

République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Mouloud MAMMERI de TIZI-OUZOU
Faculté de Génie Electrique et Informatique
Département : Informatique



Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique
Option : Système Informatique.

Thème :

**Optimisation d'énergie dans les réseaux
de capteurs sans fil.**

Dirigé par : M^{er} A.DIB.

Réalisé par :

OURAD karima.

ZEGROUR Zahia.

Promotion : 2013 - 2014

Remerciements

Nous remercions Dieu le tout puissant de nous avoir donné la volonté et le courage pour accomplir ce modeste travail.

Nous remercions notre promoteur M^{er} A. DIB pour avoir accepté de nous encadrer et pour son soutien et ses conseils qui nous ont permis de mener à bien ce projet.

Que les membres du jury trouvent ici nos remerciements les plus vifs pour avoir accepté d'honorer par leurs jugements de notre travail.

Nos sincères sentiments vont à tous ceux qui de près ou de loin ont contribué à la réalisation de ce projet, à nos amis (es) particulièrement Badia et Nacera, à nos chères familles.



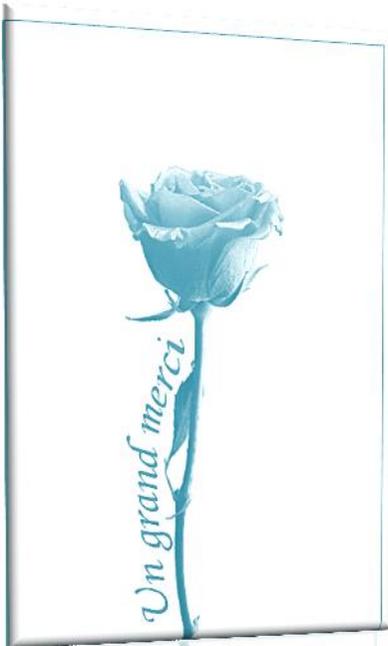
Zahia et Karima.

Dédicaces

Je dédie ce modeste travail :

A ceux qui m'ont tout donné sans rien en retour.

A ceux qui m'ont encouragé et soutenu dans mes moments les plus durs.



A mes parents pour leur amour et leur support affectif.

A mon frère et mes sœurs.

A mes neveux et ma nièce.

A mon cher fiancé et sa famille.

A mes proches et toute la famille Ourad.

A mes chers amis (es).

A mon binôme Zahia et sa famille.

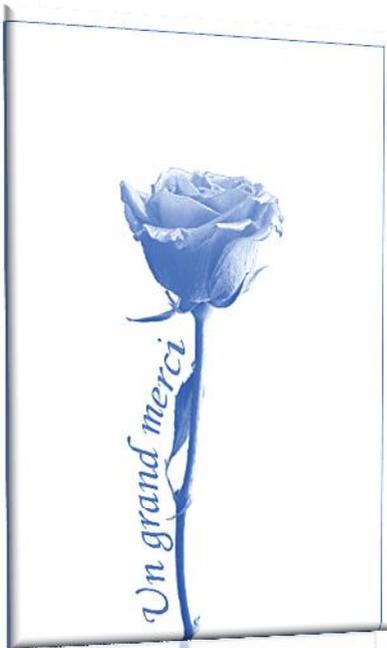
Karima

Dédicaces

Je dédie ce modeste travail :

A ceux qui m'ont tout donné sans rien en retour.

A ceux qui m'ont encouragé et soutenu dans mes moments les plus durs.



A mes chers parents.

A mon cher frère Nadir.

A mes chères sœurs Nadia et Lila.

A toute ma famille.

A nounou.

A tous mes amis (es).

A mes amis de la promo.

A mon binôme Karima et sa famille.

A tous ceux qui m'aiment et que j'aime.

Zahia.

**« Celui qui trouve sans chercher est celui qui a longtemps
cherché sans trouver. »**

Gaston Bachelard (1884-1962)

Résumé:

Un réseau de capteur est composé d'un grand nombre de capteurs déployés dans des régions à surveiller, et communiquant entre eux à travers un médium sans fil. Le routage de données collectées dans le réseau consomme la plus grande partie d'énergie des nœuds capteurs. A cet effet, plusieurs approches de routage ont été proposées pour conserver la ressource énergétique au niveau des capteurs ainsi prolonger la durée de vie du réseau. Dans ce travail nous proposons d'adapter un protocole existant au routage multi-saut. Pour améliorer d'avantage la consommation d'énergie, nous avons choisis le protocole Leach qui est l'un des algorithmes de routage hiérarchiques le plus populaires pour les réseaux de capteurs sans fil, dans ce protocole les nœuds doivent acheminer leurs données à la station de base tout en minimisant la consommation d'énergie. Cette tâche devient plus ardue particulièrement quand le nombre de nœuds déployés augmente. Notre solution vise à exploiter plus équitablement l'énergie des nœuds sélectionnés cluster-Head, et à économiser l'énergie dissipée lors de l'acheminement des données capturées à la station de base en utilisant une communication multi-saut. Notre solution est évaluée avec un simulateur à événement discret OMNeT++ que nous avons choisis d'utiliser pour son architecture modulaire. Une analyse a été faite pour interpréter les résultats obtenus. Ces résultats ont montré que notre protocole permet une réduction de la dissipation d'énergie et une durée de vie du réseau plus grande.

Mots clés: Réseaux de capteurs sans fil, conservation d'énergie, routage, le protocole LEACH, clustering, OMNeT++.

Liste des acronymes:

RCSF	Réseau de Capteurs Sans Fil.
WSN	Wireless Sensor Networks.
ADV	Adversting.
REQ	Request.
CHs	Cluster Heads.
CSMA	Carrier Sense Multiple Access.
CDMA	CODE Division Multiplexed Access.
IEEE	Institute of Electrical and Electronics Engineers.
DVS	Dynamique Voltage Scaling
J-Sim	Java-SIMultaor.
LEACH	Low Energy Adaptative Clustering Hierarchy.
PEGASIS	Power-Efficient Gathering in Sensor Information Systems.
MAC	Media Access Control.
MANET	Mobile Ad hoc Network.
OMNET++	Objective Modular NETwork Test-bet in C++.
OPNET	Optimum Network Performance.
OSI	Open Système Interconnexion.
SB	Station de Base.
TDMA	Time Division Multiplexed Access.
CSMA	Carrier Sense Multiplexed Access.
Wi-Fi	Wireless-Fidelity.
SPIN	Sensor Protocols for Information via Negotiation.

DD	Directed Diffusion.
TEEN	Threshod sensitive Energy Efficient Network protocol.
APTEEN:	Adaptive Threshod sensitive Energy Efficient Network protocol.
MCR	Multi-hop Clustering Routing Protocol.
MTE	Routage multi-sauts.
HEED	Hybrid, Energy-Efficient, Distributed approach.
CSIP	Collaborative Signal and Information Processing.

Liste des figures :

- Fig. I.1. Architecture d'un nœud de capteurs.....** Erreur ! Signet non défini.
- Fig. I.2. Architecture d'un réseau de capteurs.** Erreur ! Signet non défini.
- Fig. I.3. La pile protocolaire d'un réseau de capteur.....**Erreur ! Signet non défini.
- Fig. I.4.Capteurs dans un corps humain.** Erreur ! Signet non défini.
- Fig. I.5. Application environnementale.** Erreur ! Signet non défini.
- Fig. I.6. Les techniques de conservation d'énergie..** Erreur ! Signet non défini.
- Fig.II.1.Topologie plate.** Erreur ! Signet non défini.
- FI.II.2.Topologie hiérarchique.** Erreur ! Signet non défini.
- Fig.II.3. Fonctionnement du protocole SPIN** Erreur ! Signet non défini.
- Fig.II.4.Fonctionnement du protocole DD** Erreur ! Signet non défini.
- Fig.II.5. Organigramme de la phase Setup du protocole leach.** Erreur ! Signet non défini.
- Fig.II.6. Organigramme de la phase steady du protocole leach.....** Erreur ! Signet non défini.
- Fig.II.7.Fonctionnement du protocole TEEN** Erreur ! Signet non défini.
- Fig.III.1.Exemple du protocole MCR.....**Erreur ! Signet non défini.
- Fig.IV.1. Architecture du réseau selon le modèle de simulation implémenté.....**Erreur ! Signet non défini.
- Fig.IV.2. la couche réseau ou sera implémenté notre protocole.....** Erreur ! Signet non défini.
- Fig.IV.3. Nouveau projet sous OMNET.....** Erreur ! Signet non défini.

Fig.IV.4. Nouveau projet nommé Leach sous OMNET.....Erreur ! Signet non défini.

Fig.IV.5.le protocole Leach sous OMNET. Erreur ! Signet non défini.

Fig.IV.6.Moyenne d'énergie consommée..... Erreur ! Signet non défini.

Fig.IV.7. Durée de vie du réseau. Erreur ! Signet non défini.

Fig.IV.8. Ecart type de l'énergie consommée..... Erreur ! Signet non défini.

Liste des tableaux :

Tab. II.1. Comparaison entre ces différents protocoles.Erreur ! Signet non défini.

