algorithmes de communication et de routage maximisant la durée de vie du réseau.

Plusieurs protocoles ont été développés ou adaptés afin de diminuer la dissipation globale d'énergie et augmenter en conséquence la durée de vie du réseau.

LEACH est le premier et le plus populaire des algorithmes de routage hiérarchique avec une efficacité énergétique, ce dernier à plusieurs variantes nous avons à titre d'exemple (LEACH-C, PEGASSIS, DS-LEACH, CV-LEACH, T-LEACH).

Dans ce mémoire nous allons effectués une comparaison entre deux variantes très ressenties de LEACH qui son DSLEACH et CVLEACH. Ces deux protocole se base sur LEACH et essaye de

remédier ou problème de la couverture de réseau en utilisant différentes techniques, mais on ne sait pas lequel d'entre eux est le plus efficace.

Notre travail consiste alors à étudier ces protocoles les implémenter et les comparer à l'aide de simulation pour sélectionner le meilleur d'entre eux (en terme de consommation énergétique).

Ce manuscrit s'articule autour de quatre chapitres :

Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil, leurs caractéristiques, leurs architectures, et leurs domaines d'applications.

Dans le second chapitre, on s'intéresse au routage dans les RCSF, on va commencer par les facteurs de conception de ces réseaux, pour classer par la suite les protocoles de routage selon différents critères, et nous allons terminer par la citation de quelques exemples de protocoles existant.

Le troisième chapitre est consacré à la description détaillée des protocoles de routage LEACH, DSLEACH et CVLEACH.

Dans le quatrième chapitre nous allons détailler notre plateforme de simulation et les étapes d'implémentation de nos protocoles, pour passer à la comparaison des résultats de simulation.

Enfin, nous terminons par une conclusion générale qui résume nos conclusions tirées de l'étude de l'état de l'art et de la discussion des résultats obtenus par notre étude comparative, et présenter les perspectives envisageables comme amélioration de notre travail.

1. Généralités sur les réseaux de capteur sans fil

1.1 Introduction

L'observation et le contrôle des phénomènes physiques sont importants dans de nombreuses applications industrielles et scientifiques. Ces observations sont parfois effectuées dans des environnements hostiles et inaccessibles. Les récentes avancées technologiques dans le domaine des communications sans fil, la disponibilité de capteurs intelligents à faible coût munis de puissantes capacités de calcul et de perception ont contribué au développement des réseaux de capteurs sans fil. Ce type de réseau est amené à résoudre divers problèmes. Aujourd'hui on peut facilement envisager un large éventail d'applications des réseaux de capteurs sans fil: la surveillance environnementale, les maisons intelligentes, la sécurité, la santé, etc. Les réseaux de capteurs auront, dans le futur, un impact significatif dans notre vie quotidienne. Dans ce chapitre, nous présenterons en premier lieu l'environnement sans fil en général. Nous entamerons par la suite les RCSF en présentant leurs caractéristiques, leurs architectures de communication et architecture protocolaire après avoir vu la définition d'un capteur et son anatomie puis aborderons les facteurs de conception des RCSF. Pour terminer, on décrit les domaines d'application des RCSF.

1.2 L'environnement sans fil

1.2.1 Définition

Un réseau sans fil (ou non filaire) est un réseau informatique qui connecte les différents équipements entre eux par ondes radio, c'est, comme son nom l'indique, un réseau dans lequel les terminaux peuvent communiquer sans liaison filaire. Les terminaux du réseau se déplacent librement, tandis que le système doit assurer toutes les fonctionnalités et tous les services d'un réseau classique. La notion de nœuds se restreint sur les extrémités d'une connexion. Elle peut contenir différents terminaux tel que les routeurs, les ordinateurs, les concentrateurs, les commutateurs, etc.

1.2.2 Catégories

La classification des réseaux sans fil peut s'effectuer selon le périmètre géographique ou l'infrastructure du réseau. Selon le périmètre géographique dans lequel il se situe, un réseau sans fil appartient à l'une des quatre catégories suivantes : réseaux personnels sans fil (WPAN), réseaux métropolitains sans fil (WLAN) réseaux locaux sans fil (WMAN) et réseaux étendus sans fil (WWAN). Selon l'infrastructure, les réseaux sans fil peuvent être classifiés en réseaux sans infrastructure ou avec infrastructure.

1.2.3 Classification selon l'infrastructure

Les réseaux sans fil peuvent être classifiés en deux catégories : les réseaux avec une infrastructure et les réseaux sans infrastructure.

➤ Les réseaux avec infrastructure

Ce type de réseaux demande une infrastructure logistique et matérielle fixe. Cette infrastructure est représentée par un ou plusieurs points d'accès appelé(s) stations de base auxquels sont connectées les unités mobiles. Ce type de réseau utilise un modèle de communication cellulaire (exemple les réseaux : GSM). Toutes les communications sont gérées par un ou plusieurs points d'accès (une station de base). Toutes les données qu'un hôte (unité mobile ou utilisateur mobile) peut émettre sont transmises au point d'accès (la station de base). Seul le point d'accès renvoie les données aux autres membres du réseau. Plusieurs points d'accès peuvent être reliés ensemble (par câble ou par relais wifi). En d'autres termes, une unité mobile du réseau ne peut pas communiquer directement avec une autre unité mobile, donc une gestion centralisée des unités mobiles est nécessaire. L'unité mobile doit tout d'abord passer par le point d'accès de sa cellule.

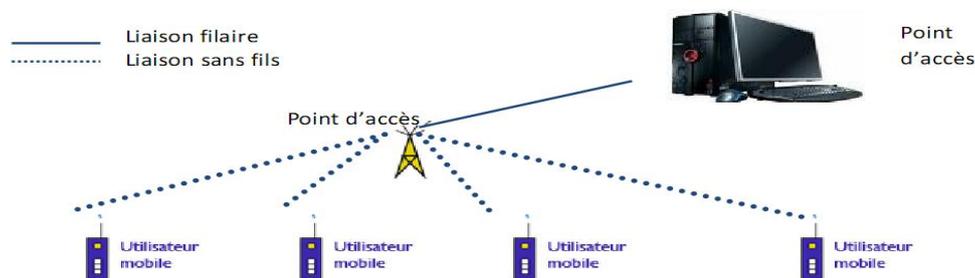


Figure 1.1 : Réseau avec infrastructure

➤ Les réseaux sans infrastructure

Dans ce type de réseaux (appelé aussi réseau ad-hoc) il n'y a pas de gestion centralisée par un point d'accès (ou une station de base). Chaque membre du réseau retransmet les informations qu'il reçoit aux autres membres du réseau constituant ainsi un réseau point à point. En d'autres termes, c'est un réseau dans lequel chaque unité joue en même temps le rôle de client (hôte) et de point d'accès. Il n'y a pas d'infrastructure préexistante, c'est un ensemble d'unités interconnectées à travers une interface sans fil sans administration ni support fixe.

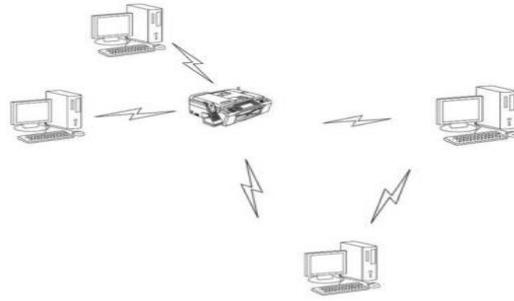


Figure 1.2: Réseau sans infrastructure

Les réseaux ad-hoc sont des réseaux sans fil capables de s'organiser sans infrastructure définie au préalable. Les réseaux ad-hoc, dans leur configuration mobile, sont connus sous le nom de MANETs (Mobile Ad-hoc NETWORKS). Dans un MANET chaque nœud communique directement avec son nœud voisin. Pour communiquer avec des nœuds plus éloignés, il lui est nécessaire de collaborer avec d'autres nœuds qui se chargeront de l'acheminement des données vers le nœud destination. Chaque nœud se comporte comme un routeur. Chaque nœud assure les tâches qui sont réalisées par l'administration centrale fixe dans les réseaux avec infrastructure. En effet, chaque nœud prend ses décisions en fonction de la situation actuelle du réseau, car le réseau est formé dynamiquement et aucune pré-installation relative aux rôles de chaque nœud n'est requise au préalable. A un instant donné en fonction de la position des nœuds, de la configuration de leur émetteur récepteur, des niveaux de puissances de transmission et des interférences entre les canaux il y a une connectivité sans fil qui existe entre les nœuds.

1.3 Réseaux de capteurs

Les RCSFs (Réseaux de Capteurs Sans Fil) sont des types particuliers de réseaux ad-hoc. Ils sont composés de dispositifs micro-électroniques appelés : capteurs.

1.3.1 Définition d'un capteur

Un capteur est un mini-composant, ayant une capacité de traitement embarquée et une mémoire, il permet d'acquérir des données sur son environnement, les traiter et les communiquer. Un nœud dans un réseau de capteurs peut être un dispositif de mesure, un dispositif d'interaction ou les deux en même temps [BR 08]. Un capteur a une capacité de

traitement embarquée et une mémoire. Un nœud peut avoir un ou plusieurs dispositifs de captures : Dispositif de capture acoustique, sismique, infrarouge, magnétique, etc. Et selon l'application, donne ou plusieurs unités optionnelles. Il peut mesurer 5x2x1 cm ou même 1x1x1 cm

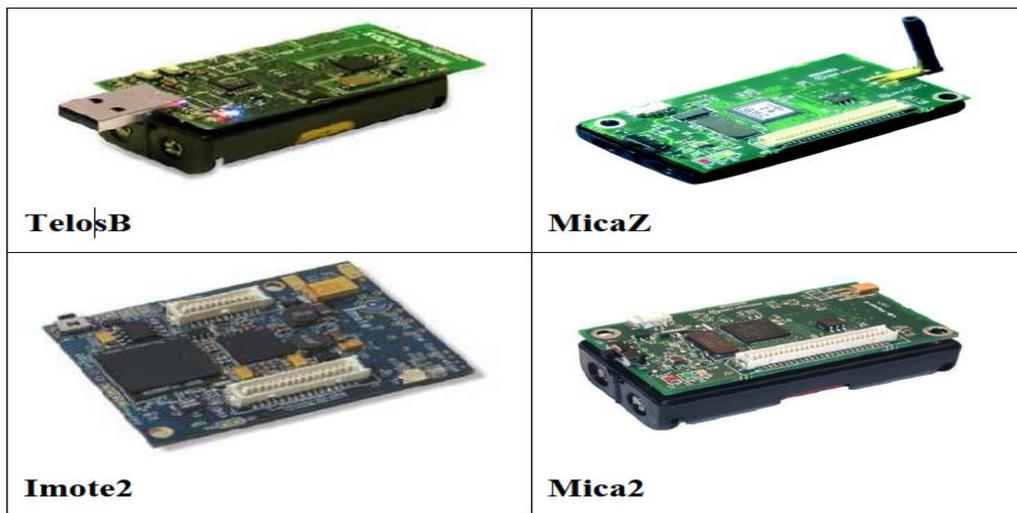


Figure 1.3: Quelques capteurs

1.3.2 Anatomie d'un capteur

Un nœud capteur (dit "mote" en anglais) est composé principalement d'un processeur, une mémoire, un émetteur/récepteur radio [IA 02], un ensemble de capteurs, et une pile (voir figure suivante).

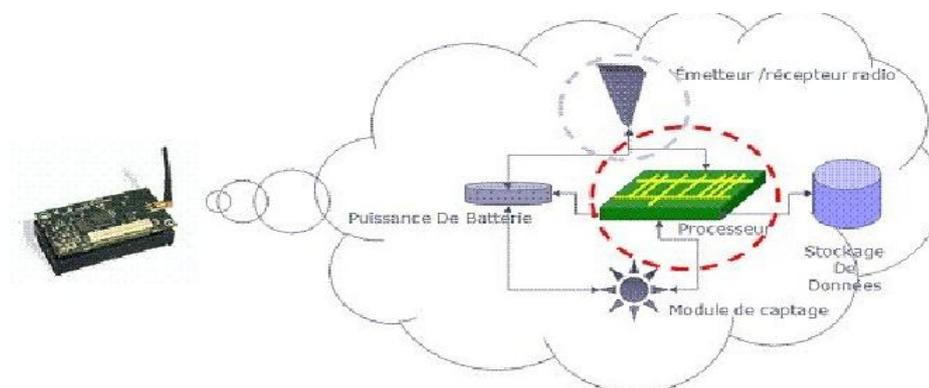


Figure 1.4: Anatomie d'un nœud senseur

Module de captage: Composée d'un capteur qui obtient des mesures sur les paramètres environnementaux et d'un convertisseur Analogique/Numérique qui convertit l'information relevée et la transmet à l'unité de traitement [AM 10].

L'unité de traitement : Composée d'un processeur et d'une mémoire intégrant un système d'exploitation spécifique (TinyOS par exemple). Cette unité possède deux interfaces, une interface pour l'unité d'acquisition et une interface pour l'unité de communication. Elle acquiert les informations en provenance de l'unité d'acquisition et les envoie à l'unité de communication. Cette unité est chargée aussi d'exécuter les protocoles de communications qui permettent de faire collaborer le capteur avec d'autres capteurs. Elle peut aussi analyser les données captées.

L'unité de communication : unité responsable de toutes les émissions et réceptions de données via un support de communication radio. Elle peut être de type optique, ou de type radiofréquence.

La batterie : Un capteur est muni d'une batterie pour alimenter tous ses composants. Cependant, à cause de sa taille réduite, la batterie dont il dispose est limitée et généralement irremplaçable. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influence directement sur la durée de vie des capteurs. Il existe des capteurs qui sont dotés d'autres composants additionnels comme le système de positionnement GPS (Global Positioning System) et un mobilisateur lui permettant le déplacement.

1.3.3 Définition d'un réseau de capteur

Un réseau de capteurs sans fil (RCSF ou WSN : Wireless Sensor Network) est défini dans [S 05] un ensemble de nœuds capteurs dont le nombre varie de quelques dizaines d'éléments à plusieurs milliers, voire des millions. Les capteurs ne sont pas intégrés à une quelconque architecture de communication préexistante, mais ils communiquent à l'aide d'un réseau Ad Hoc sans fil. L'alimentation électrique de chaque capteur est assurée par une batterie individuelle dont la consommation pour la communication et le calcul lié au traitement de l'information doit être optimisée. Le déploiement des nœuds est soit aléatoire (avion, missile) ou déterministe (manuelle, robots). Le déploiement aléatoire rend difficile la détermination des nœuds voisins. Par contre, le déploiement déterministe deviendra difficile si

le nombre de nœuds est très élevé. Le déploiement des nœuds est aussi dense mais il serait impossible pour plusieurs utilisateurs si la mise en place du réseau en entier se révélait très coûteuse.

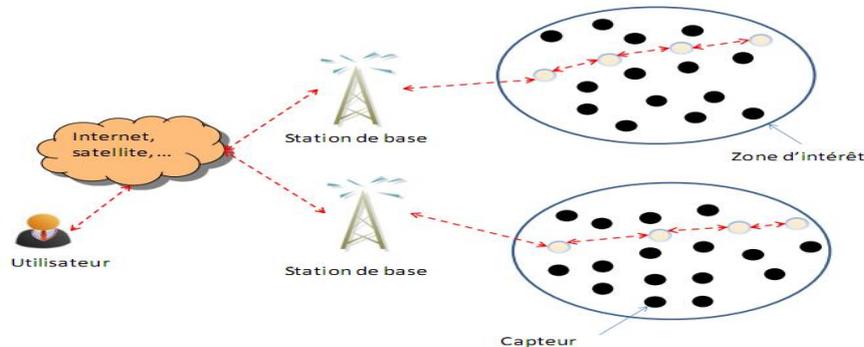


Figure 1.5: exemple d'un RCSF

1.3.4 Caractéristiques des réseaux de capteurs

Énergie limitée: Les RCSF visent la consommation d'énergie puisque l'alimentation de chaque nœud est assurée par une source d'énergie limitée et généralement irremplaçable à cause de l'environnement hostile où il est déployé. De ce fait, la durée de vie d'un RCSF dépend fortement de la conservation d'énergie au niveau de chaque nœud.

É Modèle de communication: Les nœuds dans les RCSF communiquent selon un paradigme plusieurs-à-un (many to one). En effet, les nœuds capteurs collectent des informations à partir de leur environnement et les envoient toutes vers un seul nœud qui représente le centre de traitement.

É Densité de déploiement : Elle est plus élevée dans les RCSF que dans les réseaux Ad Hoc. Le nombre de nœuds capteurs peut atteindre des millions de nœuds pour permettre une meilleure granularité de surveillance. De plus, si plusieurs nœuds capteurs se retrouvent dans une région, un nœud défaillant pourra être remplacé par un autre. Cependant, la densité de déploiement donne naissance à des challenges pour la communication entre les nœuds. En effet, elle provoque des collisions ou des endommagements des paquets transmis.

É Absence d'adressage fixe des nœuds: Les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les

RCSF. Ces derniers utilisent un adressage basé sur l'attribut du phénomène capté, on parle donc de l'adressage basé-attribut. En effet, les requêtes des utilisateurs ne sont pas généralement destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés par un attribut. Par exemple, identifier un ensemble de nœuds par « les nœuds qui captent le volume du CO₂ dépassant 0,0375 % dans l'atmosphère ».

É Limitations de ressources physiques : A cause de la miniaturisation des composants électroniques, les performances des nœuds capteurs sont limitées. Par conséquent, les nœuds capteurs collaborent en traitant partiellement les mesures captées et envoient seulement les résultats à l'utilisateur. Une autre conséquence, ces limitations imposent des portées de transmission réduites contraignant les informations à être relayées de nœud en nœud avant d'atteindre le destinataire. C'est la raison pour laquelle les RCSF adoptent des communications multi-sauts.

É Sécurité: En plus des problèmes de sécurité rencontrés dans les réseaux Ad Hoc en général, les RCSF rencontrent d'autres handicaps dus à leurs challenges, à savoir l'autonomie et la miniaturisation des capteurs. Cela engendre l'inapplicabilité des mécanismes de défense utilisés dans les réseaux Ad Hoc tout en créant d'autres mécanismes de sécurité pour les RCSF. De plus, l'absence d'une sécurité physique dans l'environnement hostile où ils sont déployés expose les nœuds à un danger qui tend vers la falsification de l'information. En effet, les nœuds capteurs eux-mêmes sont des points de vulnérabilité du réseau car ils peuvent être modifiés, remplacés ou supprimés.

1.3.5 Architecture d'un réseau de capteurs

Les nœuds senseurs sont usuellement éparpillés dans un champ senseur. Chacun de ces nœuds senseurs a la possibilité de collecter une donnée et de router la donnée vers le nœud puits (sink). Les données sont routées en multi sauts vers le puits à travers une architecture sans infrastructure. Le nœud puits communique avec le nœud gestionnaire de la tâche à travers Internet ou par satellite. Le réseau de senseurs décrit par la figure suivante est influencé par plusieurs facteurs de conception incluant : tolérance aux fautes, extension (scalability), coût de production, environnement opératoire, topologie du réseau senseur, contraintes hardwares, média de transmission et l'énergie de consommation [LE 03].

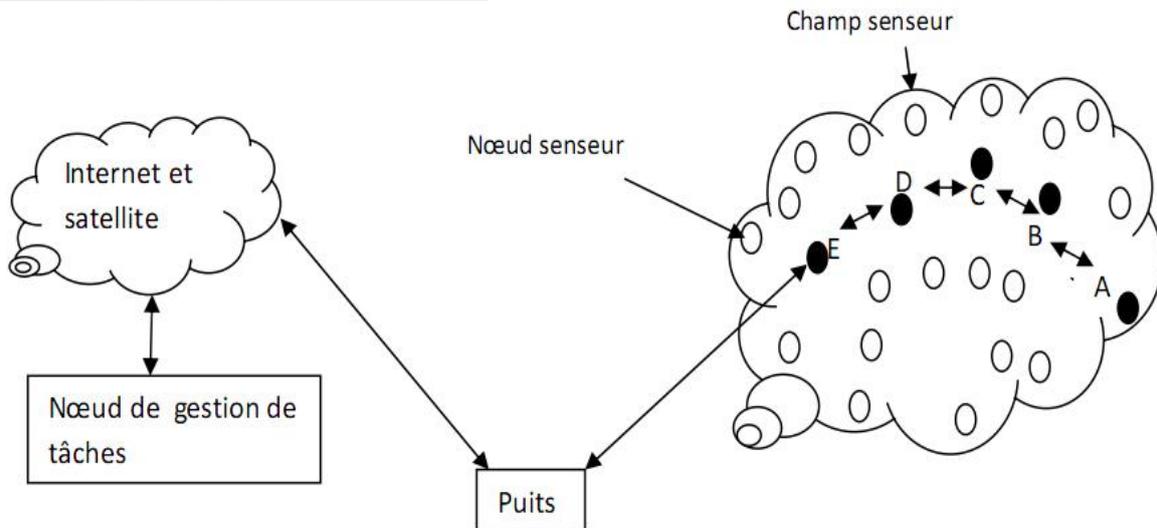


Figure 1.6: Architecture d'un RCSF

1.3.6 Architecture protocolaire des réseaux de capteurs

Dans les réseaux de capteurs sans fil, les nœuds capteurs utilisent une pile protocolaire spécifique en trois dimensions composée de cinq couches et de trois niveaux (niveau de gestion d'énergie, niveau de gestion de la mobilité, niveau de gestion des tâches). Les trois niveaux sont accessibles par toutes les couches de la pile. Il existe une interaction entre les différentes couches et les différents niveaux. [VH 03]

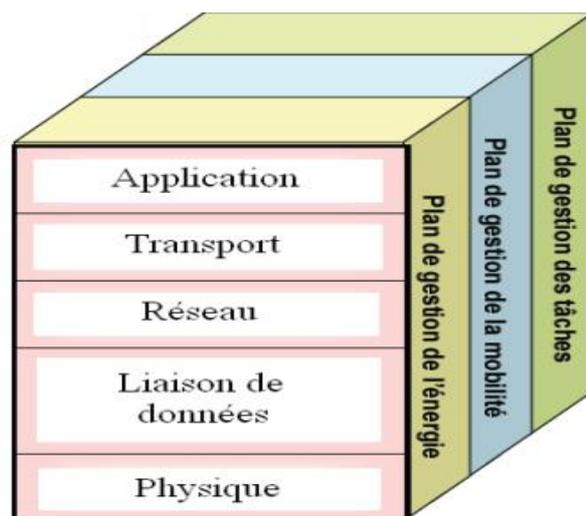


Figure 1.7: pile protocolaire des RCSF

- **Couche application :**

Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs, gérée directement par les logiciels. Parmi les protocoles d'application, nous citons : SMP (Sensor Management Protocol) et TADAP (Task Assignment and Data Advertisement Protocol).

- **Couche transport :**

Elle vérifie le bon acheminement des données et la qualité de la transmission. Dans les RCSF, la fiabilité de transmission n'est pas majeure. Ainsi, les erreurs et les pertes sont tolérées. Par conséquent, un protocole de transport proche du protocole UDP et appelé UDP-Like (User Datagram Protocol Like) est utilisé. Cependant, comme le protocole de transport universel est TCP (Transmission Control Protocol), les RCSF doivent donc posséder, lors d'une communication avec un réseau externe, une interface TCP-splitting pour vérifier la compatibilité entre ces deux réseaux communicants.

- **Couche réseau :**

Elle s'occupe du routage de données fournies par la couche transport. Elle établit les routes entre les nœuds capteurs et le nœud puits et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc. Les protocoles de routage conçus pour les RCSF sont différents de ceux conçus pour les réseaux Ad Hoc puisque les RCSF sont différents selon plusieurs critères comme :

- l'absence d'adressage fixe des nœuds tout en utilisant un adressage basé-attribut.
- l'établissement des communications multi-sauts.
- l'établissement des routes liant plusieurs sources en une seule destination pour agréger des données similaires, etc.

Parmi ces protocoles, nous citons : LEACH (Low-Energy Adaptive Clustering Hierarchy) et SAR (Sequential Assignment Routing).

- **Couche liaison de données :**

Elle est responsable de l'accès au media physique et la détection et la correction d'erreurs intervenues sur la couche physique. De plus, elle établit une communication saut-par-saut entre les nœuds. C'est-à-dire, elle détermine les liens de communication entre eux dans une distance d'un seul saut. Parmi les protocoles de liaison de données, nous citons: SMACS (Self-organizing Medium Access Control for Sensor networks) et EAR (Eavesdrop And Register).

- **Couche physique :**

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

1.3.7 Domaines d'applications des RCSF

Les capteurs ouvrent de nouveaux horizons à la gestion de l'information. Ils forment une "peau virtuelle" avec le monde réel qui nous informe des événements physiques se déroulant autour de nous. Cela a naturellement attiré plusieurs domaines d'applications qui ont mis en œuvre les facilités que les capteurs leur offrent. En voici une liste non exhaustive.

- **Le domaine militaire :**

Comme pour beaucoup d'autres domaines, les applications militaires ont été les locomotives de la recherche pour les réseaux de capteurs. Pour les militaires, un réseau de capteurs offre des avantages très précieux. Il s'agit d'un réseau qui s'installe rapidement, dynamiquement et sans aucune infrastructure. Ainsi, il offre un atout de taille pour surveiller les mouvements de l'ennemi, communiquer à bas coût entre les unités avec une logistique peu compliquée.

- **La surveillance environnementale :**

La petite taille et les capacités relativement grandes au niveau de calcul et de communication des capteurs permettent de les placer aux endroits que les humains ne peuvent ou ne veulent pas accéder, comme par exemple les grandes forêts, les volcans, les profondeurs des océans, les régions polaires, ou encore d'autres planètes que la terre. On peut aussi utiliser les WSN pour la surveillance du degré de maturité des récoltes (raisin), la mesure de la qualité de l'eau ou de l'air.

[GZ 02]

- **L'industrie :**

Les industriels s'intéressent au potentiel des capteurs pour diminuer les coûts du contrôle et de la maintenance des produits, de la gestion de l'inventaire, de la télésurveillance après vente, etc. En particulier, l'intégration de la technologie RFID avec les réseaux de capteurs est une des directions prometteuses de recherche dans l'industrie.

- **Les domaines urbains et domotique :**

Les capteurs entrent de plus en plus dans nos vies quotidiennes. Dans le milieu urbain, les capteurs sont déjà utilisés pour la localisation des bus, pour des tickets électroniques et pour la sécurité. Une des applications est la surveillance du trafic routier avec les réseaux de capteurs déployés sur les autoroutes [PC 04].

- **Le domaine médical**

La recherche sur l'usage des capteurs intelligents dans le domaine médical inclut les moyens d'hospitalisation à domicile, l'intégration des micro-capteurs "dans" le corps (e.g. construire un BAN - Body Area Network) et la gestion des urgences [NO 00]. Parmi les applications les plus utiles, on cite la télésurveillance des signes vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées ainsi que le contrôle à distance des données physiologiques.

1.4 Différentes problématiques dans les réseaux de capteurs

Les recherches dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques, parmi ces problématiques, nous citons :

- **Routage:**

Concevoir un protocole de routage 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...Ainsi le réseau doit passer à l'échelle sans que ses performances se dégradent.

- **Couche MAC :**

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. Ceci dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets.

- **Qualité de service :**

Des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, limiter les pertes de paquets vitaux pour la gestion du réseau, ou du moins en restreindre l'impact.

- **Cross-layer :**

Dans les modèles classiques, les concepteurs essaient d'optimiser les performances au niveau d'une couche indépendamment des autres couches. Par exemple, le routage et les fonctions de la couche MAC sont optimisés indépendamment les uns des autres. Cependant, une telle indépendance est communément considérée comme trop onéreuse pour les réseaux de capteurs. Par conséquent, une coopération permettant un compromis entre performances, dépendance et flexibilité doit être proposée pour optimiser les capacités du réseau en général.

- **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 intrinsèques imposées. Ainsi, pour concevoir un protocole efficace, il faudrait assurer une couverture maximale des capteurs composant le réseau (taux d'accessibilité supérieur 90%), minimiser le nombre des réémetteurs et des réceptions redondantes ainsi que la consommation d'énergie.

- **Sécurité :**

Pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein de leur communauté. Les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées que présentent ces réseaux.

1.5 Conclusion

Les réseaux de capteurs sans fil se propagent dans plusieurs domaines d'application. Ils sont devenus indispensables pour les mesures de certaines grandeurs physiques telles que la température, l'humidité, la vibration, etc. ou physiologiques.

Les caractéristiques de flexibilité, de tolérance aux fautes, de fidélité élevée de capture, les coûts bas et la rapidité de déploiement des réseaux de capteurs, créent beaucoup de nouveaux domaines d'application pour la télé-détection. Actuellement, cette large étendue d'applications fait que les réseaux de capteurs sont une partie intégrante de notre vie. Cependant, la réalisation des réseaux



Your complimentary
use period has ended.
Thank you for using
PDF Complete.

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

Généralités sur les réseaux de capteur sans fil

de capteurs doit effectivement satisfaire des contraintes telles que la tolérance aux fautes, la scalabilité, le coût, le matériel, le changement de topologie, l'environnement et la consommation efficace d'énergie.



*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

2. Routage dans les RCSF

2.1 Introduction

En général, le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné.

Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau, suivant certains critères de performance.

Dans les RCSF, l'utilisation des protocoles de routage conçus pour les réseaux Ad Hoc traditionnels est inappropriée.

Ceci est en raison des caractéristiques par lesquelles se distinguent les deux types de réseaux, d'où la nécessité de les améliorer ou de développer de nouveaux protocoles de routage spécifiques aux RCSF.

Plusieurs travaux de recherche dans ce domaine ont été effectués récemment et ont abouti à une multitude de protocoles destinés au RCSF.

Dans ce chapitre, nous commencerons par citer les facteurs de conception des RCSF par la suite on va donner une classification des protocoles de routage selon différents caractéristiques, pour terminer par quelques exemples de protocoles dédiés au RCSF.

2.2 Facteurs de conception des RCSF

La conception des réseaux de capteurs est influencée par de nombreux facteurs comme la tolérance aux pannes, les coûts de production, la consommation d'énergie, l'environnement ou la topologie du réseau. Ces facteurs représentent la base de la conception de protocoles ou d'algorithmes pour les réseaux de capteurs.

- **Tolérance aux pannes :**

Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences) 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 [KO 02].

- **Coût de fabrication :**

Les nœuds sont des produits fabriqués en série du fait de leur grand nombre. Il faut que le coût de fabrication de ces nœuds soit tel que le coût global du réseau ne soit pas supérieur à celui d'un réseau classique afin de pouvoir justifier son intérêt.

- **Topologie du réseau :**

En raison de leur forte densité dans la zone à observer, il faut que les nœuds capteurs soient capables d'adapter leur fonctionnement afin de maintenir la topologie souhaitée. On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau :

ÉDéploiement : Les nœuds sont soit répartis de manière prédéfinie soit de manière aléatoire (lancés en masse depuis un avion). Il faut alors que ceux-ci s'organisent de manière autonome.

ÉPost-Déploiement - Exploitation : Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.

ÉRedéploiement : L'ajout de nouveaux capteurs dans un réseau existant implique aussi une remise à jour de la topologie.

- **Consommation d'énergie :**

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner. Les réseaux de capteurs fonctionnant selon un mode de routage par saut, chaque nœud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un nœud implique un changement dans la topologie et impose une réorganisation du réseau.

2.3 Classification des protocoles de routage

Pendant la conception d'un protocole de routage pour réseau de capteurs sans fil, il est nécessaire de prendre en compte l'énergie et les ressources limitées des nœuds capteurs, la qualité du canal sans fil, la possibilité de perdre des paquets et la latence (temps d'attente).

L'efficacité en consommation d'énergie représente une métrique de performance significative qui influence directement la durée de vie du réseau en entier.

Plusieurs stratégies de routage ont été proposées pour les réseaux de capteurs sans fil.

Leurs principes de fonctionnement diffèrent suivant la philosophie de la classe à laquelle ils appartiennent. On donne une classification qui se base sur quatre critères :

- La topologie de réseau,
- Le fonctionnement du protocole,
- Le paradigme de communication,
- Le mode d'établissement des chemins.

2.3.1 Protocoles basés sur la structure du réseau

2.3.1.1 Routage plat

Dans le routage plat, tous les nœuds jouent typiquement le même rôle, les nœuds capteurs collaborent pour accomplir la tâche globale du réseau. En raison du nombre important des nœuds capteurs, il n'est pas faisable d'assigner un identifiant pour chaque nœud. Cette considération a mené au routage centré-données, où la station de base envoie des requêtes à certaines régions du réseau et attend des retours de données à partir des nœuds capteurs situés dans ces régions.

Puisque les données sont demandées par le biais des requêtes, la désignation des attributs est nécessaire pour indiquer les propriétés de ces données. Des premiers travaux sur le routage centré-données, tels que les protocoles SPIN et la diffusion dirigée, ont enregistré une économie d'énergie grâce à la négociation entre les nœuds du réseau et l'élimination des données redondantes.

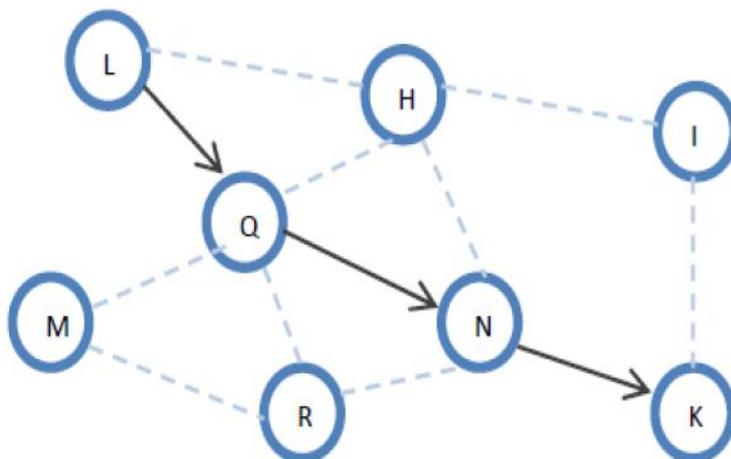


Figure 2.1: Topologie plate

2.3.1.2 Le routage hiérarchique

Dans cette architecture, il y a la notion de cluster : le réseau est subdivisé en clusters et chaque cluster est constitué de nœuds simples et d'un nœud leader. Seul le cluster-head communique avec les autres nœuds ou avec la station de base. Tous les nœuds d'un cluster envoient les données au cluster-head. Les avantages de cette classe sont que la sociabilité et la consommation efficace de l'énergie sont assurées. Les nœuds leader sont soit des super nœuds (des nœuds plus puissants comparés aux nœuds simples, i.e. possède un processeur plus puissant, une mémoire plus importante et des ressources d'énergie plus importantes) ou alors ils sont élus périodiquement suivant leurs quantité d'énergie restante (le nœud dont la quantité d'énergie restante est plus importante est élu cluster-head). Le cluster-head est le nœud responsable de la coordination des activités à l'intérieur du cluster et de la dissémination de l'information entre les clusters vers la station de base [AK 05].

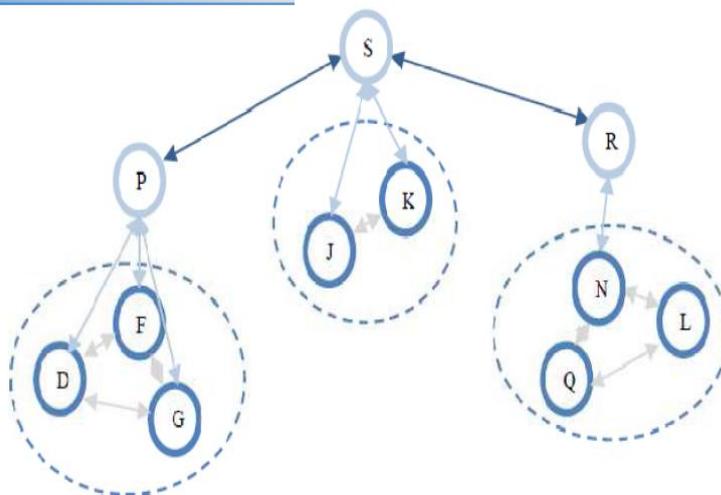


Figure 2.2: La topologie hiérarchique

2.3.2 Les protocoles de routage en fonction de leurs modes opératoires

2.3.2.1 Protocoles de routage multi-chemins

Les protocoles de cette sous-classe utilisent des chemins de routage multiples au lieu d'un chemin unique entre une source et une destination. La tolérance aux fautes d'un protocole est mesurée par la vraisemblance qu'un chemin alternatif existe entre une source et une destination lorsque le chemin principal échoue. Cette tolérance peut être renforcée en découvrant des chemins multiples entre la source et la destination aux dépens d'une consommation énergétique et d'un trafic de contrôle supplémentaires. Ces chemins alternatifs sont maintenus en veille par la source en envoyant des messages périodiques. Par conséquent, la fiabilité du réseau peut être augmentée tout en accusant une surcharge de contrôle supplémentaire pour garantir la validité des chemins alternatifs.

2.3.2.2 Le routage basé sur les requêtes

Dans le routage basé sur les requêtes, les nœuds propagent de voisin en voisin une requête pour données émise par un nœud du réseau, et les données correspondant à la requête seront envoyées au nœud qui a initié la requête.

Généralement, ces requêtes sont décrites dans un langage naturel ou des langages d'interrogation de niveau élevé. Par exemple, dans une application de type militaire, le client C1 peut soumettre une requête au nœud N1 et demander : Y a-t-il des véhicules en mouvement dans la région 1 du champ de bataille ? Tous les nœuds disposent des tables de requêtes reçues pour des tâches de

capture, et envoient les données associées aux tâches une fois obtenues. La diffusion dirigée est un exemple de ce type de routage, où le nœud station de base envoie des messages d'intérêt aux nœuds capteurs. Pendant que le paquet d'intérêt est propagé dans tout le réseau, les gradients de la source à la station de base sont de nouveau installés.

Lorsque la source a des données pour l'intérêt, elle les envoie le long du chemin du gradient d'intérêt. Pour diminuer la consommation énergétique, l'agrégation de données est exécutée sur la route.

2.3.2.3 Routage basé sur la négociation

Ces protocoles emploient des descripteurs de données à un niveau élevé afin d'éliminer la transmission des données redondantes sur la base de la négociation.