Tab.IV.1. messages échangé dans LEACH.Erreur ! Signet non défini.

Tab.IV.2.les paramètres et valeurs utilisés sous OMNET.Erreur ! Signet non défini.

Tab. IV.3.Moyenne d'énergie consommée.....Erreur ! Signet non défini.

Tab. IV.4. Ecart type de l'énergie consommée.Erreur ! Signet non défini.

Sommaire

Introduction générale

Chapitre I : Les réseaux de capteurs sans fil

I.1.Introduction:	1
I.2.Architecture d'un nœud capteur :	1
I.3. Architecture de communication d'un RCSF :	2
I.4.Pile protocolaire:	4
I.5.Standards de communication dans les réseaux de capteurs sans fils :	6
I.6.Applications des RCSF :	6
I.7. Facteurs et contraintes des RCSF :	8
I.8.Consommation d'énergie dans les RCSF :	10
I.9.Facteurs intervenants dans la consommation d'énergie :	10
I.10.Techniques de minimisation de la consommation d'énergie :	12
I.11.Conclusion:	15

Chapitre II : Les protocoles de routage dans les RCSF_S

II.1.Introduction:	Erreur ! Signet non défini.
II.2.Routage dans les réseaux de capteurs:	Erreur ! Signet non défini.
II.2.1.Facteurs pour le développement de protocole de routage:	Erreur ! Signet non défini.
II.3.Classification des protocoles de routage:	Erreur ! Signet non défini.
II.3.1.Classes principales des protocoles de routage selon la topologie:	Erreur ! Signet non défini.
II.3.2.sous-classes des protocoles de routages:	Erreur ! Signet non défini.
II.4.Exemples des protocoles de routages dans les RCSF:	Erreur ! Signet non défini.
II.4.1.Exemples de protocoles de routage plat:	Erreur ! Signet non défini.

II.4.1.1.

SPIN:.....Erreur !
Signet non défini.

II.4.1.2. Diffusion dirigé

(DD):.....Erreur ! Signet non défini.

II.4.2. Exemples de Protocoles

hiérarchiques:.....Erreur ! Signet non défini.

II.4.2.2. PEGASIS (Power-Efficient Gathering in Sensor Information

Systems):.....Erreur ! Signet non défini.

II.4.2.3. HEED (Hybrid, Energy-Efficient, Distributed

approach):.....Erreur ! Signet non défini.

II.4.2.4. TEEN (Threshold-sensitive Energy Efficient sensor Network

protocol):.....Erreur ! Signet non défini.

II.4.2.5. APTEEN (Adaptive Threshold-sensitive Energy Efficient sensor Network

protocol):.....Er
reur ! Signet non défini.

II.4.2.6.

SPEED:.....Erreur !
Signet non défini.

II.5. Comparaison entre ces différents protocoles

:.....Erreur ! Signet non défini.

II.6. ConclusionEr

reur ! Signet non défini.

Chapitre III : Etat de l'art des protocoles à multi-sauts dans les RCSF_s

III.1. Introduction

.....Erreur ! Signet non
défini.

III.2. Protocoles de routage à

multi-saut:.....Erreur ! Signet non défini.

III.2.1. Le protocol MCR (Mult-hop Clustering Routing

Protocol):.....Erreur ! Signet non défini.

III.2.2. Protocole

LCH......Erreur ! Signet non
défini.

III.2.3. Le routage multi-sauts MTE (Minimum Transmission Energy)Erreur ! Signet non défini.

III.3.Synthèse des protocoles a

multi-sauts :.....Erreur ! Signet non défini.

III.4.Conclusion

:.....Erreur ! Signet non défini.

Chapitre IV : Etude de la solution proposée

IV.1.

Introduction:.....Erreur ! Signet non défini.

IV.2. Problématique et motivations

:.....Erreur ! Signet non défini.

IV.3. Notre

objectif :.....Erreur ! Signet non défini.

IV.4. Etude de la solution

proposée:.....Erreur ! Signet non défini.

IV.4.1.Choix de simulateur:.....Erreur ! Signet non défini.

4.1.1. La Simulation à événements discrets:Erreur ! Signet non défini.

1. TOSSIM.....Erreur ! Signet non défini.

2. PowerTOSSIM :.....Erreur ! Signet non défini.

3. AvroraErreur ! Signet non défini.

4. WSNetErreur ! Signet non défini.

5. J-SimErreur ! Signet non défini.

6. OPNETErreur ! Signet non défini.

7. OMNETErreur ! Signet non défini.

IV.5. Implémentation et simulation sous

OMNET:.....Erreur ! Signet non défini.

IV.5.1. Paramètres de la simulation:Erreur ! Signet non défini.

IV.5.1.1. modèle de simulation implémenté:Erreur ! Signet non défini.

IV.5.1.2. Type de messages échangés:Erreur ! Signet non défini.

IV.5.1.3.Scénario utilisé:.....Erreur ! Signet non défini.

IV.6.

Résultats:.....Erreur !

Signet non défini.

IV.6.1. Critères de performances:.....Erreur ! Signet non défini.

IV.5.2. Discussion des résultats:Erreur ! Signet non défini.

IV.7.

Conclusion:.....Erreur !

Signet non défini.

Introduction générale

Introduction générale :

Les capteurs sont des composants très petits. Ils sont utilisés essentiellement pour surveiller un environnement. Plusieurs technologies ont contribué à produire des composants de type capteur plus petits. En plus, récemment ces composants sont munis d'un système de communication leur permettant de communiquer avec d'autres capteurs, et ainsi les réseaux de capteurs sans fil sont nés.

Un réseau de capteurs sans fil (WSN) est constitué de plusieurs nœuds (capteurs) dispersés sur une zone géographique vaste afin de collecter des informations sur des évènements bien définis, et de les acheminer vers un nœud particulier de traitement, appelé puits (sink ou station de base).

Aujourd'hui les réseaux de capteurs deviennent de plus en plus répandus. Ils sont utilisés dans divers domaines. Leurs applications sont de plus en plus nombreuses et diversifiées, citant par exemple: le domaine scientifique, logistique, militaire, domotique, environnementale ou encore la santé. Selon MIT's Technology Review, il s'agit de l'une des dix nouvelles technologies qui bouleverseront le monde de notre manière de vivre et de travailler. Ces réseaux sont différents des autres réseaux sans fil car ils ont en général les spécificités suivantes : une grande densité, faible débit, faible capacité d'énergie et un environnement inaccessible. Ces deux dernières spécificités ont fait de l'énergie une contrainte très importante puisque les batteries des capteurs ne sont pas généralement rechargeables. La consommation d'énergie au niveau des capteurs a une grande influence sur la durée de vie du réseau en entier. Il est donc impératif de mettre en place des protocoles de routage efficaces en énergie, et qui prennent en compte les contraintes imposées par ces capteurs.

Pour prolonger la durée de vie d'un réseau de capteur sans fil tout en assurant les trois tâches principales d'un nœud capteur : capture, traitement et l'envoi des données, il faut bien conserver l'énergie des nœuds capteurs. Parmi ces trois tâches, l'envoi des données ou la communication est la tâche qui consomme la plus grande partie de l'énergie. Ceci a motivé

des travaux de recherche à se focaliser sur les couches réseau et MAC (Medium Access Control). Dans notre travail, nous nous intéressons à la couche réseau. Les protocoles de cette couche doit être utilisé efficacement pour prendre en considération la contrainte d'énergie dans les réseaux de capteurs. La plupart des protocoles de communications dans les réseaux Ad-Hoc ne s'adaptent pas aux particularités des réseaux de capteurs. D'où, la nécessité de les améliorer ou de développer de nouveaux protocoles.

Dans ce mémoire, nous proposons de développer un protocole existant Leach qui est l'un des algorithmes de routage hiérarchiques les plus populaires pour les réseaux de capteurs sans fil, il est basé sur l'approche de clustering; Son principal avantage est de minimiser la consommation énergétique des éléments du réseau [12]. Dans ce protocole le réseau est divisé en clusters et chaque cluster possède un nœud maître appelé cluster-Head. Les CHs sélectionnés assurent non seulement la gestion de leurs clusters et la collecte de données mais aussi la transmission de ces données collectées à la station de base. Avec le protocole Leach la communication entre les nœuds CHs et la station de base est une communication directe (en un saut) elle permet de minimiser le délai de livraison de données mais elle épuise l'énergie des CHs sur les réseaux à grande échelle. Car ces derniers sont obligés d'accroître la puissance du signal pour lutter contre l'affaiblissement due à la distance, aux collisions; ainsi nous proposons une communication à plusieurs sauts (en multi sauts) entre les nœuds CHs et la station de base cela permet de réduire efficacement la consommation de l'énergie en adoptant des communications en plusieurs petits sauts, cela minimise les collisions, grâce à l'utilisation des nœuds à faible portée.

Ce mémoire se divise en quatre chapitres :

Dans le **premier chapitre**, nous définissons les réseaux de capteurs sans fil, leur architecture leurs domaines d'applications, et les contraintes liées à ce type de réseau. Nous présentons les différents facteurs intervenants dans la consommation d'énergie et les techniques de minimisation de cette consommation.