Des décisions de communication sont également prises sur la base des ressources disponibles au niveau des nœuds capteurs. SPIN est un exemple de protocoles de routage via la négociation. La motivation principale réside dans le fait que l'utilisation de l'inondation pour dissémination produira la duplication des données envoyées, ainsi les nœuds recevront les copies doubles des mêmes données. Cette opération consomme un surplus d'énergie et de traitement en envoyant les mêmes données par différents capteurs. SPIN est conçu pour disséminer les données d'un nœud à tous les autres nœuds, en supposant que ces capteurs sont de potentielles stations de base. Donc, l'idée principale du routage via la négociation est de supprimer l'information double et d'empêcher l'envoi des données redondantes au prochain capteur ou à la station de base, en échangeant une série de messages de négociation avant même la transmission effective des données. [KA 04]

2.3.3 Routage en fonction d'établissement des chemins

2.3.3.1 Les protocoles proactifs

Dans les protocoles proactifs les routes sont créées à l'avance dans une phase d'initialisation. Les tables de routage des nœuds sont enregistrées dans tous les nœuds. Si un changement de topologie survient, la modification des tables de routage doit être diffusée.

2.3.3.2 Les protocoles réactifs

Les protocoles réactifs sont des algorithmes où les routes sont créées à la demande, c'est-à-dire, dès qu'une donnée doit être routée. La donnée parcourt un chemin créé dynamiquement

saut par saut. Dans chaque nœud, un traitement de la donnée et un calcul (dépend du mécanisme de routage) sont effectués pour choisir le prochain saut.

2.3.3.3 Les protocoles hybrides

Les protocoles hybrides utilisent une combinaison des deux principes. Par exemple, une partie des routes est découverte au préalable dans une phase d'initialisation. Durant la transmission de données, s'il n'y a pas de chemins prédéterminés, les nœuds créent dynamiquement le chemin vers la station de base. [RJ 07]

2.3.4 Routage en fonction de paradigme de communication

2.3.4.1 Node-centric

Ce paradigme se base sur l'identification individuelle des nœuds participant à la communication. Les différents mécanismes de propagation se focalisent sur un découpage logique en sous réseaux. Ce genre de paradigme est très utilisé dans les réseaux ad-hoc. Il s'intègre bien avec l'utilisation de ce type de environnement. Cependant, il est très peu utilisé dans les réseaux de capteurs sans fils car il est très gourmand en termes de ressources. Il est généralement utilisé pour les applications qui nécessitent une interrogation individuelle des capteurs.

2.3.4.2 Data-centric

Ce paradigme est basé sur le fait que la donnée est plus importante que le nœud lui-même dans un réseau de capteurs sans fils. Dans ce paradigme, toutes les communications sont identifiées par leurs données et le routage doit être réalisé en se basant sur cette propriété. Le système peut être vu comme une base de données distribuée où les nœuds forment des tables virtuelles alimentées par les données captées. En abandonnant l'identification des nœuds, un capteur perd sa valeur individuelle, qui sera attribuée aux données.

2.3.4.3 Position-centric

Dans la majorité des applications WSN, la donnée est associée à une position (par exemple une position est toujours associée à la température captée dans une application de détection de feu). Position-centric se base sur ce principe et utilise la position des nœuds comme moyen principal de routage et d'adressage. Dans ce cas, le routage s'effectue grâce à des techniques géométriques afin d'acheminer l'information d'une zone géographique vers une autre.

Cette approche présente des avantages et des inconvénients. Elle ne nécessite aucune table de routage car le nœud achemine ses paquets seulement en se basant sur la position de la destination et sur des informations locales sur ces voisins directs. Un autre avantage important est l'indépendance de la mobilité. La même approche peut être utilisée même si les nœuds changent de position d'une façon continue (WSNs mobiles). Cependant, ce type de mécanismes nécessite le déploiement d'une solution de positionnement, dont le degré de précision requis dépend de l'application ciblée. L'utilisation du GPS reste trop coûteuse dans un WSN. Pour cela, plusieurs méthodes de localisation et de positionnement des autres capteurs ont été développées [DN 05].

2.4 Exemples de protocoles de routage dans les WSN

2.4.1 Protocoles plats

2.4.1.1 SPIN (détecteur d'information par négociation)

L'un des protocoles de routage opérant par la négociation est désigné par l'acronyme SPIN. Cette famille de protocoles adaptatifs est proposée pour pallier au problème d'ignorance de ressources confronté dans la technique d'inondation en utilisant la négociation et l'adaptation aux ressources disponibles.

La conception des protocoles SPIN est basée sur deux idées :

- a) Si les nœuds capteurs se limitent, dans leurs rapports, sur l'envoi des données qui décrivent les informations captées et non pas l'envoi de ces dernières en entier (telles que les images), ils peuvent opérer avec plus d'efficacité et économiser ainsi une quantité considérable d'énergie.
- b) Les nœuds capteurs doivent contrôler en permanence les changements dans leurs ressources énergétiques.

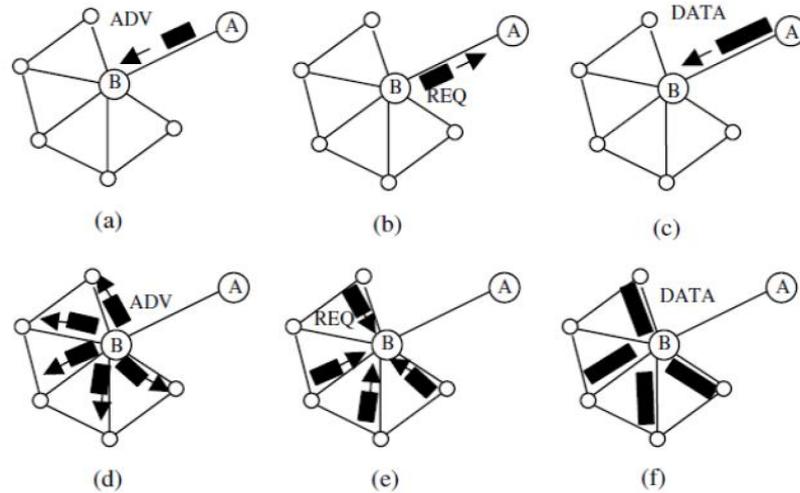


Figure 2.4 : fonctionnement de protocole SPIN

Il identifie les données avec un descripteur haut niveau ou une métadonnée. La caractéristique principale de SPIN est qu'avant une transmission, des métadonnées sont échangées à travers un mécanisme d'annonce de donnée. Chaque nœud qui reçoit une nouvelle donnée l'annonce à tous ses voisins. Les voisins intéressés récupèrent la donnée en envoyant un message de demande (une requête).

Les métadonnées résolvent les problèmes classiques du flooding (donnée redondante) et économisent l'énergie. Il n'y a pas de standard de métadonnée. Il y a trois types de paquets définis dans SPIN :

ADV : quand un nœud a une donnée à envoyer, il avertit ses voisins en utilisant ce message.

REQ : un nœud envoie ce message s'il désire recevoir une donnée.

DATA : ce message contient la donnée avec un entête contenant la métadonnée.

Avant d'envoyer un message DATA, le nœud capteur diffuse un message ADV qui contient le descripteur, c'est-à-dire la métadonnée de la donnée DATA, comme le montre la figure dans (a). Les nœuds qui reçoivent ce paquet vérifient si les données les intéressent. Si oui, ils répondent par un paquet REQ. Le nœud qui a initié la communication envoie alors un paquet DATA pour chaque réponse REQ reçue. Le nœud capteur voisin répète ce processus. Ensuite, chaque nœud qui fait office de relais peut très bien agréger ses propres données aux données qui sont déjà contenues dans le paquet.

2.4.1.2 EAR (Energy Awar Routing)

Le problème avec les protocoles actuels est qu'ils trouvent le chemin optimal en énergie et l'utilisent pour toutes les communications. Cependant, ce n'est pas la meilleure des méthodes pour prolonger la durée de fonctionnement d'un réseau. Utiliser fréquemment le même chemin optimal induit l'épuisement de la batterie des nœuds qui constituent ce chemin et, par conséquent, le réseau sera partitionné.

Shah et Rabaey ont proposé l'usage d'un ensemble de chemins sous optimales de façon à augmenter occasionnellement la durée de vie du réseau. Ces chemins sont choisis en utilisant une fonction probabiliste qui dépend de la consommation d'énergie sur chaque chemin. Le critère de survie du réseau est considéré alors comme la métrique principale ciblée par ce type de protocoles. EAR est un protocole réactif, la communication est initiée par la destination. La station de base initie le chemin et le maintient, et plusieurs chemins sont maintenus entre la source et la station de base.

Le protocole EAR travaille en trois phases :

- **Phase d'installation :**

La station de base initie les communications. Elle diffuse un paquet de requêtes dans tout le réseau (flooding) pour trouver tous les chemins vers le nœud source qui intéresse l'utilisateur. Le paquet requête contient un champ coût.

La station de base initie ce champ coût à 0.

Le coût total en énergie est calculé pour chaque nœud. Ainsi, si la requête est envoyée du nœud n_i vers le nœud n_j , ce dernier calculera le coût induit sur ce chemin par :

$$C_{n_i, n_j} = C_{n_i, n_k} + C_{n_k, n_j}$$

Avec :

C_{n_i, n_j} : est le coût énergétique du saut de n_i vers n_j

Le nœud affecte une probabilité à chacun de ces voisins dans la table de routage correspondant à un chemin formé. La probabilité est inversement proportionnelle au coût.

$$P_{n_i, n_j} = \frac{1/C_{n_i, n_j}}{\sum_{n_k \in N} 1/C_{n_i, n_k}}$$

Avec :

$n_k \in N$ où n_k est un voisin de n_i

Le nœud n_i calcule alors le coût moyen pour atteindre la destination, en utilisant les voisins dans la table d'envoi, et ceci en utilisant la formule suivante :

$$C_{n_i} = \frac{\sum_{k \in \text{voisins}(n_i)} C_{n_k} \cdot P_{n_i, n_k}}{\sum_{k \in \text{voisins}(n_i)} P_{n_i, n_k}}$$

Avec :

C_{n_k} où n_k est dans la table de routage de n_i

Si le C_{n_i} d'un saut est trop élevé par rapport à un seuil déterminé par l'utilisateur de l'algorithme : $(C_{n_i} > \text{seuil})$ le nœud est éliminé de la table de routage :

$$C_{n_i} > \text{seuil} \Rightarrow n_i \text{ éliminé}$$

Où k est un identifiant de la table de routage du nœud n_i .

- **Phase de communication de données :**

Une fois que les tables de routage de tous les nœuds sont construites, le nœud source envoie le paquet de données vers un nœud voisin (à partir de sa table de routage) selon une probabilité. La probabilité qu'un voisin soit choisi est celle qui se trouve dans la table de routage T_{n_i, n_j} . Les nœuds voisins font de même jusqu'à ce que le paquet arrive à destination.

La métrique d'énergie : est un paramètre important de l'algorithme. Ce paramètre peut contenir plusieurs informations : l'énergie restante dans un nœud, la topologie du réseau, le coût d'utilisation du chemin. Les concepteurs de cet algorithme ont choisi une métrique simple :

$$E_{n_i} = \frac{E_{n_i}^{\text{restant}}}{E_{n_i}^{\text{initiale}}}$$

Ou :

E_{n_i} : est l'énergie utilisée pour une transmission et réception du nœud i vers le nœud j .

E_{n_i} : Énergie résiduelle dans le nœud i normalisée à l'énergie initiale du nœud.

et α et β sont des facteurs de poids choisis pour trouver le chemin d'énergie minimum. Ils contrôlent la pondération du chemin.

- **Phase de maintenance des routes :**

Cette phase a pour but de mettre à jour les chemins entre les nœuds du réseau et la station de base. Elle reprend les opérations de la première phase c'est-à-dire détecte toutes les occurrences du flooding. Cette phase est lancée périodiquement.

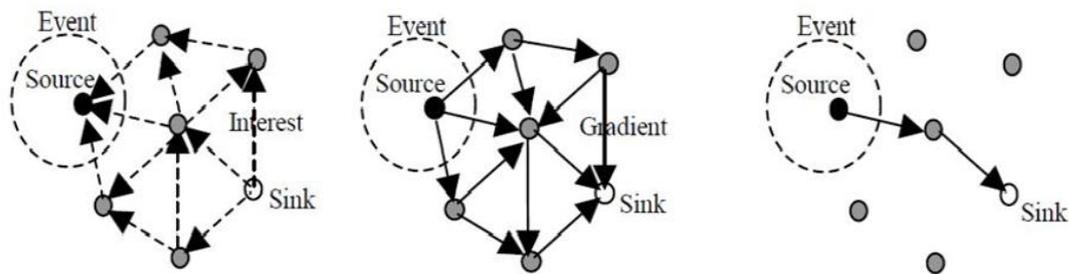
2.4.1.3 DD (Directed Diffusion)

Cet algorithme, est basé sur le paradigme de dissémination de données, suivant lequel, le nœud puits diffuse son intérêt, qui présente une description des tâches affectées à tous les nœuds capteurs du réseau. Chaque intérêt est nommé par attribution de pairs (attribut, valeur). Le champ attribut représente un nom d'objet (par exemple : intervalle, durée d'envoi des données...). Le champ Valeur représente la valeur qu'a prise l'attribut durant le captage. L'intérêt (la requête) est diffusé par le puits (la station de base) vers les voisins. Chaque nœud qui reçoit un intérêt l'enregistre dans sa mémoire cache pour une utilisation ultérieure. Les intérêts sont des descriptions de tâches pour les nœuds. Ils sont utilisés pour la comparaison avec les données reçues. La raison principale derrière l'utilisation d'un tel système est de se débarrasser des opérations inutiles de routage de couche réseau afin d'économiser l'énergie.

Le gradient est un coût acheminé vers le voisin d'où provient l'intérêt pour renforcer le trafic dans ce chemin. Ce chemin est caractérisé par : le débit de données, la durée d'envoi des données, la date d'expiration obtenue à partir des intérêts reçus. Les chemins sont établis entre le puits (station de base) et les sources, grâce aux gradients et intérêts. En se basant sur les caractéristiques des chemins, un chemin est choisi et renforcé si le débit de données est faible. Le puits envoie le message original à travers le chemin choisi et renforce les nœuds sources de ce chemin. Le renforcement des chemins est réalisé dans le but d'envoyer des données plus fréquemment.

La réparation d'un chemin entre une source et le puits est effectuée s'il y a une panne. Un nouveau chemin est identifié pour remplacer le chemin en panne. La réinitialisation du renforcement se fait en se basant sur les chemins où sont envoyées les données avec un débit réduit.

En résumé, le protocole DD repose sur trois étapes comme le montre la figure suivante :



a) propagation des intérêts. b) construction de gradient. c) livraison des données à travers un chemin renforcé.

Figure 2.5 : les phases de communication du protocole Directed Diffusion

2.4.1.4 RR (Rumor routing)

Routing Rumor est une variante du protocole Directed Diffusion. Il élimine la notion d'inondation aveugle pour la propagation des intérêts et essaye de trouver un compromis entre l'inondation des intérêts et la propagation des données. Ce protocole se base sur une topologie plate et utilise un routage hybride. La communication au niveau de ce protocole est initiée par la source à l'observation d'un événement.

Rumor Routing se situe entre la diffusion d'événements et la diffusion de requêtes. L'idée est de cheminer les requêtes vers les nœuds qui ont observé un événement assez particulier au lieu d'inonder tout le réseau de requêtes pour rapporter les informations à propos de l'événement.

Dans le but de diffuser des événements à travers le réseau, l'algorithme emploie des paquets à longue vie nommé : AGENT. Quand un nœud reçoit un agent d'événement, il met à jour sa table locale et/ou la table de l'agent pour les entrées communes, suivant le nombre de sauts optimal. C'est à dire que si l'agent possède une route plus courte vers un certain événement, le nœud met à jour sa table ; et si le nœud connaît des événements non connus par l'agent, il ajoute les entrées nécessaires dans la table de l'agent. Avant de quitter le nœud, son identificateur est ajouté à la liste des nœuds visités par l'agent. La durée de vie de l'agent TTL (Time-To-Live) est décrétementée d'un pas et si cette durée reste toujours positive, l'agent est renvoyé vers un des voisins choisi d'une façon aléatoire parmi les nœuds non encore visités.

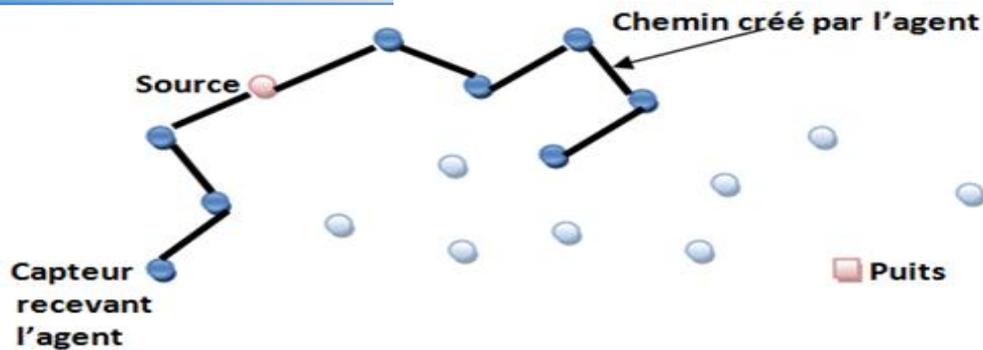


Figure 2.6 : Agent événement

Les résultats des simulations effectuées dans [BR 02] nous indiquent que Rumor Routing minimise la consommation d'énergie de manière importante par rapport à l'inondation par événement et qu'il prend en compte les pannes des nœuds. Néanmoins, cet algorithme est performant seulement si le nombre d'événements est réduit, car les tables d'agents et les tables de relais au niveau des nœuds augmentent très rapidement de taille, ce qui engendre une très grande consommation en termes d'énergie (le parcours des agents de très grande taille) et en termes de mémoire (les tables de relais locales au niveau des nœuds)

2.4.2 Protocoles hiérarchiques

2.4.2.1 EEUC (An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks)

Li et al ont proposé le protocole EEUC [CL 05] qui aborde le problème des points chauds (hot spots) les CHs qui entourent la station de base vont mourir plus vite que les autres CHs, car tout le trafic du réseau destiné à la station de base passe par ces nœuds et donc ont plus de charge que les nœuds distants, en particulier lors de l'adoption du routage multi-sauts comme mécanisme de transmission. L'idée principale derrière EEUC est d'utiliser un mécanisme de partitionnement de taille inégale pour grouper les nœuds lors du processus de clustering afin d'équilibrer la consommation d'énergie des CHs, et résoudre le problème des points chauds. Pour ce faire, les clusters les plus proches de la station de base ont des tailles plus petites que ceux qui sont plus loin, ce qui va permettre une diminution de la charge des CHs entourant la station de base et donc leur offrir la possibilité de conservation d'énergie.

EEUC procède d'une manière concurrente pour choisir les CHs. Initialement, tous les nœuds ont la même probabilité pour devenir des CHs candidats, où chaque nœud tire un nombre aléatoire entre

1 et 0. Si le nombre tiré est inférieur à un certain seuil prédéfini T , le nœud s'endormira. Sinon, il devient CH candidat et diffuse par la suite un message de compétition pour annoncer sa décision. Ce message contient l'identité du nœud, le diamètre du rayon de concurrence et son niveau d'énergie résiduelle. Le diamètre du rayon de concurrence est calculé en fonction de sa distance à la station de base. En fonction de ces diamètres, une répartition contrôlée de CHs dans le réseau est exercée.