Le deuxième chapitre est consacré au routage dans les réseaux de capteurs sans fil, ainsi que certains exemples de protocoles qui ont été élaborés dans des travaux récents réalisés dans le but de conserver l'énergie des nœuds capteurs.

Le troisième chapitre, est sur l'état de l'art des protocoles de routage à multi-sauts dans les réseaux de capteurs.

Dans le **quatrième chapitre**, nous décrivons l'objectif de notre solution et le mécanisme qu'on va utiliser pour l'atteindre. Nous présentons certains exemples de simulateurs dédiés aux réseaux de capteurs sans fils, ainsi les performances de notre solution en termes d'énergie sont détaillées à travers l'outil de simulation OMNeT++.

On conclut ce mémoire par une conclusion générale; Nous proposons également des perspectives de recherche futures.

» Chapitre I«

*Réseaux de capteurs
sans fil.*

» Chapitre II «

*Les protocoles de
routage dans les réseaux
de capteurs sans fil.*

» Chapitre III «

*Etat de l'art des
protocoles hiérarchique
à multi-sauts dans les
réseaux de capteurs
sans fil.*

I.1.Introduction:

Les capteurs sont généralement munis d'une ressource énergétique à durée limitée et sont souvent déployés dans des endroits inaccessibles difficile d'aller les chercher un par un pour les recharger, ceci fait que la durée de vie d'un réseau de capteurs sans fil dépend de la consommation d'énergie.

Dans ce chapitre, nous allons présenter leurs architectures de communication et leurs applications. Les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil. Par la suite, nous décrivons la problématique de la consommation d'énergie dans les réseaux de capteurs sans fil et les facteurs intervenants dans la consommation d'énergie, enfin, nous présenterons les techniques de minimisation de la consommation d'énergie.

I.2.Architecture d'un nœud capteur :

Un nœud de capteurs est composé de 04 unités principales : l'unité de capture, l'unité de traitement, l'unité de communication et l'unité d'énergie :

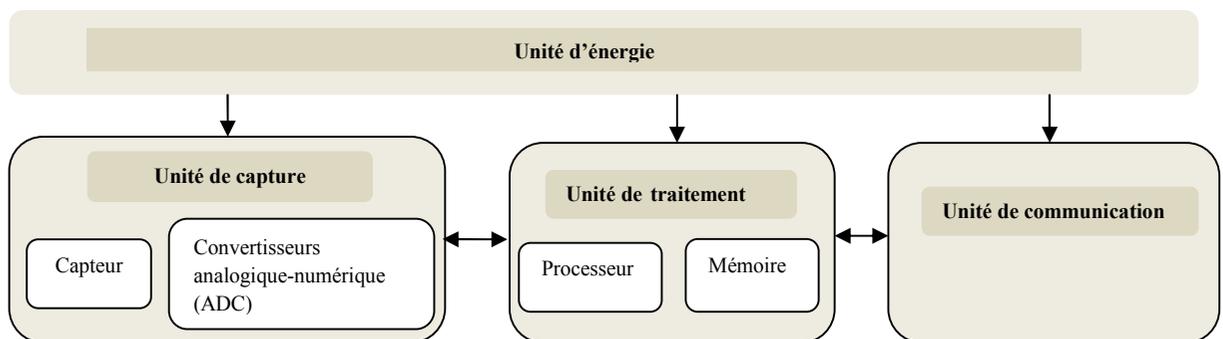


Fig. I.1. Architecture d'un nœud de capteurs.

- **Unité de capture (*Sensing unit*)** : Elle est composée d'un dispositif de capture physique (pour la mesure des grandeurs physiques) et d'un convertisseur analogique-numérique (ADC : Analog to Digital Converters), elle permet la capture des données

de l'environnement local et leurs conversions en données numériques afin de pouvoir être traitées par l'unité de traitement.

- **Unité de traitement (*Processing unit*):** C'est l'unité principale du capteur, les données captées sont communiquées au processeur où elles sont stockées dans la mémoire, elle commande également toutes les autres unités notamment l'unité de communication.

Les microcontrôleurs utilisés dans les réseaux de capteurs sont à faible consommation d'énergie. La majeure partie de l'énergie allouée au microcontrôleur est consommée par leur mémoire, c'est pourquoi on lui adjoint de la mémoire flash moins coûteuse en énergie.

Sur certains capteurs cette unité peut embarquer un système d'exploitation pour faire fonctionner le capteur.

- **Unité de communication (*Transceiver unit*):** son rôle est de transmettre et recevoir l'information, elle est équipée d'un émetteur/récepteur appelé aussi transceiver (transmitter et receiver) permettant la communication entre les différents nœuds du réseau. la quantité d'énergie nécessaire à la transmission augmente avec la distance entre les nœuds.
- **Unité d'énergie (*Power unit*):** Elle correspond le plus souvent à une batterie ou une pile et c'est l'élément principal de l'architecture du capteur elle fournit en énergie toutes les autres unités, généralement elle est ni rechargeable ni remplaçable, ces ressources limitées au niveau des capteurs représente la contrainte principale lors de conception de protocoles pour les RCSF.

I.3. Architecture de communication d'un RCSF :

Un réseau de capteurs sans fil est généralement constitué de plusieurs nœuds de capteurs répartis dans une zone (champs de captage), ces nœuds sont reliés à une ou plusieurs passerelles (sink ou station de base) qui permettent l'interconnexion avec d'autres réseaux (Internet, satellite...) et la récupération des données (un centre de traitement des données) :

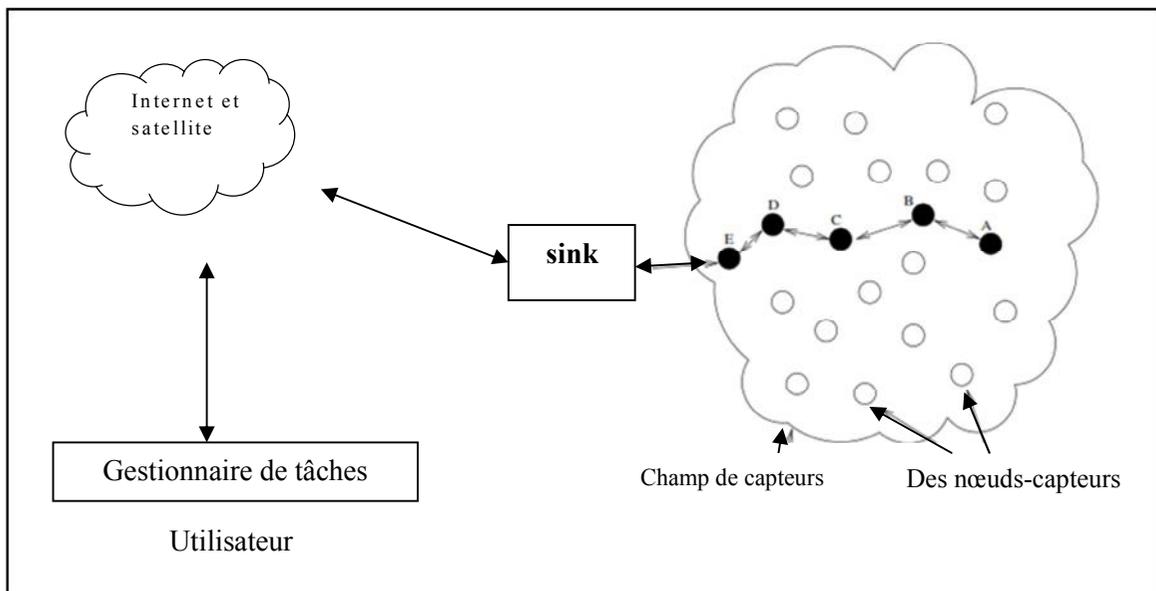


Fig. I.2. Architecture d'un réseau de capteurs.

- **Les nœuds de capteurs** : leur rôle est de collecter les données et de les router vers la station de base.
- **Le champ de captage** : il est considéré comme étant la zone d'intérêt (ou un espace de collecte) pour le phénomène capté, donc les nœuds capteurs y sont placés.
- **Le sink (puits, station de base)** : c'est un nœud particulier chargé d'accueillir, stocké les données en provenance des autres nœuds et de diffuser les différentes requêtes au réseau.
Sa source d'énergie doit être illimitée vu qu'il reste toujours actif pour recevoir les données.
- **Centre de traitement de données (gestionnaire de tâches)**:son rôle consiste à regrouper et à traiter les données collectées par le puits et extraire les informations utiles.

I.4.Pile protocolaire:

La pile protocolaire employée par la station de base (puits) et les autres nœuds de capteurs du réseau est illustrée par la figure : (Fig. I.3.)

Cette pile prend en charge le problème de consommation d'énergie, intègre le traitement des données transmises dans les protocoles de routage, et facilite le travail coopératif entre les capteurs, elle comprend la couche application, la couche transport, la couche réseau, la couche liaison de données, la couche physique, le plan de gestion d'énergie, le plan de gestion de mobilité, le plan de gestion des tâches :

- **Couche application :**

Cette couche assure l'interface avec les applications. Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

- **Couche transport :**

Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

- **Couche réseau :**

La couche réseau s'occupe de l'acheminement des données fournies par la couche transport.

Elle établit les routes entre les nœuds capteurs et le nœud puits et sélectionne le meilleur chemin en terme d'énergie, délai de transmission, etc.

- **Couche liaison de données :**

Elle spécifie comment les données sont expédiées entre deux nœuds/routeurs dans une distance d'un saut. Elle est responsable du multiplexage des données, du contrôle d'erreurs, de l'accès au media, 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.

- **Couche physique :**

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

➤ Les plans de gestion :

- **Plan de gestion d'énergie :** contrôle l'utilisation de la batterie, par exemple : après la réception d'un message, le capteur éteint son récepteur afin d'éviter la duplication des messages déjà reçus. Et si le niveau d'énergie devient bas, le nœud diffuse à ses voisins une alerte les informant qu'il ne peut pas participer au routage. L'énergie restante est réservée au captage.
- **Plan de gestion de mobilité :** détecte et enregistre le mouvement du nœud capteur. Ainsi, un retour arrière vers l'utilisateur est toujours maintenu et le nœud peut garder trace de ses nœuds voisins. En déterminant leurs voisins, les nœuds capteurs peuvent balancer l'utilisation de leur énergie et la réalisation de tâche.
- **Plan de gestion de tâche :** balance et ordonnance les différentes tâches de captage de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de captage au même temps ; certains nœuds exécutent cette tâche plus que d'autres selon leur niveau de batterie.

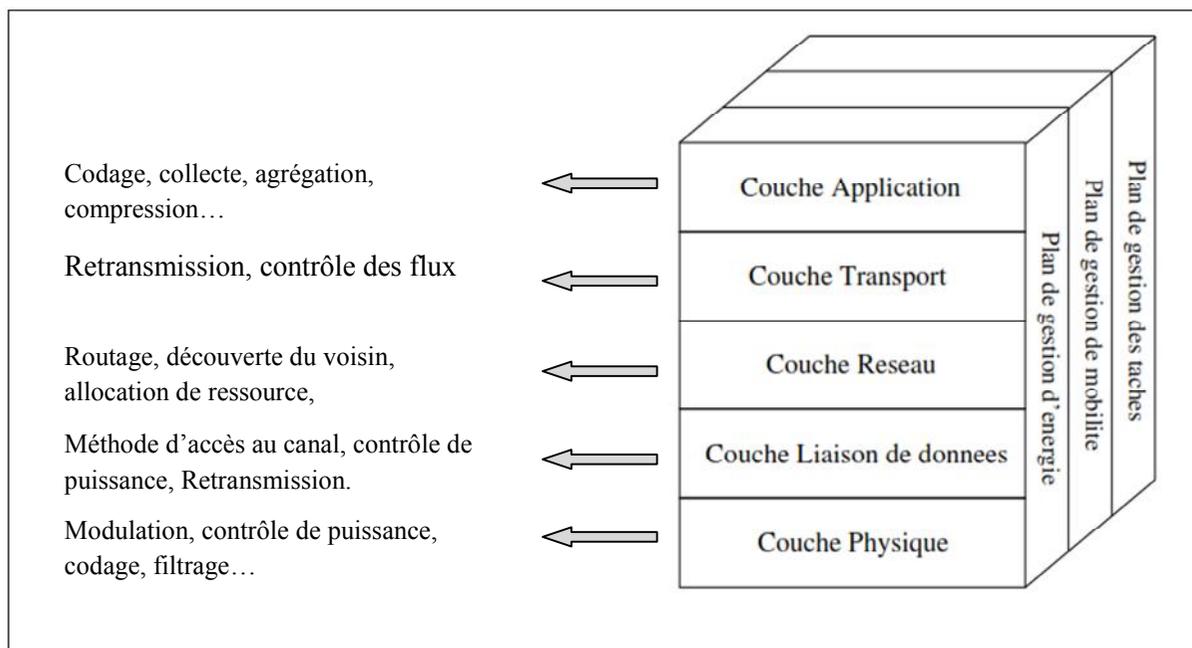


Fig. I.3. La pile protocolaire d'un réseau de capteur.

I.5. Standards de communication dans les réseaux de capteurs sans fils :

Pour assurer la communication entre les nœuds d'un réseau de capteurs, il y a une multitude de normes sans fil comme le wifi (le standard IEEE 802.11) et le **wimax** (le standard IEEE 802.16) qui s'adressent au transport de données à haut débit.

Les capteurs sont parmi les dispositifs qui n'ont pas besoin d'une largeur de bande passante très élevée. Mais ils ont plutôt besoin d'un temps de latence faible ainsi qu'une consommation d'énergie très basse. Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs sans fil se retrouvent les standards bluetooth et zigbee:

- **Bluetooth: IEEE 802.15.4 :**

Elle utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Un grand défaut de cette technologie est sa trop grande consommation d'énergie.

- **zigbee:**

Beaucoup moins connue que Bluetooth, ZigBee est une norme de transmission de données sans fil permettant la communication de machine à machine. Sa très faible consommation électrique et ses coûts de production très bas en font une candidate idéale pour la domotique ou le matériel de type capteur, télécommande ou équipement de contrôle dans le secteur industriel.

Les débits autorisés sont relativement faibles, entre 20 et 250 Kbits/s, mais c'est véritablement sa très faible consommation électrique qui en fait son atout principal. ZigBee fonctionne sur la bande de fréquences des 2,4 GHz et sur 16 canaux, sa portée était au début d'une dizaine de mètres, elle est désormais de 100 mètres.

I.6. Applications des RCSF :

- **Applications orientées temps :** elle représente les applications où l'acquisition et la transmission des données capturées sont liées au temps : un instant précis, une période d'acquisition.

- **Applications orientées requêtes :** un capteur envoie de l'information uniquement suite à une demande explicite de la station de base.
- **Applications hybrides :** Ce type d'application met en œuvre les trois modes de fonctionnement: décrits précédemment. Par exemple, dans un réseau conçu pour le suivi d'objets, le réseau peut combiner entre un réseau de surveillance (applications orientées temps) et un réseau de collecte de données par événements (applications orientées événements).

I.7. Facteurs et contraintes des RCSF :

- **Durée de vie du réseau :**

C'est l'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où l'énergie du premier nœud s'épuise. Selon l'application, la durée de vie exigée pour un réseau peut varier entre quelques heures et plusieurs années.

- **Ressources limitées :**

Les nœuds capteurs ont une capacité de traitement et de mémoire limitée. Ils sont caractérisés par une bande passante limitée. Typiquement, le débit utilisé est de quelques dizaines de Kb/s. Le but de ce faible débit est de minimiser l'énergie consommée lors de transfert de données entre les nœuds. Un débit de transmission réduit n'est pas handicapant pour un réseau de capteurs où les fréquences de transmission ne sont pas importantes.

- **Facteur d'échelle:**

Les réseaux de capteurs peuvent contenir des centaines voire des milliers de nœuds capteurs. Un nombre aussi important engendre beaucoup de transmissions inter nodales et nécessite que le nœud « Sink » soit équipé d'une mémoire importante pour stocker les formations reçues.

- **Topologie dynamique :**

L'instabilité de la topologie des réseaux de capteurs est le résultat ces facteurs essentiels suivants:

La mobilité des nœuds : Dans certaines applications, les nœuds capteurs et les stations de base sont mobiles, les nœuds capteurs peuvent être attachés à des objets mobiles qui se déplacent librement et arbitrairement, introduisant ainsi une topologie instable du réseau.

La défaillance des nœuds : Les nœuds capteurs peuvent être déployés dans des environnements hostiles, la défaillance d'un nœud capteur est, donc très probable.

Un nœud capteur peut devenir non opérationnel : Les capteurs sont généralement dispersés sur une zone de surveillance d'une façon aléatoire et sont la plupart du temps, déployés dans des endroits inaccessibles, donc toute intervention pour changer ou recharger les batteries est pratiquement impossible.

L'ajout de nouveaux nœuds : de nouveaux nœuds peuvent facilement être rajoutés.

Il suffit de placer un nouveau capteur qui soit dans la portée de communication d'au moins un autre nœud capteur du réseau déjà existant.

- **Agrégation de donnée :**

Le coût (consommation de l'énergie) de transmission des données est jusqu'à 70% par rapport au coût général dans un capteur, et comme les réseaux de capteurs sont denses, deux nœuds voisins peuvent capter la même information.

Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie.

L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données. Avec cette technique, les nœuds intermédiaires rassemblent l'information reçue de plusieurs sources. Cette technique est connue aussi sous le nom de fusion de données.

I.8. Consommation d'énergie dans les RCSF :

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

- ✓ **Energie de capture** : cette phase comporte l'échantillonnage des signaux, la conversion analogique/numérique et activation de la sonde du capteur. En général, l'énergie de capture représente un faible pourcentage de l'énergie totale consommée par un nœud.
- ✓ **Energie de traitement** : cette phase comporte : l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel).
Par contre l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.
- ✓ **Energie de communication** : les communications consomment beaucoup plus d'énergie que les autres tâches, L'énergie de communication se divise en deux parties:
L'énergie de réception et l'énergie de l'émission.