Dès qu'on s'éloigne de la station de base, le diamètre de rayon de concurrence ainsi que la taille des clusters sont augmentés (plus les CHs sont proches de la station de base plus la taille des clusters sera petite).

Au niveau de chaque CH candidat, une liste de ses CHs candidats adjacents est maintenue.

La décision de devenir CH final est faite en se basant sur le niveau d'énergie résiduelle, de sorte que le CH candidat doté de plus d'énergie résiduelle parmi ceux maintenus au niveau de la liste, prévaut. En cas d'égalité, le nœud de plus petite identité est choisi. Par la suite, le nœud gagnant (CH final) diffuse un message aux autres nœuds CHs candidats voisins concurrents pour les informer. Ces derniers se retirent et abandonnent la concurrence dès qu'ils reçoivent ce message.

Une fois les CHs finaux sont sélectionnés, ils diffusent des messages d'annonce à travers le réseau. Et selon la force du signal reçu, chaque nœud rejoint le cluster du CH le plus proche, tout en confirmant son appartenance par un message de jointure. Pour organiser la communication intra-clusters EEUC adopte le même mécanisme utilisé dans LEACH (combiner les deux protocoles TDMA et CDMA pour collecter les données des nœuds membres). Or que pour le routage inter-clusters, une nouvelle méthode est introduite. Selon un seuil calculé, les CHs envoient leurs données à la station de base. Si la distance qui leur sépare à la station de base est inférieur à ce seuil, les données sont transmises directement à cette dernière, sinon les CHs devront trouver le meilleur nœud relais (le plus adjacent et celui doté de plus d'énergie résiduelle) pour router les données et y parvenir à la station de base.

2.4.2.2 HEED (Hybrid Energy-Efficient Distributed Clustering)

Younes et Fahmy [FAH] ont proposé un algorithme de clustering distribué appelé HEED pour les réseaux de capteurs ad hoc. HEED ne fait aucune restriction sur la distribution et la densité des nœuds ainsi qu'il ne dépend pas de la topologie du réseau et sa taille. HEED sélectionne les cluster-heads selon un critère hybride regroupant l'énergie restante des nœuds et un second paramètre tel que le degré des nœuds. Il vise à réaliser une distribution uniforme des

cluster-heads dans le réseau et à générer des clusters équilibrés en taille. Un nœud u est élu comme cluster-head avec une probabilité PCH égale à :

$$PCH_u = \frac{E_u}{\sum_{u \in N} E_u} \cdot \frac{1}{N_{opt}}$$

Où :

- E_u : est l'énergie restante du nœud u ,
- $\sum_{u \in N} E_u$: l'énergie globale dans le réseau,
- N_{opt} : est le nombre optimal de clusters.

Cependant, l'évaluation de N_{opt} présente certaine difficulté puisque HEED opère sans d'autres protocoles de routage et en absence de toute commande centrale ainsi que calcule le nombre optimal de clusters N_{opt} . D'autre part, avec HEED, la topologie en clusters ne réalise pas la consommation minimale d'énergie dans les communications intracluster et les clusters générés ne sont pas tellement équilibrés en taille.

2.4.2.3 PEGASIS (Power-Efficient Gathering in Sensor Information Systems)

Dans [RA 02], Lindsey et Raghavendra ont proposé une version améliorée de LEACH appelée

PEGASIS. L'idée principale de PEGASIS est de former une chaîne entre les nœuds de sorte que chaque nœud reçoive et communique à un voisin proche.

Les données collectées sont transmises d'un nœud à un autre qui les agrège jusqu'à ce qu'elles arrivent à un nœud particulier qui les transmet à la station de base. Les nœuds qui transmettent les données à la station de base, sont choisis tour à tour selon un round-robin dans le but est de réduire l'énergie moyenne dépensée par un nœud durant une période (round).

Contrairement à LEACH, PEGASIS évite la formation des clusters et procure à un seul nœud dans la chaîne, l'envoi de données à la station de base.

Les résultats de simulation ont montré que PEGASIS peut prolonger de deux à trois fois la durée de vie d'un réseau de capteurs relativement à LEACH en fonction du critère choisi pour évaluer la durée de vie d'un réseau i.e. quand 1%, 20%, 50% ou 100% des nœuds épuisent leurs batteries. Un tel gain de performance est réalisé par l'élimination de l'overhead causé par le processus de formation de clusters dans LEACH, et par réduction du nombre de transmissions et de réceptions par utilisation de l'agrégation de données. Bien que l'overhead du clustering soit évité, PEGASIS

exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données. Cependant, un tel ajustement de la topologie pourrait causer un overhead important en particulier dans les réseaux les plus utilisés. En outre, PEGASIS suppose que tout nœud est capable de communiquer directement avec la station de base. Or, cette supposition est loin de la réalité, car les capteurs communiquent généralement en mode multi-sauts pour atteindre la station de base. D'autre part, PEGASIS suppose que tous les nœuds maintiennent une table contenant les localisations de tous les autres nœuds dans le réseau.

En résumé, PEGASIS est adapté seulement aux capteurs sans fil dont les nœuds sont immobiles. Son évaluation dans des environnements mobiles pourrait dégrader considérablement ses performances.

2.4.2.4 LEACH (Low Energy Adaptive Clustering Hierarch)

Parmi les premiers protocoles de routage hiérarchiques conçus pour les réseaux de capteurs sans fil, on retrouve le protocole LEACH [WR 00]. Pour une meilleure allocation de ressources et contrôle d'énergie, ce protocole s'appuie sur le principe de partitionnement du réseau en clusters, selon l'amplitude du signal reçu. Les nœuds organisés en clusters transmettent leurs données aux CHs, qui à leur tour, fusionnent ces données rassemblées et les envoient vers la station de base. De ce fait, les CHs doivent posséder plus d'énergie que les nœuds membres.

Le fonctionnement de ce protocole est faite cycliquement où chaque cycle est décomposé en deux grandes phases: initialisation et transmission. Initialement, les clusters sont formés et les cluster-heads sont sélectionnés. Les nœuds s'auto-élisent pour être des CHs en se basant sur le pourcentage de CHs désiré pour ces derniers, et le nombre de cycles pendant lequel un nœud a joué le rôle de CH. Ainsi, un nœud choisit une valeur aléatoire entre 0 et 1. Si ce nombre est inférieur à une valeur seuil $T(n)$, ce nœud se déclare CH pour le cycle courant. Les nœuds qui ont été élus comme cluster-heads informent alors leur voisinage de leur élection par envoi de message. A la réception des annonces faites par ces CHs, les nœuds non cluster-heads, selon la puissance du signal reçu, déterminent le cluster auquel ils veulent appartenir puis communiquent leurs choix au CH choisi.

$$\frac{1}{1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}}$$

Le système fixe le nombre de clusters en fonction de quelques paramètres comme la topologie du réseau, le coût des opérations de communication et de calcul. Généralement, la valeur optimale de CHs représente 5% des nœuds du réseau. Après la réception de tous les messages des nœuds désirant être membres d'un cluster, les CHs élus peuvent alors déterminer le nombre de ces membres et les ordonner par la suite, en créant des tables TDMA¹ (Time Division Multiplexed Access) afin d'éviter les collisions intra-cluster. En outre, un code CDMA² est choisi par les CHs afin d'être utilisé par les nœuds de son cluster lors de la phase de transmission de données durant laquelle chaque nœud ne peut émettre ses données vers le CH de son cluster que durant l'intervalle alloué. Ce dernier va collecter toutes les données reçues, les agréger et les router à la station de base.

2.4.2.5 DSLEACH (Density of Sensor based LEACH)

Le protocole DSLEACH est un protocole qui étudie la densité des nœuds durant les différents ronds de réseau. A cause de la grande distance entre certains nœuds et le CH, il ne va pas avoir de données envoyées de ces nœuds vers la SB, d'autre part, l'espace de basse densité de nœuds doit être laissé sans CH. En LEACH, tous les nœuds utilisent la même formule et probabilité pour être CH ; cependant en DSLEACH, les nœuds dans un espace de basse densité ont une grande probabilité et les nœuds dans un espace de grande probabilité ont une petite probabilité de devenir CH. Similaire à Leach DSLEACH inclut deux grandes phases:

Phase d'initialisation : Durant cette phase, les nœuds rentrent en concurrence pour devenir CH, quelques uns de ces nœuds réussissent de devenir CH, et les autres sont des membres des clusters. La probabilité (Pi) de devenir CH décroît quand le nombre de ronds (r) est décrétement. Après l'estimation de (Pi) chaque nœud choisit un nombre aléatoire (x) entre 0 et 1 et le compare avec (Pi), si Pi > x alors le nœud se déclare CH, sinon il doit se présenter au sein d'un autre CH pour devenir membre de son cluster. Les CHs diffusent un message

¹ Le schéma d'accès multiple à répartition de temps permet de diviser le temps en intervalles (*time-slot*) attribués à chaque nœud. Ainsi, un seul nœud a le droit d'accès au canal mais doit émettre ses données pendant les intervalles de temps qui lui sont accordés.

² Le schéma d'accès multiple par répartition en code permet de côtoyer plusieurs nœuds simultanément. En effet, il ne divise ni la plage de fréquences ni l'intervalle de temps. Ainsi, des nœuds peuvent émettre leurs données continuellement et selon une large plage de fréquence. Le protocole CDMA utilise des techniques d'étalement de spectre afin d'éviter les collisions entre les transmissions simultanées des nœuds.

d'advertisement en utilisant le protocole CSMA MAC. Et les nuds membres attendent la réception d'un signal de la part des CHs. Tout nud non CH décide à quel cluster il va appartenir en se basant sur la force de signal reçu par les CHs voisins, puis envoie une demande d'appartenance au CH correspondant, utilisant le protocole CSMA MAC protocole.

Phase de transmission : Au cours de cette étape tout membre envoie la donnée collectée vers son CH et à la fin, chaque CH crée un message (signal) de toutes les informations agrégées et l'envoie vers la SB. Après un certain temps déterminé à priori, le rond courant termine et un nouveau rond commence.

2.4.2.6 CVLEACH (Coverage based energy efficient LEACH)

CVLEACH est la couverture en fonction d'énergie efficace d'algorithme LEACH qui augmente le nombre de nuds membres d'un cluster en créant non chevauchement de grappes. CVLEACH travaille également en ronds - qui sont fixés à la quantité de temps pendant laquelle le nud sélectionné ne peut agir en tant que tête cluster.

Phase d'initialisation : Comme LEACH, au début de chaque round chaque nud sélectionne un nombre aléatoire compris entre 0 et 1 et calcule la valeur de seuil $P_i(t)$, Si le nombre aléatoire sélectionné est inférieur à la valeur seuil $P_i(t)$, le nud s'auto élire en tant que chef de cluster. Toutefois, avant qu'un nud puisse s'annoncer comme un chef de cluster, il vérifie s'il a reçu un message annoncé par autres chefs de clusters, si c'est le cas il se retire en tant que tête de groupe au cours de ce tour et décide de se joindre à l'un des groupes. Un nud envoie un message pour rejoindre l'un des groupes pour lesquels l'énergie minimum est requise pour la communication avec le chef de groupe. Cet algorithme assure également qu'un nud peut devenir tête de groupe uniquement une fois dans 1 tour.

Phase de transmission : Durant cette phase si un nud est CH alors il peut collecter les données des nuds membres, les agréger et les envoyer vers la SB. Les nuds membres seront activés et peuvent envoyer leurs données au CH selon l'ordonnanceur TDMA envoyé par le CH. Pour le reste de temps TDMA, les nuds membres passent à l'état sommeil pour économiser l'énergie et à la fin du round actuel les nuds endormis se réveillent pour être prêts pour le round suivant.

Remarque :

Ces trois derniers protocoles seront bien détaillés dans le chapitre trois.

Protocole de routage	Type	QoS	Agrégation de données
SPIN	PLAT	-	✓
Directed diffusion	PLAT	-	✓
EAR	PLAT	-	-
CADR	PLAT	-	-
COUGAR	PLAT	-	✓
LEACH	Hiérarchique	-	✓
PEGAGIS	Hiérarchique	-	✓
AQUIRE	PLAT	-	-
TEEN	Hiérarchique	-	✓
MCF	PLAT	-	-
HEED	Hiérarchique	-	-
MECN	Géographique	-	-
GAF	Géographique	-	-
TTDD	Hiérarchique	-	-
GEAR	Géographique	-	-
Rumor	PLAT	-	✓
SPEED	Géographique	-	-
GBR	PLAT	-	✓
SAR	Géographique	✓	-

Table 1 : Résumé de quelques protocoles de routage pour les RCSF.

2.5 Conclusion

Le routage dans les réseaux de capteurs est un domaine de recherche d'une grande fertilité. Nous avons tenté dans ce chapitre de faire une description globale des techniques en usage dans ce domaine, ayant comme objectif commun l'augmentation de la durée de vie du réseau, sans compromettre la livraison des données.

Bien que tous ces protocoles se révèlent prometteurs, ils ne répondent cependant pas à tous les défis que rencontrent ceux qui les déploient.



*Your complimentary
use period has ended.
Thank you for using
PDF Complete.*

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

3. Protocoles LEACH, et ses variantes

3.1. Introduction

Les réseaux de capteurs sans fil se distinguent par leur densité importante des nœuds, leur autonomie énergétique limitée, leur topologie mobile, etc. Ces contraintes ont posé plusieurs défis pour la conception et la gestion des réseaux de capteurs sans fil. L'un de ces défis est la mise en revue et le développement des nouveaux protocoles de routage adéquats à la nature particulière des réseaux de capteurs sans fil. Il arrive que plusieurs nœuds dans un réseau de capteurs sans fil combinent les données capturées localement et les communiquent à la station de base (SB). Il apparaît donc qu'il existe des différences considérables entre les réseaux de capteurs sans fil et les réseaux traditionnels. C'est pourquoi il n'est pas toujours possible d'utiliser les mêmes protocoles. Dans notre étude, nous nous sommes basés sur le routage basé clustering, il consiste à partitionner le réseau en groupes (cluster), où dans chacun d'eux une seule nœud est sélectionné comme leader pour jouer le rôle spécial de point de transfert.

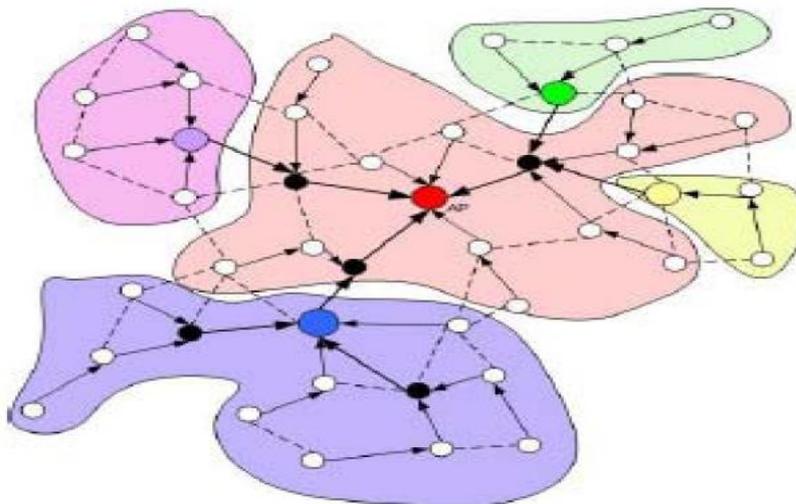


Figure 3.1 : routage basé clustering

Le protocole de routage de base dans ce type est bien le protocole LEACH, son principe est la division du réseau en un ensemble de clusters selon la force reçue du signal et d'utiliser les CHs (Clusters -Heads) comme des routeurs pour passer les données à la station de base. Dans ce chapitre nous présenterons une étude sur le protocole de base LEACH, ensuite une analyse détaillée sur ces deux variantes DSLEACH et CVLEACH

3.2 Le protocole de base LEACH : (*Low Energy Adaptive Clustering Hierarchy*)

LEACH [WR 00] est un protocole auto organisateur basé sur le clustering adaptatif, qui utilise la rotation randomisée des têtes de grappe pour distribuer équitablement la charge d'énergie entre les nœuds capteurs dans le réseau. Il est considéré comme étant l'une des premières approches de routage hiérarchique basées sur le clustering. LEACH est fondé sur deux hypothèses de base :

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

L'idée derrière LEACH est de former des clusters des nœuds capteurs selon la force reçue du signal et d'utiliser les têtes locales de grappe (cluster-Head) comme des routeurs pour conduire des données à la station de base.

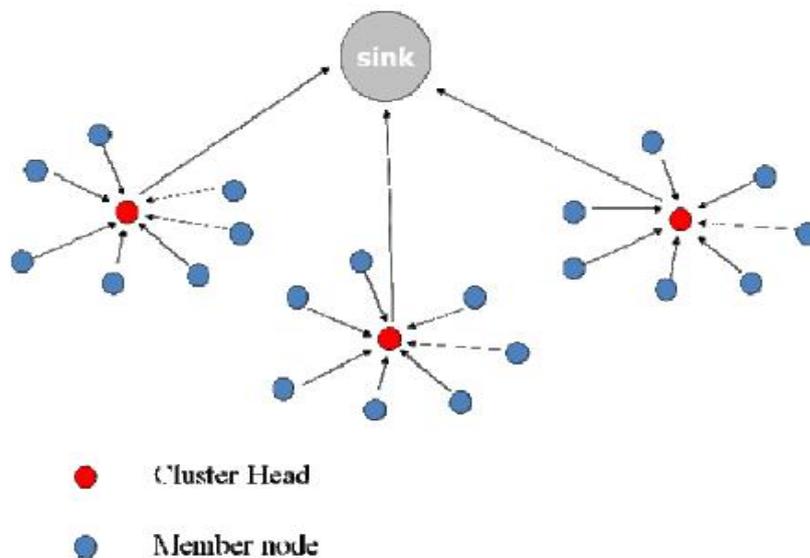


Figure 3.2: un réseau avec clustering

Les dispositifs principaux de LEACH sont :

- Coordination et contrôle localisés entre les nœuds : pour l'initialisation et le traitement de grappe.
- Rotation randomisée de cluster : effectuée par "la station de base" ou "les têtes de cluster".

- Compression locale (*agrégation*) : Les nœuds CH compressent les données arrivant des nœuds appartenant à leurs grappes respectives, et envoient un paquet d'agrégation à la station de base afin de réduire la quantité d'information qui doit être transmise à la station de base.

Dans LEACH, le traitement est séparé dans des cycles de longueur constante, où chaque cycle commence par une phase d'initialisation suivie d'une phase de transmission. La durée d'un cycle est déterminée.

3.2.1 Phase d'initialisation

La phase d'initialisation repose sur trois sous-phases comme l'indique la figure 3.3 :

La phase d'annonce, la phase d'organisation des groupes et la phase d'ordonnement, qui permettent la formation des clusters et l'élection des CHs qui seront détaillées ci-dessous.