I.9. Facteurs intervenants dans la consommation d'énergie :

La consommation d'énergie dépend de plusieurs facteurs suivants :

- ✓ **Etat du module radio** : est le composant qui consomme le plus d'énergies car il assure la communication entre les nœuds .Le module radio opère dans quatre modes de fonctionnement : inactif, émission, réception, et sommeil.

Une observation importante dans le cas de la plupart des radios est que le mode « inactif » induit une consommation d'énergie significative, presque égale à la consommation en mode réception.

Ainsi, plutôt que de passer en mode « inactif » quand l'on a ni à émettre ni à recevoir de données il est plus judicieux d'éteindre complètement la radio.

Un autre facteur déterminant est que, le passage de la radio d'un mode à un autre engendre une consommation d'énergie importante due à l'activité des circuits électroniques.

✓ Accès au médium de transmission :

- **Réception indésirable (*overhearing*) :** quand un nœud reçoit des paquets qui ne lui sont pas destinés. Le coût de l'écoute abusive peut être un facteur dominant de la perte d'énergie quand la charge de trafic est élevée et la densité des nœuds est grande.
- **Collision :** elles sont la première source de perte d'énergie. Quand deux trames sont émises en même temps et se heurtent, elles deviennent inexploitables et doivent être abandonnées. Ce que nécessite de les retransmettre de nouveau.
- **La surcharge :** l'envoi, la réception, et l'écoute des paquets de contrôle consomment de l'énergie. Comme les paquets de contrôle ne transportent pas directement des données, ils réduisent également le débit utile effectif.
- **Ecoute de la porteuse à vide (*idle listening*) :** un nœud dans l'état inactif « idle » est prêt à recevoir un paquet, mais il n'est pas actuellement en train de recevoir quoi que ce soit. Ceci est coûteux et inutile dans le cas des réseaux à faible charge de trafic.
- **Non disponibilité du récepteur (*over emitting*) :** le phénomène de surémission quand un nœud envoie des données et le nœud destinataire n'est pas prêt à les recevoir.

✓ Routage des données :

Le routage dans les RCSF est un routage multi-sauts. L'acheminement des paquets d'une source donnée à une destination se fait à travers plusieurs nœuds intermédiaires. Ainsi, un nœud consomme de l'énergie soit pour transmettre ces données soit pour relayer les données des autres nœuds.

Dans ce contexte, une mauvaise politique de routage peut avoir des conséquences graves sur la durée de vie du réseau.

I.10. Techniques de minimisation de la consommation d'énergie :

Maximiser la durée de vie du réseau revient à minimiser la consommation énergétique des nœuds, Nous présentons ici les techniques de conservation d'énergie, ces techniques sont appliquées soit au niveau de la couche liaison (sous couche MAC) ou au niveau de la couche réseau.

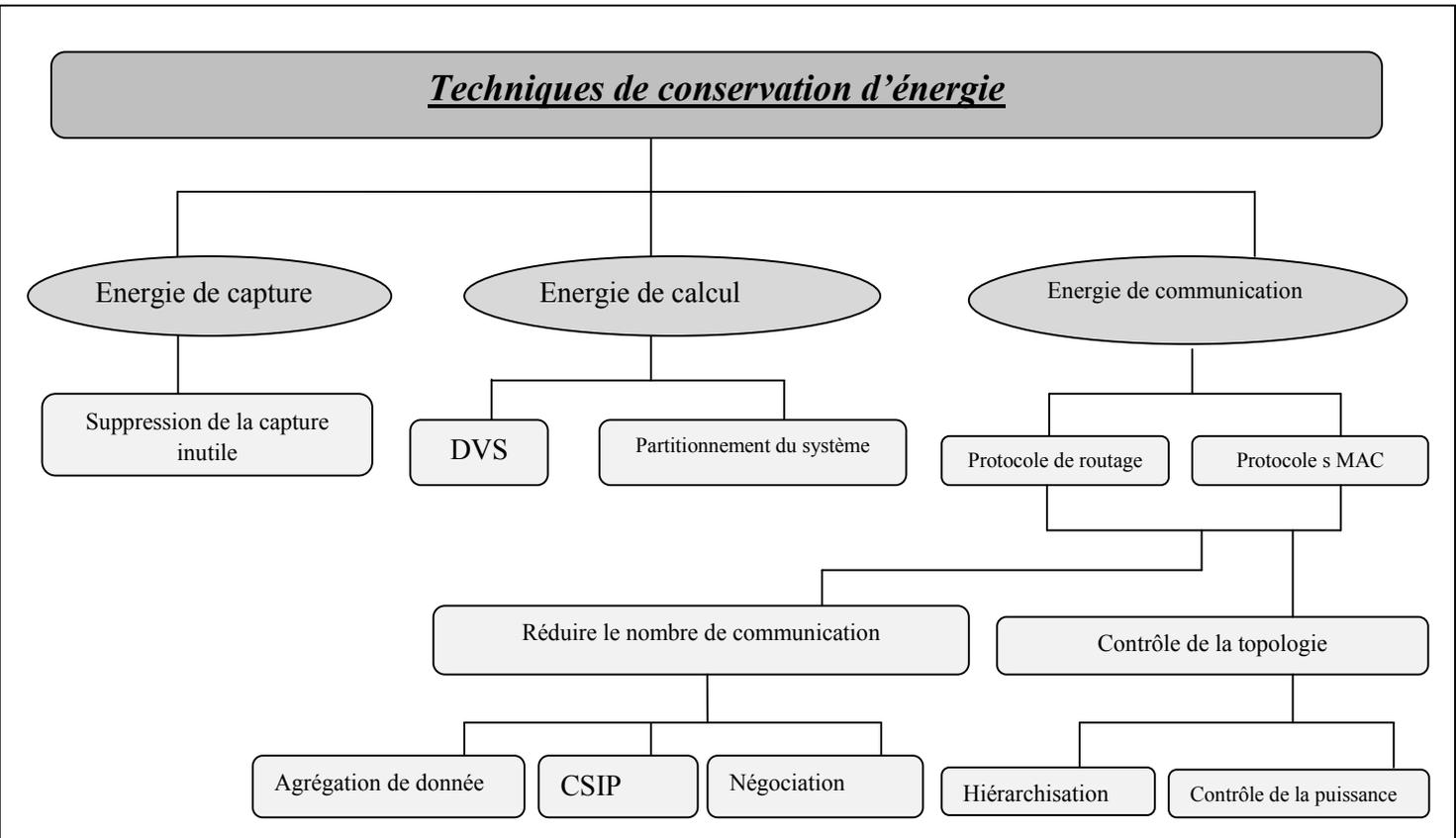


Fig. I.6. Les techniques de conservation d'énergie.

- **Energie de capture** : La seule solution apportée pour la minimisation de la consommation d'énergie au niveau de la capture Consiste à supprimer la capture inutile donc à réduire les durées de captures.

- **Energie de calcul :** L'énergie de calcul peut être optimisée en utilisant deux techniques :
 - **L'approche DVS (Dynamique Voltage Scaling)** [01] : qui consiste à ajuster de manière adaptative la tension d'alimentation et la fréquence de microprocesseur pour économiser la puissance de calcul sans dégradation des performances.
 - **L'approche de partitionnement de système :** qui consiste à transférer un calcul prohibitif en temps de calcul vers une station de base qui n'a pas de contraintes énergétiques et qui possède une grande capacité de calcul [02].

- **Energie de communication :**

Une technique a été proposée dans [0 3]. Cette technique profite de la densité élevée des capteurs déployés pour se permettre d'endormir certains d'entre eux, afin que tous les capteurs ne soient pas actifs en même temps.

Une panoplie de protocoles a été proposée dans la littérature. La grande partie de ces protocoles est destinée à la couche réseau et MAC.

Les protocoles de la couche réseau dans les RCSF peuvent être divisés selon ce qui suit :

Structure du réseau en routage linéaire : dans le routage linéaire, tous les nœuds ont typiquement les mêmes rôles ou fonctionnalités.

Routage hiérarchique : dans le routage hiérarchique, les nœuds joueront différents rôles dans le réseau.

- **Réduire le nombre de communication :**

La minimisation de la consommation d'énergie pendant la communication est étroitement liée aux protocoles développés pour la couche réseau et la couche MAC. Ces protocoles se basent sur plusieurs techniques : agrégation de données, négociation et CSIP (Collaborative Signal and Information Processing). Cette dernière technique est une discipline qui combine plusieurs domaines [04] : la communication et le calcul à basse puissance, traitement de signal, algorithmes distribués et tolérance aux fautes, systèmes adaptatifs et théorie de fusion

Chapitre I : Les réseaux de capteurs sans fil

des capteurs et des décisions. Ces techniques ont pour but de réduire le nombre d'émission/réception des messages donc réduire le nombre de communication.

- **Le contrôle de la topologie** : [05] permet l'ajustement de la puissance de transmission et le regroupement des nœuds capteurs (hiérarchisation) :

- ✓ **Le contrôle de la puissance de transmission** : il n'a pas seulement un effet sur la durée de vie de la batterie d'un nœud capteur, mais aussi sur la capacité de charge du trafic qui est caractérisée par le nombre de paquets transmis avec succès vers une destination. En outre, il influe sur la connectivité et la gestion de la densité (le nombre de nœuds voisins). Ainsi, il peut conserver l'énergie à deux niveaux : explicitement par l'application de puissances faibles d'émissions et implicitement en réduisant la contention avec d'autres nœuds transmetteurs. Le module de contrôle de la puissance est souvent intégré dans les protocoles soit de la couche réseau soit de la couche MAC [06].

- ✓ **La hiérarchisation** : consiste à organiser le réseau en structure à plusieurs niveaux. C'est le cas, par exemple, des algorithmes de groupement (clustering), qui organisent le réseau en groupes (clusters) avec des chefs de groupe (cluster head) et des nœuds membres [07].

I.11.Conclusion:

Dans les réseaux de capteurs sans fil, chaque nœud est alimenté par une source d'énergie limitée et généralement irremplaçable. Dans un nœud capteur, l'énergie est consommée en assurant la capture, le calcul (traitement) et la communication. Cette dernière représente une grande portion de l'énergie totale consommée. De ce fait, les réseaux de capteurs doivent, prendre en compte la conservation d'énergie. Ils doivent intégrer des mécanismes qui permettent aux utilisateurs de prolonger la durée de vie du réseau en entier, la communauté de recherche est en train de développer et de raffiner plusieurs techniques de conservation d'énergie.

Dans le chapitre suivant, nous examinerons en détail les solutions proposées pour la gestion effective de l'énergie au niveau de la couche réseau.

» Chapitre II «

*Les protocoles de
routage dans les réseaux
de capteurs sans fil.*

Chapitre II : Les protocoles de routage dans les RCSEs

II.1.Introduction:

Le routage de données sur les WSNs est considéré comme le domaine le plus exploré parmi les domaines de recherche sur les WSNs. Le routage dans les réseaux de capteurs consiste à déterminer les routes reliant les nœuds capteurs et les nœuds puits et acheminer les données captées à travers ces routes.

Dans ce chapitre nous présentons les facteurs décisifs de développement d'un protocole de routage pour les WSNs, puis leur classification qui les groupe suivant un certain nombre de critères ainsi nous donnerons à la fin quelques exemples de ces protocoles :

II.2.Routage dans les réseaux de capteurs:

Le routage sur les WSNs consiste, en général, à acheminer les données collectées par les nœuds du réseau vers un seul destinataire qui est le puits (Sink). Ces données sont caractérisées par une corrélation spatiale et temporelle très importante relative à la densité élevée du déploiement des nœuds.

Par conséquent, les protocoles classiques de routage de données sont inadaptés pour les WSNs et un nouveau paradigme de routage s'avère très nécessaire. Il prend en charge les contraintes physiques relatives à l'utilisation des capteurs limités en taille de stockages et en puissance de calcul, et à l'utilisation du médium sans fil, pour la communication, qui est peu fiable et limité en distance de couverture.

II.2.1.Facteurs pour le développement de protocole de routage:

La performance des réseaux de capteurs sans fil est fondée sur les facteurs suivants :

- **L'énergie :**

Le facteur le plus important à prendre en considération est l'énergie consommée par un capteur lors de la détection et de la transmission des données captées sur le réseau. Nous avons

Chapitre II : Les protocoles de routage dans les RCSE_s

vu que la transmission est la fonction qui consomme le plus d'énergie. Un nœud dans le réseau doit savoir combien d'énergie sera utilisée pour effectuer une nouvelle tâche à laquelle il est soumis. L'énergie consommée peut varier selon le type de fonctionnalité ou l'activité qu'il a à accomplir.

- **Le temps de traitement:**

Il se réfère au temps pris par le nœud dans le réseau pour assurer l'ensemble de l'opération commençant par la détection, le traitement des données ou le stockage de données, la transmission ou la réception sur le réseau.

- **Le schéma de transmission:**

La transmission de données par les nœuds de capteurs vers la destination ou la station de base se fait par un schéma de routage à un seul saut ou à multi-saut.

- **La capacité du réseau :**

Tous les nœuds du réseau de capteurs utilisent certaines ressources du réseau qui les aident à accomplir certaines activités comme la détection ou la transformation.

- **Synchronisation :**

Dans les communications radio entre les nœuds de capteurs d'un WSN, les capteurs écoutent en permanence les transmissions et consomment de l'énergie s'ils ne sont pas synchronisés les uns les autres. Pour cela, un nœud doit avoir la même notion de temps pour se mettre en veille et se réveiller que ses voisins.

- **Contrôle de paquets:**

Un paquet envoyé avant la transmission entre deux nœuds est appelé le paquet de contrôle. Le paquet de contrôle contient le nombre de bits de données envoyés, l'adresse du nœud de destination et certaines informations qui contribuent à éviter les collisions pendant la transmission.

Chapitre II : Les protocoles de routage dans les RCSEs

- **La mobilité:**

C'est une question clé sur les WSNs car les capteurs sont souvent embarqués sur des dispositifs mobiles comme les voitures ou les animaux. Selon le degré de cette mobilité, la topologie du réseau change d'une manière très fréquente; ainsi, connaître ses voisins et reconfigurer le réseau nécessitent un nombre de messages important, donc une dépense plus importante d'énergie.

II.3.Classification des protocoles de routage:

II.3.1.Classes principales des protocoles de routage selon la topologie:

- **Routage plat:** Dans une topologie plate, tous les nœuds capteurs possèdent le même rôle et collaborent entre eux pour accomplir la tâche de routage. Les réseaux plats sont caractérisés par : la simplicité de protocole de routage, un coût de maintien réduit, une grande tolérance aux pannes ainsi une habilité à construire de nouveaux chemins suite aux changements de topologie. Cependant le réseau présente une faible scalabilité dû grand nombre de messages de contrôle .la figure suivante illustre l'organisation des capteurs dans une topologie plate.

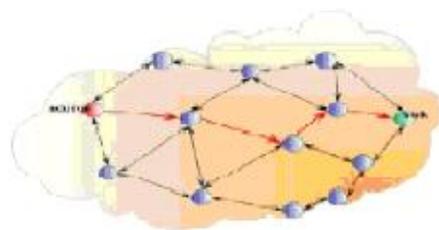


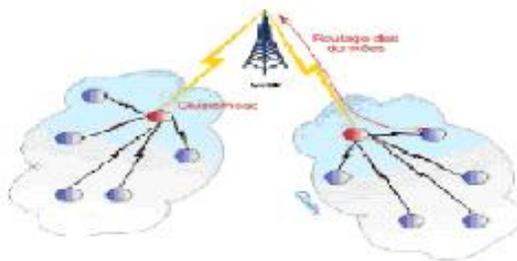
Fig.II.1.Topologie plate.

Chapitre II : Les protocoles de routage dans les RCSEs

➤ **Routage hiérarchique:**

Les méthodes de routage hiérarchique ont des avantages spéciaux liés au passage à l'échelle et à l'efficacité dans la communication. Dans une architecture hiérarchique, des nœuds à grande énergie peuvent être employés pour traiter et envoyer l'information, alors que des nœuds à énergie réduite peuvent assurer la capture à proximité de la cible. L'une des méthodes les plus utilisées dans cette topologie est le clustering. Il consiste en un partitionnement du réseau en groupes de clusters. Un cluster est constitué d'un chef (Cluster-Head) et de ses membres. La création des clusters et l'assignation des tâches spéciales aux têtes de clusters peuvent considérablement renforcer le passage à l'échelle, l'augmentation de la durée de vie et l'efficacité énergétique du système global.

Le routage hiérarchique est une manière efficace de réduire la consommation énergétique dans un cluster en exécutant l'agrégation et la fusion de données afin de diminuer le nombre de messages transmis à la station de base.



FI.II.2.Topologie hiérarchique.

II.3.2.sous-classes des protocoles de routages:

- **Routage basé sur les chemins multiples:** Les protocoles de cette sous-classe utilisent des chemins de routage multiples au lieu d'un chemin simple entre une source et une destination. La fiabilité d'un protocole est mesurée par la vraisemblance qu'un chemin

Chapitre II : Les protocoles de routage dans les RCSF_s

alternatif existe entre une source et une destination en cas de défaillance de chemin primaire. Cette tolérance à la panne 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.

➤ **Routage basé sur les requêtes:** Dans le routage à base de 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 de nouveau au nœud à l'initiative de la requête. Généralement, ces requêtes sont décrites dans un langage naturel exprimées par un schéma attribut-valeur ou des langages d'interrogation de niveau élevé par exemple SQL (Structured Query language).

➤ **Routage basé sur la négociation entre les nœuds:** 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.

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 (métadonnées) avant même la transmission effective des données.