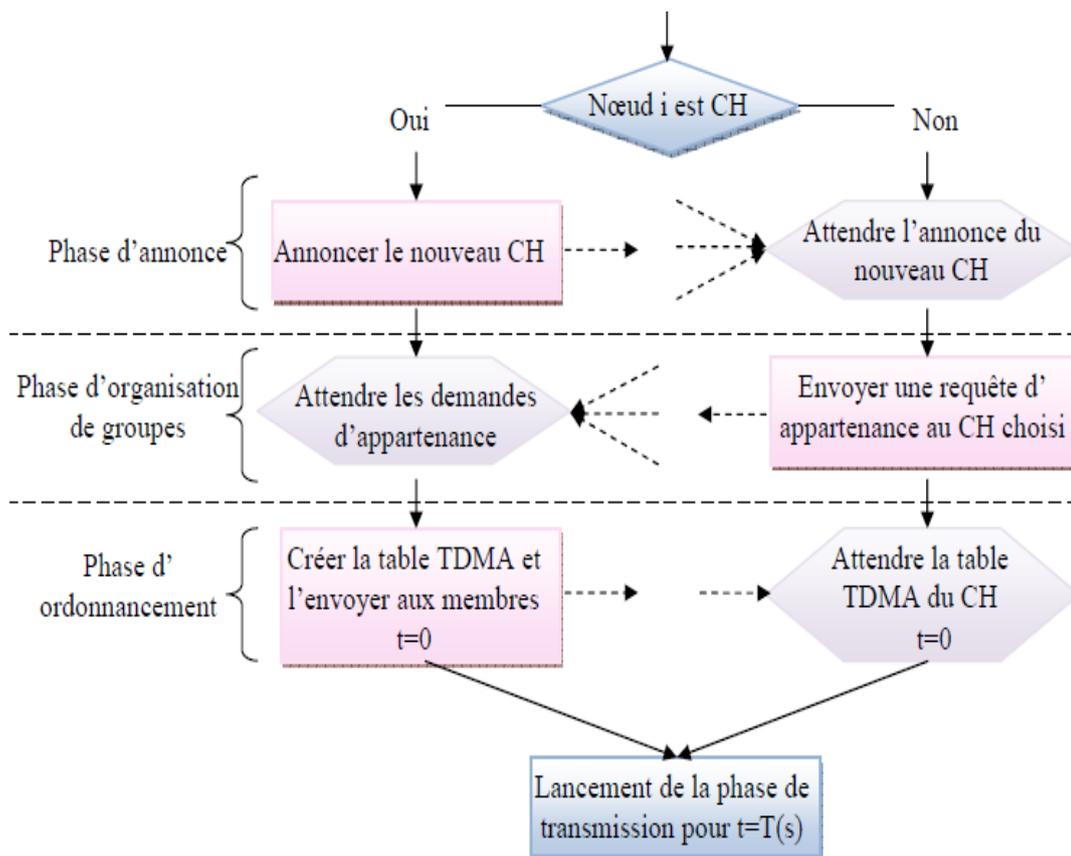


Figure 3.3 : Opération de l'étape d'initialisation de LEACH

3.2.1.1 phases d'annonce

Avant de lancer cette phase, on désire avoir un certain nombre de CH. Ce nombre, que l'on note K , est fixe et il est inchangé durant tous les rounds. On estime que le pourcentage optimal du nombre de CH désirés devrait être de 5% à 15% du nombre total de nœuds. Si ce pourcentage n'est pas respecté, cela mènera à une grande dissipation d'énergie dans le réseau. En effet, si le nombre de CH est très élevé, on aura un nombre important de nœuds (CH) qui se consacrent aux tâches très coûteuses en ressources énergétiques. Ainsi, on aura une dissipation d'énergie considérable dans le réseau. De plus, si le nombre de CH est très petit, ces derniers vont gérer des groupes de grandes tailles. Ainsi, ces CH s'épuiseront rapidement à cause de travail important qui leur est demandé. Cette phase commence par l'annonce du nouveau round par le nœud puits, et par la prise de décision locale d'un nœud pour devenir CH avec une certaine probabilité P_i au début du round r qui commence à l'instant t . Chaque nœud i génère un nombre aléatoire entre 0 et 1. Si ce nombre est inférieur à P_i , le nœud deviendra CH durant le round r . P_i est calculé en fonction de K et de round r :

$$P_i = \frac{K}{N} \cdot \frac{1}{r}$$

Où :

N est le nombre total de nœuds dans le réseau.

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

$$P_i = \frac{K}{N} \cdot \frac{1}{r}$$

$$P_i = \frac{K}{N} \cdot \frac{1}{r} \quad \text{si } i \text{ est élu } (1)$$

Où $CH_i(t)$ égal à 0 si le nœud i a déjà été CH durant l'un des $(t - t_{CH_i}) / T_{round}$ rounds précédents, et, il est égal à 1 dans le cas contraire. Donc, seuls les nœuds qui n'ont pas encore été CH, ont vraisemblablement une énergie résiduelle suffisante que les autres et ils pourront être choisis. La somme des $CH_i(t)$ ($i=0 \dots N$) représente le nombre total des nœuds éligibles d'être CH à l'instant t . Il est égal à :

$$\sum_{i=0}^{N-1} CH_i(t) = N - \sum_{i=0}^{N-1} \left((t - t_{CH_i}) / T_{round} \right) \quad (2)$$

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

$$N_{CH} = \left(N - \sum_{i=0}^{N-1} \left((t - t_{CH_i}) / T_{round} \right) \right) \times \frac{1}{N} = \frac{N - \sum_{i=0}^{N-1} \left((t - t_{CH_i}) / T_{round} \right)}{N}$$

La probabilité P_{CH} est basée sur la supposition que tous les nœuds sont initialement homogènes et commencent avec la même quantité résiduelle d'énergie et meurent approximativement en même temps.

3.2.1.2 La phase d'Organisation de groupes

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

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

3.2.1.3 la phase d'ordonnancement

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

aléatoirement un code dans une liste de codes de propagation CDMA. Il le transmet par la suite à ses membres afin de l'utiliser pour leurs transmissions.

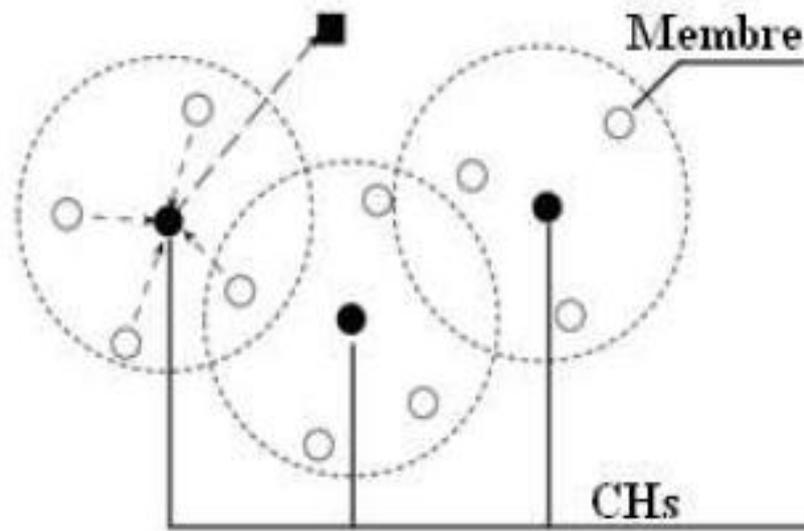


Figure 3.4 : Interférence lors d'une communication dans LEACH

3.2.2 Phase de transmission

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

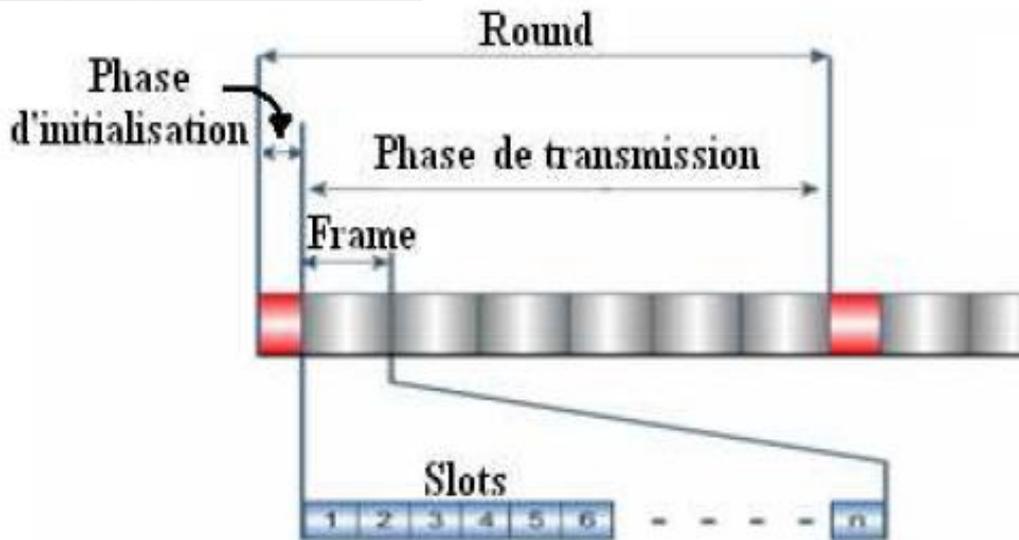


Figure 3.5 : Répartition du temps et différentes phases pour chaque round

3.2.3 Avantages et limites

Le protocole LEACH engendre beaucoup d'avantages en ce qu'il offre comme bonne manipulation de ressources du réseau en respectant plusieurs contraintes telle que la consommation d'énergie.

Bien que LEACH puisse augmenter la durée de vie du réseau, il présente certaines limitations.

Il choisit aléatoirement la liste des clusters heads et il ne pose aucune contrainte sur leurs distributions ainsi que sur leurs niveaux d'énergie. Donc, les clusters heads peuvent se concentrer dans un même endroit et par conséquent, il pourrait exister des nœuds isolés (sans cluster head).

3.3. Le protocole DSLEACH (*Density of Sensor based LEACH*)

DS-LEACH [JB 09] est un algorithme basé, qui étudie la densité locale des nœuds durant les différents rounds de réseau de clustering et obtient des améliorations pour les prochains rounds s'il ya un événement imprévu dans le réseau (comme l'échec d'un nœud ou sa mort) l'algorithme doit avoir une grande agilité pour recevoir le minimum de désastre. DS-LEACH est basé sur LEACH et similaire à lui, il contient quatre sous étapes : annonce, clustering, Schedule création (TDMA) et transmission des données.

La principale amélioration par rapport à LEACH est effectuée à la phase d'annonce et le clustering où le nœud se présente comme CH. Dans DS-LEACH on considère la densité des nœuds comme critère quand on forme les CHs.

L'un des problèmes de LEACH est le manque de participation de quelques nœuds dans les clusters à cause de la grande distance entre ces nœuds et les CHs. Donc, il n'y aura pas de données envoyées du capteur non membre de cluster à la SB, d'autre part l'espace à basse densité de nœuds va être laissé sans CH. En LEACH tous les nœuds utilisent la même probabilité et formule pour devenir CH, par contre, en DSLEACH les nœuds dans une basse densité vont avoir une grande probabilité de devenir CH, et les nœuds dans une grande densité auront une basse probabilité de devenir CH.

L'expérience montre que environ 12% des nœuds ne sont pas attribués à un cluster dans LEACH, DSLEACH tente de diminuer ce pourcentage. Pour réaliser ça, nous ne devons pas attendre des informations supplémentaires plus que celles exigées en LEACH.

On assume qu'il y a des rounds où à chaque round les nœuds sont divisés en cluster, similaire à LEACH chaque round inclue deux grandes phases : la phase d'initialisation et la phase de transmission.

3.3.1 Principe de fonctionnement

3.3.1.1 Phase d'initialisation

Elle inclue la phase d'annonce, phase de clustering et création de Schedule. Les senseurs rentrent en compétition pour devenir CH, où chaque nœud i calcule la probabilité de devenir CH par la formule suivante :

$$P_i = \max \left\{ \frac{1}{n_i}, 0 \right\}$$

Où :

- n_i indique le nombre de fois dont l'em nœud est devenu CH.
- n_i est le nombre de nœud dans le cluster, ou le nœud i a été son CH au passé proche (nombre de clusters initiale).
- t : est le numéro de rond courant (commence par 1).
- P_i : donne la probabilité de nœud i pour devenir CH au round t

Initialement α_i doit être fixé à un nombre aléatoire supérieur à α_0 (le nombre de cluster initial nommé aussi pourcentage de CH).

Après quelques rounds, si le nœud i devient CH, donc α_i doit être mis à jour pour devenir le nombre de nœuds dans ce cluster. La valeur de α_i doit être mise à jour à chaque fois que i devient CH.

La valeur de α_i dépend de α_0 , α_i et r . Si α_i est grand c.à.d. que les clusters ont une grande densité, puis la probabilité que α_0 devient CH à nouveau (au futur prêt) est diminué.

Si α_i est grand c.à.d. α_0 est devenu CH plusieurs fois donc la probabilité que α_0 devient CH à nouveau est diminué. Et la probabilité augmente quand le nombre de ronds r augmente.

Après estimation de la valeur de α_i , chaque nœud génère un nombre aléatoire x entre 0 et 1 et compare x avec α_i :

- Si $\alpha_i > x$ donc α_0 se déclare comme CH.
- Sinon, il attend le message d'avertissement pour choisir auquel cluster va appartenir.

Chaque nœud qui se déclare comme CH pour le round courant doit diffuser un message d'avertissement utilisant le protocole CSMA MAC (Medium Access Control Carrier Sens Multiple Access) aux autres nœuds. Les nœuds qui n'ont pas été élus CH vont recevoir un signal des CHs.

À l'étape de clustering, chaque nœud non CH décide à quel cluster il devrait appartenir (en se basant sur le signal reçu de différents CHs) et envoie un message d'appartenance au CH correspondant utilisant aussi le protocole CSMA MAC.

À l'étape de création de Schedule, chaque CH crée un TDMA Schedule et le diffuse sur ces membres dans le cluster. Ce Schedule détermine chaque nœud quand il envoie ces informations au CH.

3.3.1.2 Phase de transmission

À la phase de transmission chaque membre envoie les données collectées à son CH selon Schedule TDMA. À la fin, chaque CH crée une agrégation de données et l'envoie vers la SB. Après un temps déterminé a priori, le round courant terminera et un nouveau round commence.

3.3.1 Avantages et limites

Le protocole DSLEACH offre une bonne méthode d'équilibrage dans la distribution de la charge sur les nœuds de réseau grâce à la probabilité qu'il utilise afin de construire des clusters dans les zones les moins denses .

Malgré que DSLEACH réduise le nombre de nœud non couvert dans le réseau, l'expérience montre que 4% des nœuds de réseau n'ont pas de CH.

3.4 Le protocole CVLEACH (*Coverage based energy efficient LEACH*)

L'énergie nécessaire pour communiquer avec le CH par les nœuds membres est très inférieure par rapport à l'énergie requise par les CHs ou des nœuds non membres à communiquer avec la SB. CVLEACH [KO 12] repose sur le principe de la couverture de la région à surveiller en fonction de la consommation d'énergie de l'algorithme LEACH. Il repose sur le même principe du protocole LEACH.

3.4.1 Le principe du fonctionnement

3.4.1.1 Phase d'initialisation

Au début de chaque round, chaque nœud :

- Sélectionne un nombre aléatoire x entre 0 et 1 ;
- Calcule une valeur du seuil θ ou θ_{min} donnée dans l'équation (1).

Si le nombre x est inférieur à la valeur du seuil θ , ce nœud est élu lui-même comme CH.

Toutefois, avant qu'un nœud puisse s'annoncer comme un CH parmi les autres, il vérifie s'il a reçu une annonce d'un autre CH, il décidera alors de quitter son rôle en tant que CH au cours de ce round et décide d'adhérer à un quelconque des clusters déjà formés. Un nœud envoie un message de la jointure à un des clusters formés dans lequel l'énergie minimale caractérise la communication avec le CH (i.e. le CH le plus proche). Cet algorithme assure également qu'un nœud puisse devenir un CH uniquement une fois dans un $1/p$ ronds.

3.4.1.2 Phase de transmission

Si un nœud est un CH alors il peut collecter les données des nœuds membres, les agréger et envoyer la donnée résultante à la station de base. Les nœuds membres seront activés et peuvent envoyer leurs données au CH selon l'ordonnanceur TDMA, pour le reste temps TDMA, les

Les nœuds membres passent à l'état sommeil (SLEEP) pour économiser leur énergie et à la fin de round actuel les nœuds endormis se réveillent pour être prêts pour le round suivant.

Si un nœud n'est ni CH ni un nœud membre alors il envoie ses données directement vers la station de base.

3.4.2 Les principales améliorations

Les principales améliorations sont effectuées dans la phase d'initialisation. CVLEACH fonctionne comme LEACH sauf que si un nœud est élu en tant que CH lors du dernier round q où $q < r$ ou qu'il a reçu l'annonce de CH à partir des nœuds voisins au cours d'un round particulier, ce nœud retire sa participation pour être élu en tant que CH au cours de son prochain round et décide de rejoindre l'un des clusters à partir duquel il a reçu l'annonce de CH, et le nœud qui n'est ni CH et ni un nœud membre envoie ses données directement à SB contrairement au LEACH et DSLEACH les données des nœuds qui ne sont ni membres ni CHs sont ignorées.

3.4.3 Avantages et limites

CVLEACH offre une couverture totale du réseau et il augmente le nombre de nœuds membres d'un cluster en évitant le chevauchement de grappes. Ce qui permet à tous les nœuds d'envoyer leurs données collectées à la station de base, par conséquent la réduction du nombre de nœuds inutiles (non membres). Mais l'envoi direct vers la SB par les nœuds non membres consomme plus d'énergie que l'envoi vers les CH puis vers la SB.

3.5 Conclusion

Après avoir présenté le concept de base ainsi que l'architecture de fonctionnement de protocole LEACH et ses deux variantes CVLEACH et DSLEACH on va entamer dans le chapitre qui suit l'étape de test qui constitue un point important dans le processus d'analyse des performances de ces algorithmes.



Your complimentary
use period has ended.
Thank you for using
PDF Complete.

[Click Here to upgrade to
Unlimited Pages and Expanded Features](#)

4. Tests et comparaison

4.1 Introduction

Dans le domaine des réseaux de capteurs sans fil, la simulation est une étape incontournable lorsqu'on veut tester et évaluer des modèles d'application ou des protocoles de communication. L'expérimentation réelle s'avère quelques fois très coûteuse. De plus, la simulation offre un gain considérable en temps, une flexibilité en permettant la variation des paramètres et une meilleure visualisation des résultats sous forme de graphes faciles à analyser et interpréter. Cependant, la simulation ne peut pas remplacer l'implémentation réelle. La validité du modèle simulé ne garantit pas le bon déroulement de son implémentation réelle. Des erreurs de programmation peuvent toujours survenir au moment de l'implémentation du simulateur. L'étape de la simulation doit être suivie par une implémentation physique pour vérifier le comportement réel des modèles. Dans ce chapitre, l'analyse des performances des algorithmes (LEACH, DSLEACH et CVLEACH) est évaluée à l'aide du simulateur OMNET++.

4.2 Présentation de quelques simulateurs existants

4.2.1 NS 2

« NS2 » désigne la deuxième version du NS (Network Simulator) qui a été développée à Lawrence Berkeley National Laboratory (LBNL) [NS 2]. C'est un simulateur à événements discrets dans le temps. Il a été conçu pour être un simulateur générique. Par conséquent, il constitue un support important pour la simulation des protocoles standards tels que TCP, IP, et ce, aussi bien à travers les réseaux filaires qu'à travers les réseaux sans fils. Il supporte également les réseaux ad hoc et les réseaux de capteurs et comprend quelques protocoles implémentés tel que Directed Diffusion.

NS est un simulateur multicouches et son développement suit une approche orienté-objet en utilisant deux langages de programmation : C++ et OTCL (Object Tools Command Langage dérivé de TCL). Effectivement, le noyau, les modèles, les protocoles et les composants de base sont écrits en C++, tandis que les interfaces sont implémentées en OTCL.

Des scénarios de simulation, écrits en OTCL, peuvent être introduits afin de décrire les conditions de la simulation, essentiellement, la topologie du réseau, les caractéristiques des liens physiques, les protocoles utilisés et les communications qui ont eu lieu. L'utilisation de l'OTCL permet aussi à l'utilisateur de créer ses propres procédures. Une fois le scénario défini, NS entame la simulation et génère un fichier de trace « .log ». Ce fichier est interprété par l'outil

NAM (Network Animator), associé au simulateur NS, afin de visualiser des animations de la simulation (transfert des paquets, taille des paquets, remplissage des files d'attente, etc.).

4.2.2 J-Sim

J-Sim (<http://sites.google.com/site/jsimofficial/>) est une plateforme de simulation de réseaux écrite en Java, orientée objets, basée composant et à événements discrets. Des modules peuvent être ajoutés et supprimés par plug-and-play, J-Sim est utilisé pour la simulation et l'émulation de réseau en incorporant un ou plusieurs dispositifs réels de capteur. Cette plateforme fournit le support pour la modélisation de l'objet cible, les nœuds capteurs et les stations de base, les canaux de capture et les voies de transmission sans fil, les canaux sismiques, les modèles de puissance et les modèles d'énergie.

4.2.3 TinyOs

TinyOS (<http://pcl.cs.ucla.edu/projects/TinyOs/>) est un système d'exploitation intégré, modulaire, destiné aux réseaux de capteurs miniatures. Cette plateforme logicielle ouverte et une série d'outils développés par l'Université de Berkeley est enrichie par une multitude d'utilisateurs. En effet, TinyOS est le plus répandu des OS pour les réseaux de capteurs sans-fil. Il est utilisé dans les plus grands projets de recherches sur le sujet (plus de 10.000 téléchargements de la nouvelle version). Un grand nombre de ces groupes de recherches ou entreprises participent activement au développement de cet OS en fournissant de nouveaux modules, de nouvelles applications,... Cet OS est capable d'intégrer très rapidement les innovations en relation avec l'avancement des applications et des réseaux eux-mêmes tout en minimisant la taille du code source en raison des problèmes inhérents de mémoire dans les réseaux de capteurs. La librairie TinyOS comprend les protocoles réseaux, les services de distribution, les drivers pour capteurs et les outils d'acquisition de données. TinyOS est en grande partie écrit en C mais on peut très facilement créer des applications personnalisées en langages C, NesC, et Java.

4.2.4 OMNeT++

OMNeT++ « Objective Modular Network Test-bed in C++ » (OMNeT++ Community - www.omnetpp.org. 2001-2009.) Est un environnement de simulation basé composant, modulaire ayant une architecture ouverte.

Son domaine d'application primaire est la simulation des réseaux de transmission.

OMNeT++ est une application développée par Andras Varga, chercheur à l'université de Budapest. Il possède une architecture générique de sorte qu'il peut être utilisé pour la modélisation des réseaux câblés, les communications sans fil, la modélisation de protocoles, la modélisation des multiprocesseurs de systèmes distribués, l'évaluation des aspects de la performance des systèmes logiciels complexes, la validation d'architectures matérielles, etc.í

4.3 Le choix de notre simulateur

Pour évaluer les performances de nos protocoles, le simulateur OMNet++4.1 a été utilisé comme outil d'expérimentation. Généralement, le choix d'un simulateur dépend de ses performances d'exécution et ses réponses aux besoins de l'utilisateur. Pour cela, notre choix d'OMNeT se justifie par le fait qu'il est totalement paramétrable programmable et modulaire.

Il fournit des outils, des modèles et des infrastructures qui aident à réaliser une simulation théorique. Les modèles sont assemblés à partir des composants réutilisables appelés *modules*. Ces modules peuvent être combinés de différentes manières.

Les simulations par OMNeT ++ peuvent être exécutées sous différentes interfaces d'utilisateur : interfaces graphiques, interfaces d'animation et les interfaces utilisateurs de commande en ligne. De plus, il est disponible pour les systèmes d'exploitation les plus courants (Linux, Mac OS / X, Windows).

4.3.1 Les principaux éléments d'OMNeT

OMNeT++ est un environnement de développement intégré qui est basé sur la plateforme Eclipse, et l'étend avec de nouveaux éditeurs, des vues, des assistants et des fonctionnalités supplémentaires. La capture d'écran suivante présente les principaux éléments d'OMNeT IDE.

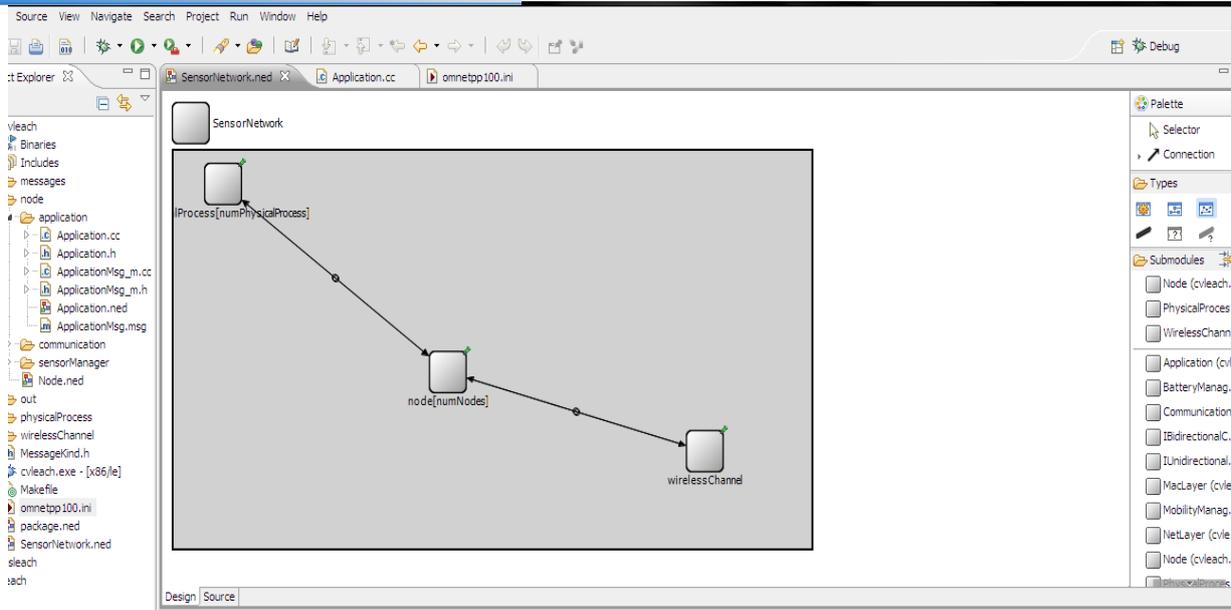


Figure 4.1 : Les principaux éléments de OMNeT IDE

4.3.1.1 L'éditeur NED

L'éditeur peut éditer des fichiers NED forme graphique ou en mode texte, et l'utilisateur peut basculer entre les deux modes, à tout moment, en utilisant les onglets au bas de la fenêtre de l'éditeur.

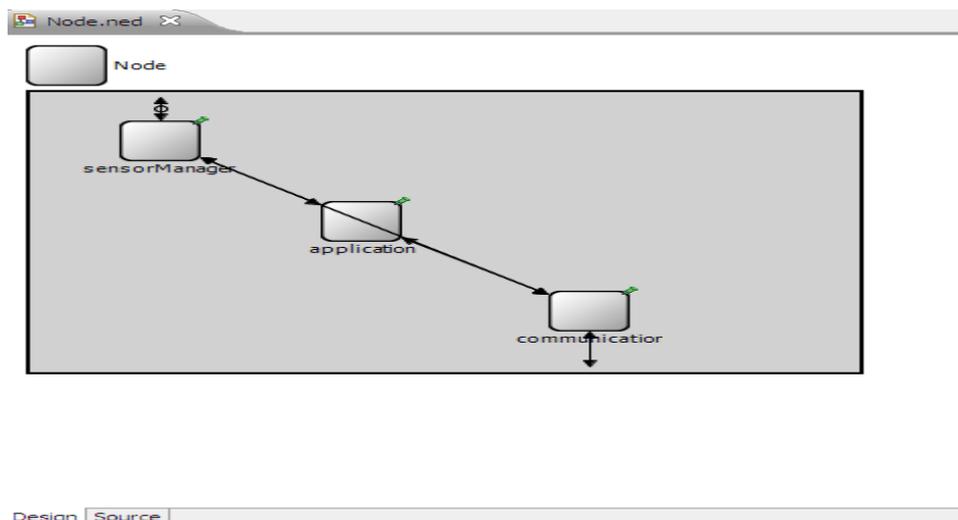
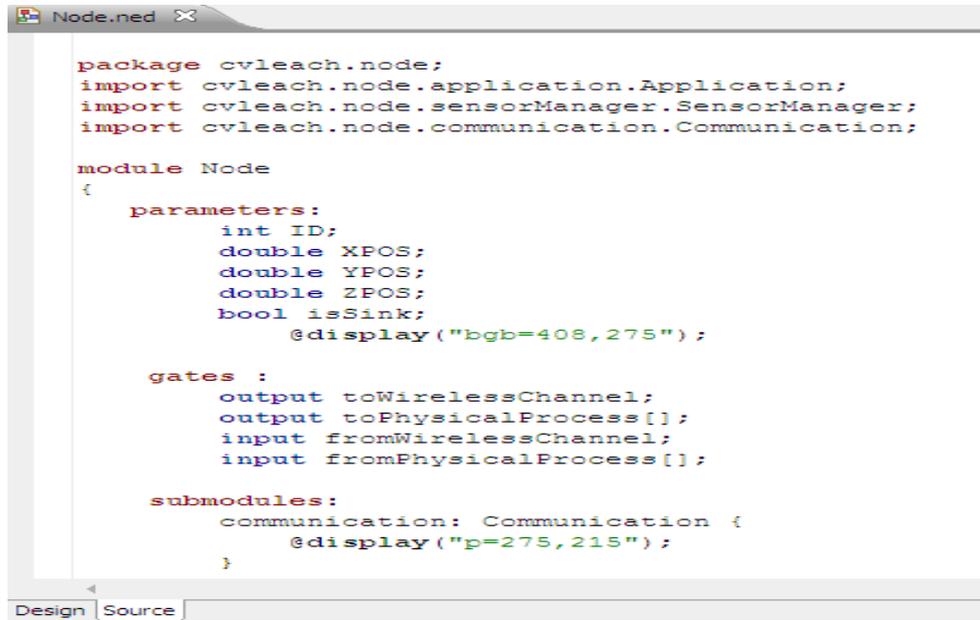


Figure 4.2: Le mode graphique de fichier NED.

En mode graphique, on peut créer des modules composés, les canaux et les types de composants et autres. Les sous-modules peuvent être créés en utilisant la palette de types de modules disponibles.



```
Node.ned
package cvleach.node;
import cvleach.node.application.Application;
import cvleach.node.sensorManager.SensorManager;
import cvleach.node.communication.Communication;

module Node
{
    parameters:
        int ID;
        double XPOS;
        double YPOS;
        double ZPOS;
        bool isSink;
        @display ("bgb=408,275");

    gates :
        output toWirelessChannel;
        output toPhysicalProcess[];
        input fromWirelessChannel;
        input fromPhysicalProcess[];

    submodules:
        communication: Communication {
            @display ("p=275,215");
        }
}
```

Figure 4.3: Le mode texte de fichier NED.

Le mode texte permet à l'utilisateur de travailler avec la source NED directement. La source NED est constamment analysé et validé en tant que l'utilisateur tape, et les erreurs sont affichées en temps réel sur la marge de gauche. La coloration syntaxique, l'indentation automatique et la conversion automatique de OMNeT NED syntaxe sont également fournis.

4.3.1.2 L'éditeur de fichiers INI

L'éditeur de fichiers INI permet à l'utilisateur de configurer des modèles de simulation pour l'exécution. Il dispose également de mode forme et mode source.

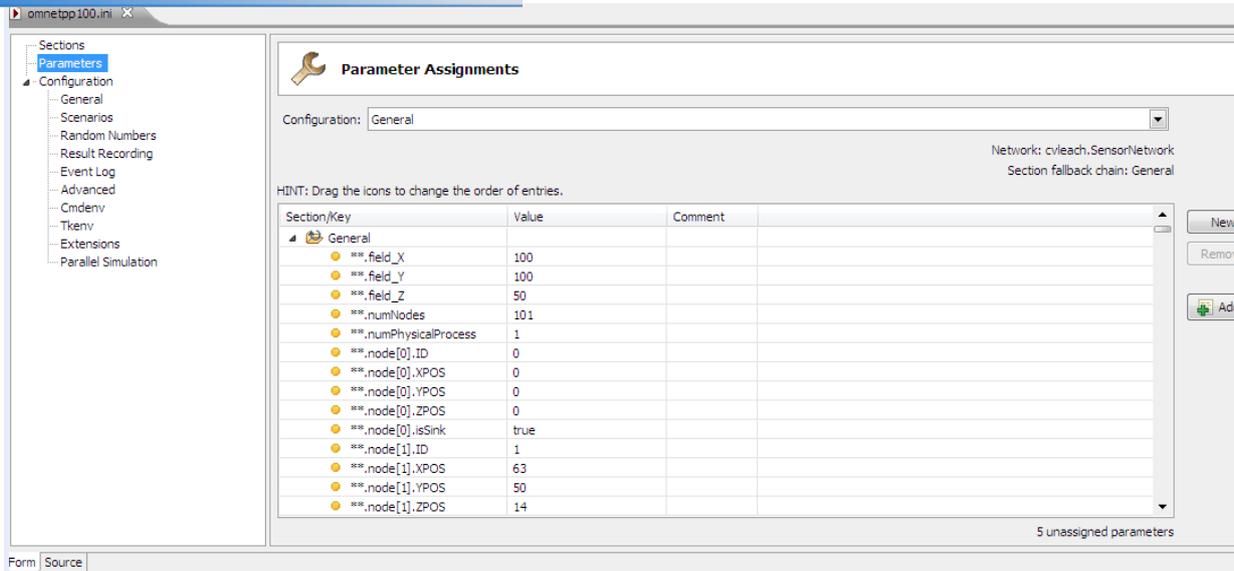


Figure 4.4: L'éditeur de fichier INI.

L'éditeur de fichier INI considère toutes les options de configuration prises en charge et leurs offres sous plusieurs formes, organisées par thèmes. Les descriptions et les valeurs par défaut sont affichées dans les info-bulles, qui peuvent être persistantes pour faciliter la lecture.

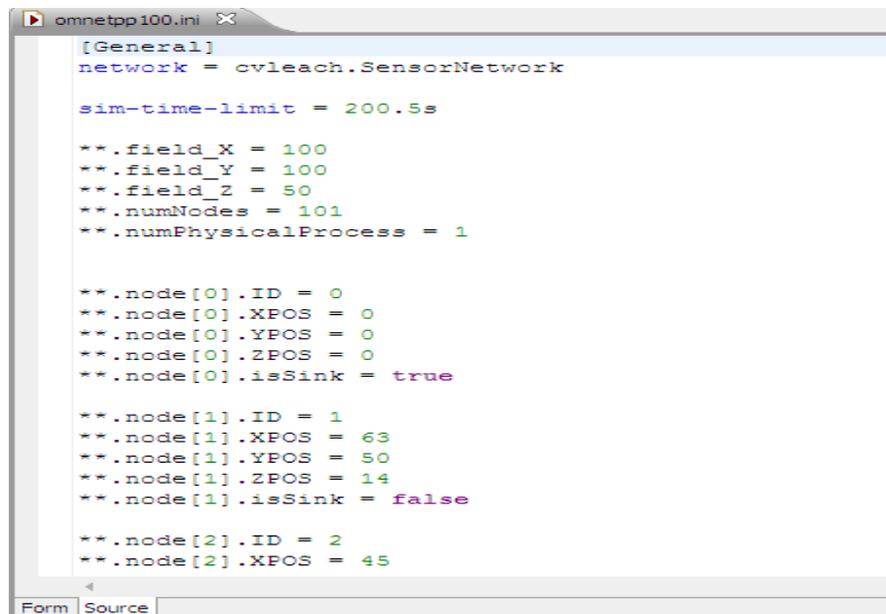


Figure 4.5 : L'éditeur de code source INI.

L'éditeur de texte permet à l'utilisateur de travailler avec le fichier INI directement, ce qui est plus efficace pour les utilisateurs avancés que les formes.

4.3.1. Principe de fonctionnement d'OMNeT++

Le fonctionnement d'OMNeT++ repose entièrement sur l'utilisation de modules qui communiquent entre eux par le biais de messages. Ces modules sont organisés hiérarchiquement. Les modules de base sont appelés les modules simples. Ceux-ci sont regroupés en modules composés. Ces modules composés peuvent eux-mêmes être regroupés en modules composés. Le nombre de niveau hiérarchique n'est pas limité.

Les modules simples sont programmés en C++, ces fichiers fournissent le comportement désiré du module.

L'utilisateur implémente ce comportement en C++ et en utilisant la bibliothèque de classes d'OMNeT++ à travers la redéfinition des trois méthodes principales:

- **Initialize()**: exécutée au démarrage de la simulation.
- **handleMessage()**: exécutée à la réception d'un message.
- **Finish()**: exécutée à la fin de la simulation.

L'utilisateur peut ajouter ses propres méthodes. Tandis que les modules composés constitués d'un ou de plusieurs modules simples, sont programmés dans le langage de haut niveau NED, ce qui rend sa configuration pour les différentes simulations très facile. Ces modules peuvent être combinés de différentes manières.

Pour la communication entre les nœuds, on échange des messages. OMNeT++ donne aux utilisateurs la possibilité d'étudier l'architecture du nœud, l'efficacité énergétique, l'architecture de communications, l'architecture du système, les protocoles, etc.

Le facteur déterminant le temps d'exécution de la simulation est le langage de programmation utilisé. Les simulations sous OMNeT++ peuvent être programmées en C ou en C++. Ce sont ces langages qui fournissent les meilleurs temps d'exécution.

4.3.2 L'architecture d'OMNeT

L'architecture d'OMNeT est construite de telle sorte que les modules simples sont à la fois les émetteurs et destinataires des messages. Les modules composés se contentent de relayer

les messages aux modules simples de façon transparente. On peut attribuer différent paramètre aux connexions reliant les modules: des délais de propagation, des débits de données, des taux d'erreur, etc.