➤ **Routage basé sur la cohérence des données:** Le traitement de la cohérence des données est une phase importante dans le fonctionnement des RCSF. Par conséquent, des algorithmes de routage utilisent différentes techniques pour traiter la cohérence (ou la non-cohérence) des données circulant dans le réseau. En général, les nœuds capteurs coopèrent entre eux afin de réaliser ce traitement. Dans le routage basé sur la non-cohérence des données, ces dernières sont envoyées aux nœuds agrégateurs du réseau après avoir reçu le traitement minimum qui inclut la suppression des doublures [08]. Pour exécuter un routage efficace en énergie, le traitement de la cohérence des données est choisi par le concepteur du

Chapitre II : Les protocoles de routage dans les RCSFs

protocole.

- **Routage avec qualité de service:** Le principe des protocoles de routage avec QoS se base sur le fait que le réseau doit être capable de satisfaire certaines métriques (latence, énergie des nœuds, bande passante, et.) tout en acheminant le maximum de données vers la station de base.

Les protocoles de cette approche sont très recommandés pour les applications de surveillance (centrales nucléaires, applications militaires, etc.).

II.4.Exemples des protocoles de routages dans les RCSF:

Les protocoles de routage minimisant la consommation d'énergie dans les RCSFs sont extrêmement nombreux et variés. Chaque protocole est conçu de façon à fonctionner suivant des critères ou des conditions particulières comme la structure du réseau, l'établissement de la route...etc. [09], nous allons présenter quelques exemples de ces protocoles:

II.4.1.Exemples de protocoles de routage plat:

Le but principal dans ce routage est de minimiser les redondances de données pouvant devenir très pénalisantes surtout en termes de consommation d'énergie et surcharge du réseau. Dans un tel routage, la station de base diffuse ces requêtes vers des régions particulières ou tout le réseau. Elle attend ensuite, les données correspondant à ses requêtes depuis les nœuds situés dans la région cible ou d'intérêt. Parmi ces protocoles, nous trouvons :

II.4.1.1. SPIN: [10]

Heinzelman et al ont proposé une famille de protocoles appelée SPIN (Sensor Protocols for Information via Negotiation), reposant sur un modèle de négociation afin de propager l'information dans un réseau de capteurs:

Chapitre II : Les protocoles de routage dans les RCSEs

L'idée de SPIN est de nommer les données en utilisant des descripteurs de haut niveau ou des métas données. Avant la transmission, les métadonnées sont échangées entre les capteurs par un mécanisme de publicité de données. Chaque nœud recevant de nouvelles données, l'annonce à ses voisins et ceux intéressés récupèrent les données en envoyant une requête.

Le but de SPIN est de remédier aux problèmes de l'inondation, qui sont :

- L'implosion due à la duplication inutile des réceptions d'un même message.
- Le chevauchement lié au déploiement dense des capteurs. En utilisant l'inondation, les capteurs d'une zone émettent tous la même donnée (ou presque).
- L'ignorance des ressources, car l'inondation ne prend pas en considération les ressources des nœuds.

Ces trois problèmes affectent la durée de vie et les performances du réseau. Pour les résoudre, SPIN adopte deux principes :

- La négociation: pour éviter le problème d'implosion, SPIN précède l'émission d'une donnée par sa description, en utilisant la notion de métadonnées. Le récepteur aura le choix par la suite d'accepter la donnée ou non. Ce mécanisme permet aussi de régler le problème de chevauchement.
- L'adaptation aux ressources: d'une manière continue, les nœuds contrôlent leur niveau d'énergie. Le protocole SPIN accommode son exécution suivant l'énergie restante du capteur, et modifie en conséquence le comportement du nœud.

Fonctionnement de SPIN :

Les communications dans SPIN se font en trois étapes :

- **Étape 1: Emission du message ADV** (ID-du-nœud+ Entête): nœud veut émettre une donnée, il émet d'abord un message ADV contenant une description de la donnée en

Chapitre II : Les protocoles de routage dans les RCSE_s

question.

- **Etape 2: Emission du message REQ** (ID-du-nœud+ Requête): Un nœud recevant un message ADV, consulte sa base d'intérêt. S'il est intéressé par cette information, il émet un message REQ vers son voisin.

Lorsque le nœud s'aperçoit que son énergie est descendue sous un certain seuil, il change son mode de fonctionnement, et ne répond à aucun message ADV

- **Etape 3: Emission du message DATA** (ID-du-nœud+ Entête+ donnée.): En recevant un message REQ, l'émetteur transmet à l'intéressé la donnée sous forme d'un message DATA.

La figure suivante illustre ces trois étapes :

Le nœud **A** annonce ses données au nœud **B** (a). **B** répond par une requête (b). **B** reçoit les données requises (c). **B** fait de la publicité à ses voisins (d) qui répondent par des requêtes (e-f).

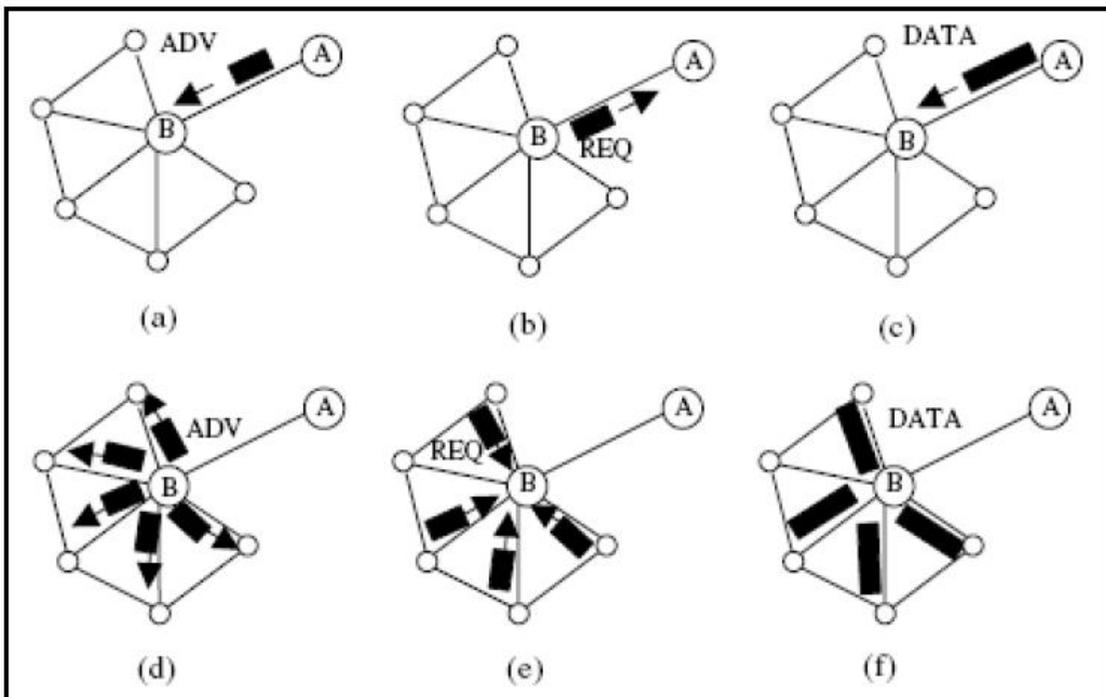


Fig.II.3. Fonctionnement du protocole SPIN [10].

Chapitre II : Les protocoles de routage dans les RCSEs

II.4.1.2. Diffusion dirigé (DD): [11] :

Directed Diffusion est un protocole de propagation de données, permettant d'utiliser plusieurs chemins pour le routage d'information. Le puits diffuse un intérêt sous forme de requête, afin d'interroger le réseau sur une donnée particulière. Il se base sur le modèle publish/subscribe. 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.

DD repose sur les éléments suivant: nomination des données, propagation des intérêts et établissement des gradients, propagation des données et renforcement des chemins.

1. Nomination des données :

- La diffusion dirigée suggère l'utilisation de paires attribut-valeur pour décrire les intérêts et les rapports de données.

Example:

```
Type = GetTemperature  
Zone = [100, 100, 120, 120]  
Interval = 10 ms //Send event every 10ms
```

2. Propagation des intérêts et établissement des gradients :

A. Propagation des intérêts :

Afin de propager l'intérêt, DD emploie l'inondation globale du réseau. Chaque nœud maintient localement un cache d'intérêt contenant les informations ci-dessous :

- La description de l'intérêt.
- Des gradients

Gradient: Un gradient est un vecteur représentant l'intérêt. Il est caractérisé par une direction et une amplitude et la date d'expiration de données.

La direction est modélisée par le voisin émetteur de l'intérêt,

L'amplitude est représentée par le débit de données.

Chapitre II : Les protocoles de routage dans les RCSE_s

B. Etablissement des gradients :

- Lorsqu'un nœud reçoit un intérêt, il parcourt son cache :
 - Si le cache ne contient aucune entrée relative à l'intérêt reçu, une nouvelle entrée est créée avec un gradient vers le voisin émetteur.
 - Dans le cas contraire, le nœud recherche un gradient vers le voisin émetteur, et met à jour en conséquence l'entrée en question.
- Après le traitement du cache, le nœud relaie l'intérêt vers ses voisins. La méthode la plus simple est d'utiliser l'inondation.

3. propagation des données :

- Les capteurs commencent la récolte d'information. Un capteur calcule le débit le plus élevé et prélève les données en conséquence.
Le nœud détermine les prochains sauts vers les puits (chacun avec son propre débit).