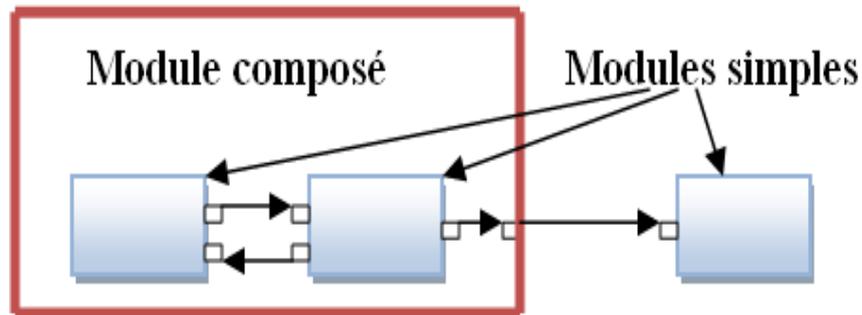


Figure 4.6: L'architecture en modules de OMNeT

4.4 Les différents modèles implémentés

Notre réseau est considéré comme un module composé de plusieurs autres modules.

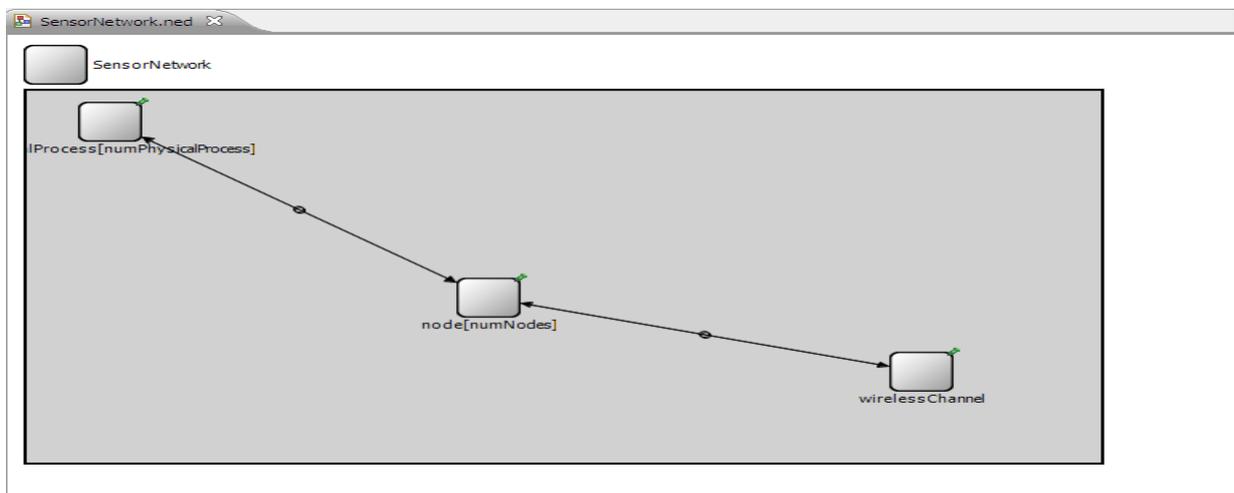


Figure 4.7: La structure de notre réseau.

4.4.1 Le module simple « Physical Process »

Un seul modèle simulant la capture d'un processus physique a été implémenté. Il s'agit de la température. A chaque requête parvenant de l'unité de capture -Sensing ce module répond en

envoyant une valeur aléatoire. Vu que le but initial de la plateforme était le test et la comparaison des protocoles de communication, nous ne nous sommes pas trop focalisés sur les modèles simulant les processus physiques ou les objets cibles.

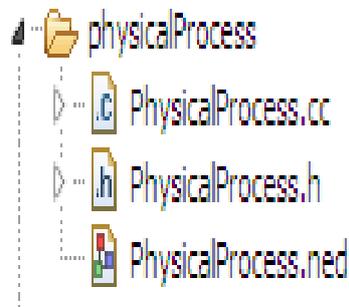


Figure 4.8: L'arborescence de module PhysicalProcess.

Remarque :

1. Il est évident qu'on peut avoir plusieurs processus physiques et/ou objets cibles à contrôler dans un même réseau. Chacun d'eux sera représenté par un module.
2. Tous les nœuds capteurs sont reliés avec le module qui représente le processus physique. Comme nous allons présenter, la communication avec ces modules se fait à travers un module modélisant l'unité de capture interne au nœud.

4.4.2 Le module simple « Wireless Chanel »

Représente le canal de transmission sans fil.

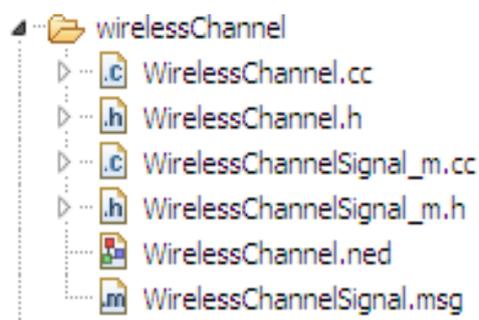


Figure 4.9: L'arborescence de module WirelessChannel.

Ce module doit donc gérer la connectivité entre les nœuds. En se basant sur la puissance du signal émis, les distances entre les nœuds, il décidera des nœuds qui devront recevoir le signal et

la puissance du signal à la réception. Tous les nœuds capteurs ainsi que la station de base doivent donc avoir des liens des communications par messages avec ce module. Au niveau de ce module, un des modèles de propagation les plus connus a été repris de Castalia est ré-implémenté. Ce modèle, présenté au niveau de (Lahcene DEHNI, 1998) consiste à estimer la puissance du signal reçu par chaque nœud en fonction de la puissance d'émission et de la distance entre l'émetteur et le récepteur en question. Si la puissance dépasse un certain seuil, le message est considéré reçu. Le modèle proposé estime la puissance du signal à la réception selon la formule:

$$P_{r,d} = P_t - 10 \log_{10} \left(\frac{d}{d_0} \right)^{\alpha} - N_{r,d}$$

Avec

- $P_{r,d}$: La puissance du signal à la réception en dB.
- P_t : La puissance de transmission.
- $10 \log_{10} \left(\frac{d}{d_0} \right)^{\alpha}$: path-loss, la perte en dB pour une distance d de l'émetteur.
- d_0 : distance de référence
- α : path-loss exponent, indique le taux avec lequel le signal s'atténue en fonction de la distance.
- $N_{r,d}$: est une variable aléatoire suivant une loi normale centrée avec un écart-type σ .

4.4.3. Le module composé « Node » :

Les nœuds capteurs ainsi que la station de base sont représentés par des modules composés.

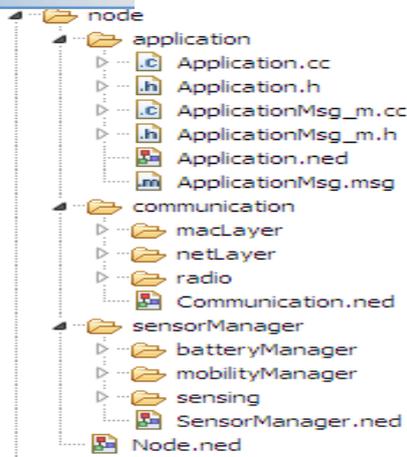


Figure 4.10 : L'arborescence de module Node.

4.4.3.1 L'architecture d'un nœud capteur au niveau de notre plateforme

Chaque nœud capteur est, lui-même, un module composé. Dans la figure, nous proposons une architecture interne pour chaque nœud capteur. Les flèches représentent les communications.

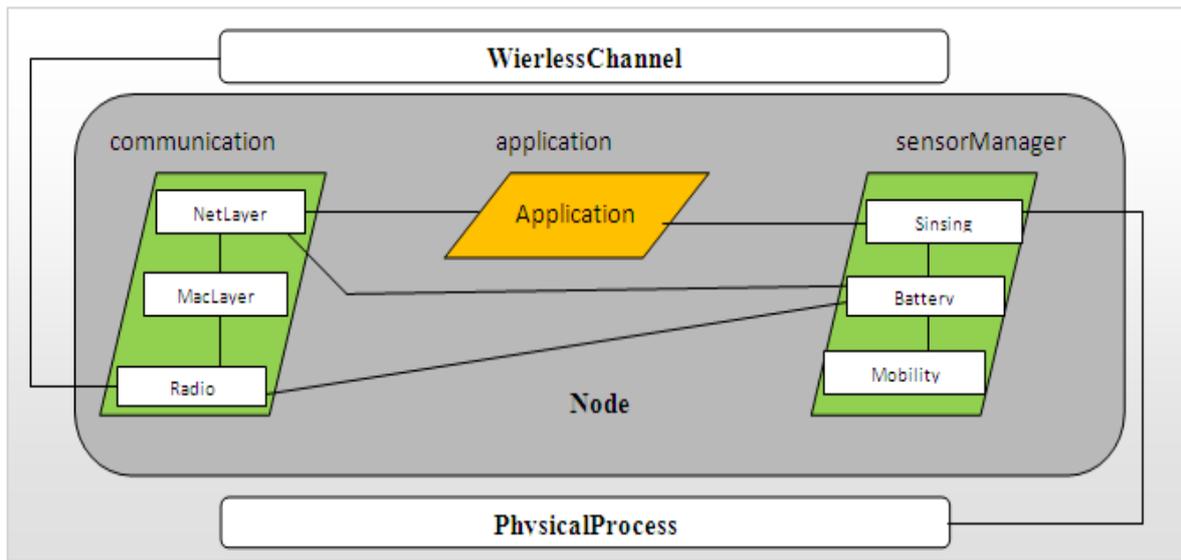


Figure 4.11: Architecture interne d'un nœud capteur.

Au niveau de cette architecture, le nœud capteur est considéré comme un module composé dont les sous-modules sont :

Le sous module communication : qui est un module composé, il contient les sous modules NetLayer, MacLayer et Radio.

- **NetLayer :**

Simule les protocoles et les algorithmes qui se déroulent au niveau de la couche réseau d'un nœud capteur indépendamment des autres modules. Il permet donc d'implémenter différentes approches de routage. Ce module a des liens de communication bidirectionnels avec les deux modules adjacents dans la pile protocole (la couche application et la couche MAC).

- **MacLayer :**

Simule les protocoles et les techniques utilisées au niveau de la couche MAC. Ce module doit avoir une connexion bidirectionnelle avec le module simulant la couche réseau et le module Radio du nœud. Au niveau de la couche MAC, un modèle simple a été implémenté. On a considéré que cette couche fait juste le relais entre les autres modules.

- **Radio :**

Va simuler à la fois la couche physique d'un capteur et l'unité de transmission/réception dans un capteur. Les différents états de la radio sont : mode veille, écoute, transmission/ réception.

Les niveaux de transmission, la puissance de transmission ainsi que les délais de transmissions sont aussi pris en compte au niveau de ce module. La consommation énergétique dans les différents états de la radio est gérée au niveau de ce module. Il est également chargé de la modulation et de la démodulation des données (transformation des données sous forme de signaux et inversement).

Le module -Radio -doit avoir une connexion bidirectionnelle avec le module -Wireless Channel- afin de pouvoir simuler l'envoi et la réception des données. De plus, une connexion bidirectionnelle avec la couche Mac est obligatoire. Afin de mettre à jour l'énergie restante au niveau du nœud d'une façon continue, ce module a une liaison directe avec le module -Battery- du capteur.

Le sous-module application : c'est un module simple.

- **Application** :

Simule les applications utilisateurs qui s'exécutent au niveau des capteurs. Au niveau de ces applications, il est possible de recevoir les données capturées par le module "Sensing " et les envoyer à travers la pile protocolaire après d'éventuels traitements. On peut aussi envoyer des requêtes demandant des données et/ou recevoir des informations parvenant à d'autres nòuds.

De plus, l'utilisateur peut utiliser des messages prédéfinis pour contrôler et commander les modules Radio et MAC du nòud (changer l'état, modifier les paramètres, etc.)

Afin de garantir ces fonctionnalités, le module *Application Layer* doit avoir des liens de communication bidirectionnelle avec le module simulant l'unité de capture ainsi que celui simulant la couche réseau.

Le sous module sensorManager : c'est un module composé, il contient les sous-modules Sinsing, Battery

- **Sensing** :

Modélise l'unité de capture au niveau d'un nòud. Il est responsable de la réception des événements générés par le processus physique. Lorsque l'application exécutée au niveau d'un nòud capteur requiert une donnée, un message est envoyé au module "Sensing " Ce dernier envoie un message au module "PhysicalProcess " qui répond en envoyant une valeur qui sera transmise. Le module sensing envoie une requête au processus physique à chaque fois que l'application demande une donnée. A la réception de la valeur requise, l'unité en question (sensing) envoie un message au module "Application" contenant l'information captée et met à jour l'énergie restante au niveau du nòud.

- **Battery** :

Elle est responsable de la mise à jour de l'énergie du nòud capteur. Si l'énergie est totalement dissipée, il envoie un message de déconnexion aux autres modules du nòud capteur.

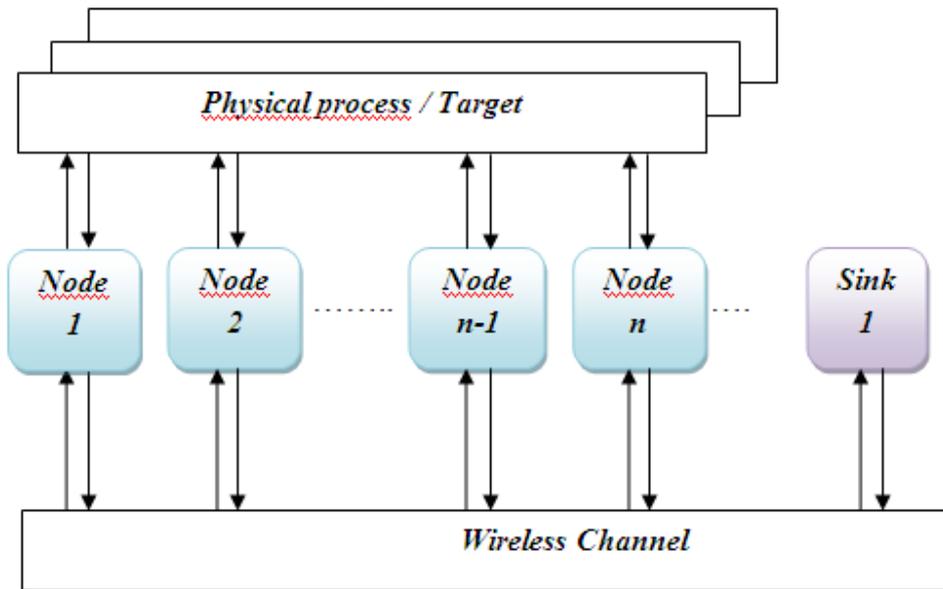


Figure 4.12: L'architecture de notre plateforme (les modules et leurs connexions).

4.4.4 Le modèle énergétique

Contrairement à la phase de traitement des données, la phase de communication consomme plus d'énergie. Dans ce travail, nous mettons en œuvre un modèle énergétique qui couvre à la fois, les communications en émission et en réception. Ainsi, pour transmettre un message codé sur k bits sur une distance d , l'émetteur consomme :

$$P_{Tx} = \frac{k}{\eta} \left(\frac{1}{\rho} + \rho \right) \frac{1}{d^\alpha} \quad (4.1)$$

Pour ne pas passer par le calcul des distances qui est un peu délicat vu le déploiement aléatoire des nœuds, nous avons décidé de prendre la valeur de " ρ " égal à la puissance de transmission que nous avons calculé au préalable et qui est P_{Tx} .

Et vu que P_{Tx} est en dBm nous devrions passer par sa transformation en watt qui se fait de la manière suivante:

$$P_{Tx}(\text{W}) = 10^{\frac{P_{Tx}(\text{dBm}) - 30}{10}}$$

Tel que X est la valeur en dBm .

Et en fin nous avons procédé à appliquer la formule :

Pour obtenir l'énergie consommée à l'émission en sachant que le temps est calculé par :

$$E_{\text{émission}} = \frac{P_{\text{émission}} \times T_{\text{émission}}}{\eta_{\text{émission}}}$$

L'énergie consommée à la réception est calculée comme suit :

$$E_{\text{réception}} = P_{\text{réception}} \times T_{\text{réception}} + E_{\text{circuit}} + E_{\text{antenne}}$$

4.4.6 Implémentation des différentes fonctionnalités

- **Procédure de formation de cluster :**

La formation de cluster est le point de différence principale entre LEACH, CVLEACH et DSLEACH, tel qu'en LEACH et CVLEACH les clusters sont formés en utilisant la formule :

$chThresholdElection = \frac{percentageOfCh}{(1 - percentageOfCh * (r \% cycle))}$; décrite dans le chapitre précédent, à la différence qu'en CVLEACH si un nœud a déjà été CH lors du dernier round ou qu'il a reçu un message d'avertissement de son voisinage il retire sa participation pour être CH au cours de ce round

Par contre, en DSLEACH la formation de clusters se réalise en utilisant la formule :

$chThresholdElection = \frac{1}{(density - (r \% density)) - (numCh/r)}$; décrite aussi précédemment.

- **Procédure d'agrégation et de transmission des données :**

La transmission des données est identique en LEACH et DSLEACH mais un peu différente en CVLEACH tel qu'en ce dernier si un nœud n'est ni CH ni membre de cluster, il envoie ses données collectées directement vers la station de base.

4.5 Simulation

4.5.1 Les métriques d'évaluation des performances

Pour quantifier les performances des protocoles LEACH, DSLEACH et CVLEACH, les métriques sélectionnées sont:

La moyenne d'énergie consommée, et son écart type.

La durée de vie de réseau.

Ces métriques sont affectées par des facteurs comme le nombre de nœuds utilisés dans les scénarios de simulation, et le temps de simulation.

Les résultats sont obtenus par l'exécution de plusieurs simulations indépendantes pour chaque scénario.

Moyenne d'énergie consommée (MEC)

La faible autonomie en énergie est le point le plus crucial dans les RCSF. Ce qui fait que l'un des principaux critères de performances pour un capteur est la durée d'utilisation efficace de son énergie embarquée avant de l'épuiser. Parce qu'une fois qu'elle sera épuisée, il n'y a pas d'autre moyen de recharger à nouveau sa batterie, s'il a été déployé à des endroits inaccessibles et n'utilisant pas de cellules solaires.

Ainsi, l'Énergie Consommée (MEC) mesure en moyenne la quantité d'énergie consommée par les capteurs, elle est obtenue par la formule mathématique suivante :

$$MEC = \frac{1}{n} \sum_{i=1}^n (E_i - E_{resid})$$

Au :

E_i : l'énergie consommée par un capteur i . Elle présente la différence entre l'énergie initiale et son énergie résiduelle.

n : est le nombre de capteurs dans le réseau.

Ecart type (ET)

En mathématiques, l'écart type est une quantité réelle positive, éventuellement infinie, utilisée dans le domaine des probabilités pour caractériser la répartition d'une variable aléatoire autour de sa moyenne.

Dans notre cas, on l'a calculé de la manière suivante :

$$ET = \sqrt{\frac{1}{n} \sum_{i=1}^n (E_i - MEC)^2}$$

Au :

MEC : Moyenne d'énergie consommée.

E_i : l'énergie de n^o uds i.

n : nombre de n^o uds de réseau.

Durée de vie du réseau

Afin de mesurer la durée de vie du réseau, nous avons fait le suivi d'évolution du nombre de n^o uds en vie au fil du temps. Cette métrique nous permet de voir la fréquence de la mort des n^o uds dans le réseau .Elle peut être utilisée pour déterminer la durée de vie du réseau selon le temps qui s'écoule avant que le dernier n^o ud en vie dans le réseau épuise son énergie

4.5.2 Paramètres de simulation

Pour que l'évaluation des performances des trois protocoles par simulation soit efficace, il fallait qu'on prenne en considération les spécificités des réseaux de capteurs pour qu'elle soit réalisable dans des conditions qui se rapprochent de la réalité. Pour cela, certains paramètres ont été pris en compte lors de la simulation. Ces paramètres sont sélectionnés en se basant sur des travaux antérieurs pour des applications similaires ainsi que sur la capacité du simulateur. Le tableau ci-dessous résume quelques paramètres utilisés :

Paramètre	Valeur
Nombre de n ^o uds du réseau	50, 75, 100, 125, 150, 175,200
Durée de simulation (secondes)	200
Protocoles de routages	LEACH, DSLEACH et CVLEACH
Energie initiale	1000 milli-Joules (mJ)
Simulateur	OMNET ++
Surface de simulation (m m m)	100 * 100* 50
Placement des n ^o uds	Aléatoire
Nombre de stations de base	1
Durée d'un round	10s

Table 2: Paramètres de simulation

4.6 Discussion des résultats

Dans ce qui suit, nous allons présenter et analyser les résultats de simulation obtenus suivant les métriques de performances discutées précédemment. Les simulations sont effectuées sur des capteurs déployés de manière aléatoire. Dans tous les scénarios de simulations générées, les trois protocoles, LEACH, DSLEACH et CVLEACH sont comparés, pour différencier leurs performances. Afin d'examiner l'influence de la densité du réseau sur les performances des trois protocoles, nous avons mené des simulations en variant le nombre de nœuds déployés dans le réseau (de 25 à 200 nœuds par un pas de 25 nœuds) augmentant ainsi la densité du réseau.

4.6.1 Moyenne d'énergie consommée (MEC)

Le graphe sur la figure 4.13 présente la moyenne d'énergie consommée en fonction de nombre de nœuds de réseau.

On peut voir sur le graphe que le protocole DSLEACH consomme moins d'énergie par rapport à CVLEACH qui à son tour consomme moins que son ancêtre LEACH.

On peut donc dire que le protocole DSLEACH est plus performant que le protocole CVLEACH qui est plus performant que LEACH, en marquant moins de moyenne d'énergie consommée.

Ce gain en énergie en DSLEACH revient à la probabilité qu'il utilise ce dernier afin de construire des clusters dans les zones les moins denses.

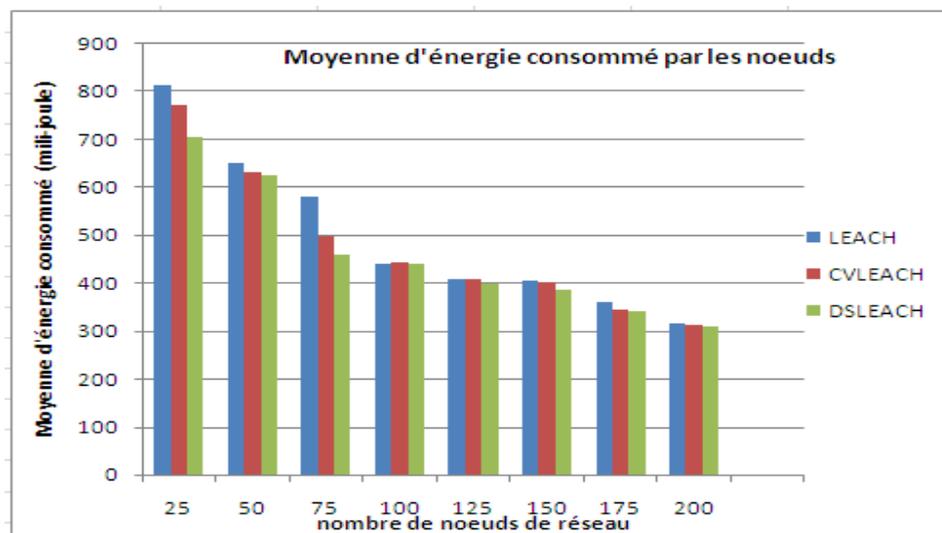


Figure 4.13 : Moyenne d'énergie consommée par les nœuds

Nous remarquons aussi que l'accroissement du nombre de nœuds déployés dans le réseau induit à une diminution dans la consommation d'énergie dans les trois protocoles. Ceci est dû à la diminution des distances de transmission en augmentant le nombre de nœuds de réseau.

4.6.2 Ecart type d'énergie consommée

Le graphe 4.14 présente l'écart type d'énergie consommée en fonction de nombre de nœuds de réseau.

L'écart type de l'énergie consommée par LEACH est toujours supérieur à celui de CVLEACH et DSLEACH. Tandis que celui de DSLEACH est la plus petite, ce qui montre que la répartition de la consommation énergétique est plus équilibrer en DSLEACH.

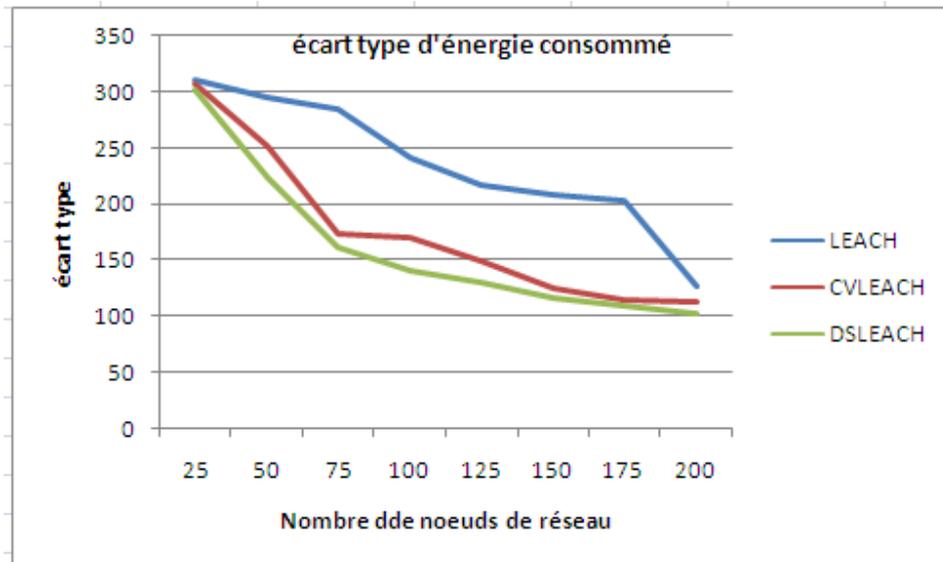


Figure 4.14 : l'écart type de l'énergie consommée .

On remarque aussi que l'écart type de l'énergie consommée diminue en fonction de nombres de nœuds, de réseau.

Ces résultats sont évidents de moment que l'énergie consommée varie en fonction du nombre de nœuds dans le réseau et la moyenne d'énergie consommée par DSLEACH est la moins grande.

4.6.3 Durée de vie du réseau

Afin d'étudier la durée de vie du réseau en fonction du nombre de nœuds déployés, nous avons suivi l'évolution de 50 et 100 nœuds dans le temps (de 100 à 1000 secondes).

Les graphes suivant présentent la durer de vie de réseau en fonction de temps écoulé.

On remarque sur les graphes que la durée de vie de réseau diminue en fonction de temps d'exécution.

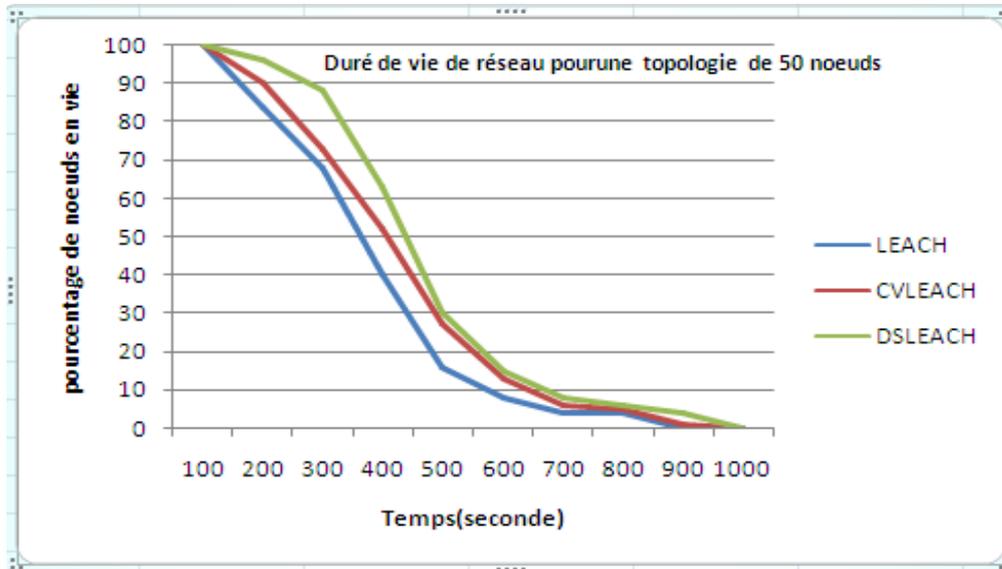


Figure 4 .15 : La durée de vie de réseau pour 50 nò uds.

Il est clair que la durée de vie du réseau offerte par le protocole DSLEACH est plus importante que celle de protocole CVLEACH et LEACH, et ce dans les deux configurations testées. Ce gain revient aussi à l'équilibrage dans la distribution de la charge qu'a présenté DSLEACH

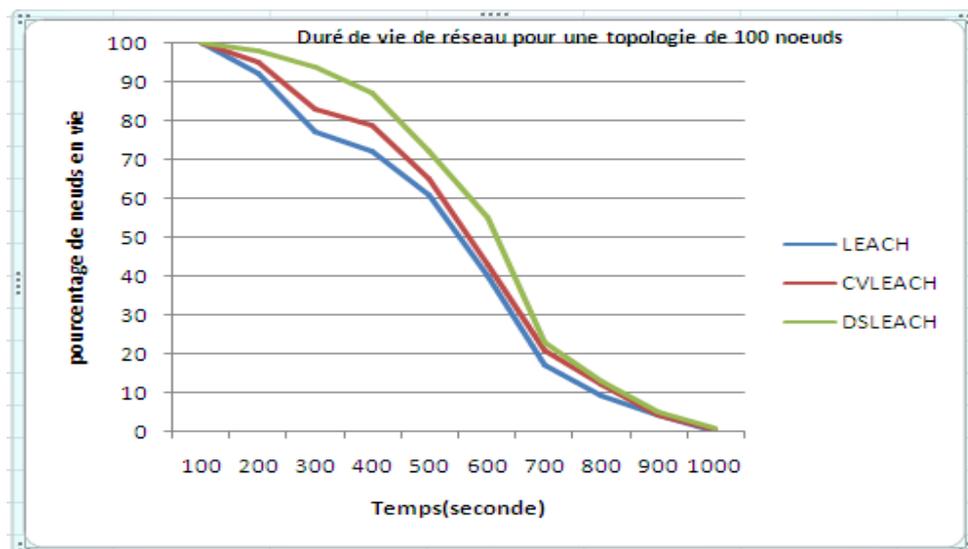


Figure 4 .16 : La durée de vie de réseau pour 100 nò uds

On peut également constater que la durée de vie du réseau augmente avec la densification du réseau ce qui est attendu du moment que l'énergie moyenne consommée diminue en augmentant le nombre de nœuds de réseau.

Ces résultats s'expliquent par la réduction des distances de transmission entre les nœuds voisins avec l'augmentation du nombre de nœuds déployés dans le réseau.

4.7 Conclusion

Nous avons présenté dans ce chapitre les résultats de simulation des trois protocoles de routages LEACH, CVLEACH et DSLEACH. L'étude comparative menée par simulation entre les performances de ces protocoles a été réalisée sur les performances du réseau dans la présence d'un nombre variant des nœuds. Les performances du réseau dans ce test de simulation a été évaluée en fonction de l'énergie consommée. Les tests ont montrés que le protocole DSLEACH est plus performant en terme d'énergie consommé et la durée de vie de réseau par rapport a CVLEACH, sa s'explique par la politique de clustering qu'utilise DSLEACH et qui permet une meilleure organisation des groupes.

Conclusion générale :

La conception des réseaux de capteurs est fortement influencée par la limitation de la ressource énergétique disponible au niveau des nœuds capteurs. Actuellement, la plupart des travaux de recherche sur ce type de réseaux, sont consacrés à la conception des protocoles de routage visant à minimiser l'énergie inhérente aux communications qui sont la source principale de consommation d'énergie afin d'optimiser la durée de fonctionnement du réseau.

Dans cette optique, le routage hiérarchique est présenté comme étant une solution prometteuse pour conserver l'énergie des nœuds, et faciliter la transmission des données capturées dans le réseau vers la station de base. Selon ce type de routage, les nœuds du réseau sont organisés en clusters gérés par un seul nœud (cluster-head).

L'étude qui a fait l'objet de ce mémoire nous a permis d'approfondir nos connaissances dans le domaine des réseaux de capteurs qui est une des technologies clefs de l'avenir et ce en raison de l'incroyable potentiel applicatif qu'elle renferme. Les algorithmes et les protocoles de communication pour les réseaux de capteurs doivent minimiser la consommation d'énergie qui reste très variable selon les protocoles utilisés.

Ce mémoire a débuté par une étude générale sur les réseaux de capteurs sans fil pour passé en second lieu au routage dans les RCSF et ses différentes classifications, nous avons fait par la suite une étude détaillée sur le fonctionnement des trois protocoles LEACH, CVLEACH et DSLEACH pour terminé par des testes et comparaison des résultats.

L'objectif fixé était de comparer entre le protocole CVLEACH et le protocole DSLEACH qui sont deux variantes de LEACH ayons pour bute la réduction de la consommation énergétique dans le réseau. Vu que dans ces deux protocoles les chercheurs se sont focalisé sur les nœuds non couvert (ils n'appartiennent a aucun cluster) par le réseau, en utilisant différentes techniques pour remédier a se problème nous avons décidé de comparais entre ses deux protocoles pour sélectionné la méthode la plus efficace en terme de duré de vie de réseau.

Ces études ont été validées à l'aide du simulateur de réseaux OMNET++. Les résultats des simulations ont bien illustré, pour les trois protocoles leurs billons énergétiques.

Au terme de ce travail on a appris que l'obtention de meilleures performances pour les réseaux de capteurs avec l'approche de clustering résulte d'une organisation efficace de ses capteurs en clusters.

Enfin nous espérons que notre travail sera bénéfique pour les prochains travaux de recherche dans le domaine des réseaux de capteurs sans fil pour les promotions à venir.

Perspectives

Les réseaux de capteurs constituent un domaine de recherche très vaste. Ils ont de nombreuses perspectives d'application dans des domaines très variés. Il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans des conditions réelles.

Afin d'être adaptés à un environnement réel, les protocoles étudiés peuvent être toujours améliorés en introduisant la mobilité des nœuds et en tenant compte de la qualité de service, voir même développer de nouveaux protocoles de routages pour les réseaux de capteurs sans fil.

Enfin, les performances des protocoles étudiés ont été prouvées à l'aide du simulateur OMNET++ et il serait préférable de réaliser ces études dans des conditions réelles de manière à comparer les performances réelles par rapport à celles effectuées par des simulations.

Bibliographie

1. [KO 02] F. Koushanfar, M. P.-V. (June 2002.). Fault Tolerance in Wireless Ad hoc Sensor Networks. *Proceedings of IEEE Sensors 2002* .
2. [BR 08] F.Brissaud, D. e. (2008). capteurs intelligents : nouvelles technologies et nouvelles problématiques pour la sureté de fonctionnement. *Maitrise des risque et de sureté de fonctionnement,Lambda Mu 16,Avignon:France* .
3. [FAH] Fahmy, O. Y. (s.d.). A hybrid, energy-efficient, distributed clustering approche.
- 4.[ZH 04] G. Zhou, T. H. (s.d.). δImpact of Radio Irregularity on WSN. *MobiSys 2004*.

Global Mobile Information Systems Simulation library. (s.d.). <http://pcl.cs.ucla.edu/projects/glomosim/>.
- 5.[O. Y] HEED A hybrid, energy-efficient, distributed clustering approach.
6. [IA 02] Ian F. Akyildiz, W. S. (2002). wireless sensor network.
- 7.[LE 03] J. Lester Hill. (Berkeley, 2003.). δSystem Architecture for Wireless Sensor Networks. *University of California* .
- 7.[JB 09] Jamshid Bagherzadeh, M. S. (2009). A Clustering Algorithm for Wireless Sensor Networks based on Density of Sensors. *Urmia University, Iran* .
- 8.[SHO] J-Sim Home page. [En ligne]: <http://sites.google.com/site/jsimofficial/>. (s.d.).
- 9.[AK 05] K. Akkaya, a. M. (May 2005). A Survey on Routing Protocols for Wireless Sensor Network. *Journal of Ad Hoc Networks, Vol. 3, No. 3* .
- 10.[AK05] K. Akkaya, M. Y. (2005). Routing Techniques in wireless sensor networks: A survey.
- 11.[KA 04] Kamel, J. A. (Dec 2004). Routing technique in WSN. *Magzine IEEE communication*
- 12.[KO 12] Kotecha, P. A. (June 2012). CVLEACH: Coverage based energy efficient LEACH algorithm. *CSE Department, Institute of Technology* .

- 13.[LADH] Lahcene DEHNI *, Y. B. (s.d.). Une nouvelle approche de routage dans les réseaux de capteurs pour l'optimisation de la consommation d'énergie. *Laboratoire d'Informatique de Paris Nord UMR 7030 du CNRS, Université Paris 13* .
- 14.[LE 03] Levis, P. e. (2003). TOSSIM: A Simulator for TinyOS Networks.
- 15.[MY 02] M. Ye, C. L. (2002). EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Network. *National Laboratory of Novel Software Technology, Nanjing University, China* .
- 16.[YOUN] M. Younis, K. e. (s.d.). A survey on routing protocols for wireless sensor networks.
- 17.[NOU00] N. Noury, T. H. (October 2000.). Monitoring behavior in home using a smart fall sensor. *IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology* .
- 18.[PC 04] P, C. C. (2004.). Sensor networks for monitoring traffic, 42nd Allerton. *Conference on Communication Control and Computing* .
- 19.[QF 03] Q. Fang, F. Z. (June 2003). Lightweight Sensing and Communication Protocols for Target Enumeration and Aggregation. *4th ACM International Symposium on Mobile Ad hoc Networking and Computing (MOBIHOC 2003)*.
- 20.[RJ 07] R. Jurdak. (2007). Wireless Ad-Hoc and Sensor network, Cross layer design perspective. *University College Dublin* .
- 21.[RA 07] Raghavendra, S. L. (September 2002). Power-Efficient Gathering in Sensor Information. *Systems. IEEE Transaction on Parallel and Distributed Systems, vol.13, no.9* .
- 22.[SAN05] Santi, P. (2005). Topology Control in Wireless Ad Hoc and Sensor Networks. *Hardcover* .sensorsim. (s.d.). <http://sensorsim.org>.
- 23.[VH 03] V. Handziski, A. K. (July 2003.). A common wireless sensor network architecture. *Technische Universität Berlin* .
- 24.[WR 00] W.R. Heinzelman, A. C. (2000). Energy-efficient communication protocol for wireless micro sensor networks. *IEEE Hawaii International Conference on System Sciences* .
Wendi Rabiner Heinzelman, A. C. (s.d.). Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *Massachusetts Institute of Technology Cambridge, MA 02139* .