- Lorsqu'un nœud reçoit une donnée, il recherche un intérêt équivalent dans son cache.
 - Si aucune entrée n'est trouvée, le paquet est supprimé (éviter les boucles).

Dans le cas contraire, en consultant la liste des gradients, le nœud relaie la donnée vers ses voisins, suivant le débit de chacun d'eux.

4. Renforcement des chemins :

- Lorsqu'un nœud reçoit un message de renforcement, il consulte son cache d'intérêt
 - Si le débit spécifié dans le message est plus grand que tous les autres débits des gradients présents, le nœud doit renforcer un de ses voisins pour envoyer des données plus fréquemment.

Ce protocole est différent de SPIN en deux aspects:

- Dans SPIN, c'est les nœuds sources de données qui diffuse eux-mêmes les métas-data pour que les autres nœuds qui les requièrent puissent les retrouver aisément lors de la phase de recherche de données. Alors que, dans DD c'est la station de base qui diffuse les requêtes sous forme d'intérêt.
- Dans DD la communication s'effectue de voisin a voisin ou chaque nœud agrège les données

Chapitre II : Les protocoles de routage dans les RCSE_s

et garde en mémoire le chemin de provenance, par contre, dans SPIN aucune sauvegarde de route n'est opérée.

Cependant, les échanges fréquents et réguliers d'intérêts et de gradient peuvent générer des surconsommations d'énergie au niveau des capteurs.

Cette figure illustre le fonctionnement du protocole DD:

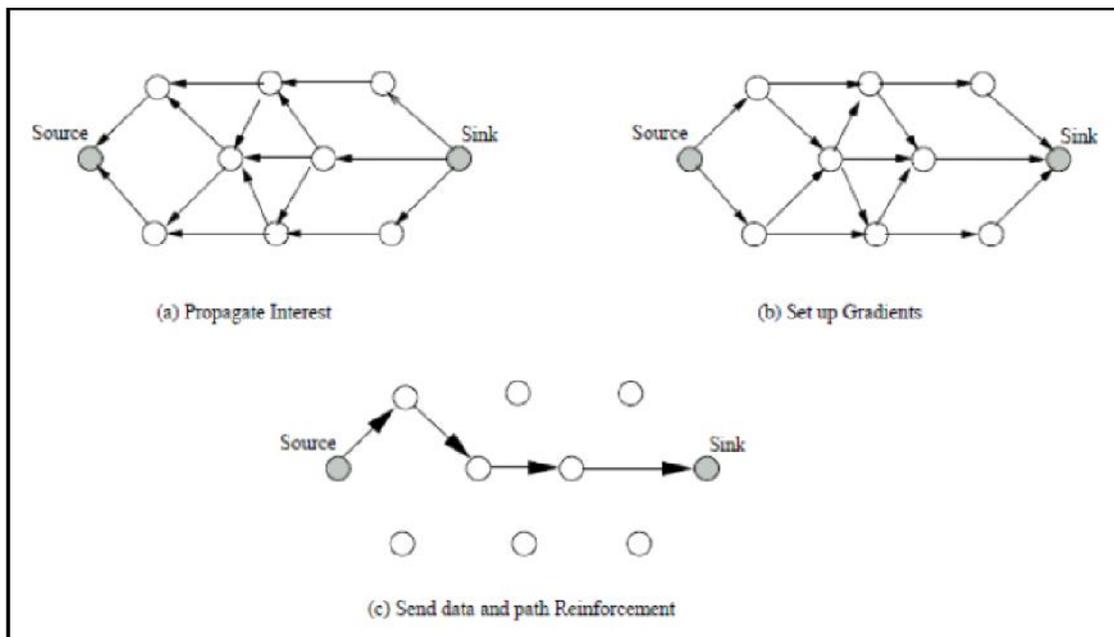


Fig.II.4.Fonctionnement du protocole DD [11].

II.4.2.Exemples de Protocoles hiérarchiques:

Ces protocoles utilisent différentes techniques d'agglomération (ou clustering) et d'agrégation de données pour minimiser le nombre de messages transmis à la station de base.

La formation des grappes et le choix de leurs chefs sont basés sur la réserve d'énergie des capteurs et sur la proximité de ces derniers du chef à élire. Parmi ces protocoles, il y a :

Chapitre II : Les protocoles de routage dans les RCSEs

II.4.2.1. LEACH (Low-Energy Adaptive Clustering Hierarchy): [12]:

Le protocole LEACH est le plus populaire des protocoles de routage hiérarchique. Son principal avantage est de minimiser la consommation énergétique des éléments du réseau.

LEACH est un protocole conçu pour le routage dans les réseaux de capteurs homogènes où les capteurs ont les mêmes caractéristiques et les mêmes capacités.

Dans ce protocole le réseau est divisé en clusters et chaque cluster possède un nœud maître appelé cluster-Head. Ce dernier prend en charge la gestion de son cluster, il communique directement avec la station de base ce qui permet de minimiser la consommation et réduire la quantité d'informations envoyées à la station de base. Le cluster-Head est élu périodiquement parmi les nœuds formant le cluster, en fonction de l'état de sa batterie.

Le protocole se déroule en rounds. Chaque round se compose de deux phases qui se succèdent plusieurs fois: la phase construction et la phase communication.

- **Phase de construction (initialisation):**

le but de cette phase est la construction des clusters en choisissant les chefs (CHs), durant cette phase, chaque nœud n choisit un nombre aléatoire compris entre 0 et 1. si ce nombre est inférieur à un seuil $t(n)$, le nœud devient cluster Head. $T(n)$ est défini comme suit:

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

Avec:

p : le pourcentage des CHs sur le réseau (généralement 5%);

Chapitre II : Les protocoles de routage dans les RCSE_s

r: numéro du round en cours;

G: l'ensemble des nœuds qui n'était pas CH dans les $(1/p)$ rounds précédentes.

Chaque nœud qui s'est élu cluster Head émet un message de notification à tous ces voisins. Les nœuds qui récoltent les messages de notification, décident leur appartenance à un cluster. La décision est basée sur l'amplitude du signal reçu: le cluster Head ayant le signal le plus fort (plus proche) est choisi. En cas d'égalité, un chef aléatoire est choisi. Chaque membre informe son chef de sa décision.

- **Phase de communication (transmission):** elle représente les communications à l'intérieur d'un cluster qui est effectuées avec la méthode TDMA (mode permettant de transmettre plusieurs signaux sur un seul canal cette méthode empêche la collision entre les messages de données ainsi elle conserve l'énergie dans les nœuds non CH).

Chaque CH établit un *schedule TDMA* pour les membres de sa zone en indiquant pour chaque nœud son slot d'émission. Les membres émettent donc les données captées pendant ce slot d'émission. Ce mécanisme permet aux différents capteurs d'éteindre leur interface de communication en dehors de leur slot d'émission, ainsi ils économisent leur énergie. Ensuite les données agrégées (au niveau des CHs) sont envoyées directement au puits.

Description de l'algorithme LEACH :

LEACH utilise un algorithme distribué où chaque nœud décide d'une manière autonome s'il sera Cluster Head ou non en calculant aléatoirement une probabilité p et en la comparant à un seuil $T(n)$; puis, il informe son voisinage de sa décision.

Chaque nœud non ClusterHead décide du cluster à rejoindre en utilisant un minimum d'énergie de transmission (en choisissant le plus proche).

L'algorithme se déroule en plusieurs rounds qui ont approximativement le même

Chapitre II : Les protocoles de routage dans les RCSE_s

intervalle de temps déterminé au préalable, Chaque round est constitué d'une phase d'initialisation et d'une phase de transmission.

- Chaque nœud au début de chaque round annonce sa probabilité (selon son niveau d'énergie courant) d'être CH à tous les autres nœuds.
- Les nœuds de chaque round avec la plus forte probabilité sont choisis comme CH
- Les CH diffuse un message d'annonce (ADV) en utilisant le protocole CSMA MAC
- Basé sur l'intensité du signal reçu, chaque nœud non CH détermine son CH pour ce round.
- Chaque nœud non CH transmet un message de demande d'adhésion au cluster (Join-REQ) vers le CH choisi à l'aide d'un protocole CSMA MAC.
- Le CH met en place un ordonnancement TDMA pour la coordination des transmissions de données au sein du cluster.

L'accès multiple par répartition temporelle (TDMA)

Cette technique est basée sur un multiplexage temporel (utilisé par exemple en téléphonie) qui consiste à dire que les utilisateurs utilisent la même fréquence tout en occupant des slots de temps différents. Chaque nœud utilise toute la bande passante allouée par le système de transmission durant le slot (c'est lui seul qui occupe le support).

L'accès multiple avec écoute de la porteuse CSMA :

Constitue une technique aléatoire dans laquelle chaque nœud doit rester à l'écoute du médium avant de transmettre ses données.

Chapitre II : Les protocoles de routage dans les RCSE_s

En cas où le médium est occupé, le nœud doit retarder son accès au médium et réessayer la même procédure jusqu'à ce que le médium devienne libre. A cet instant, il pourra transmettre ses données.

➤ Organigramme de la phase Setup :

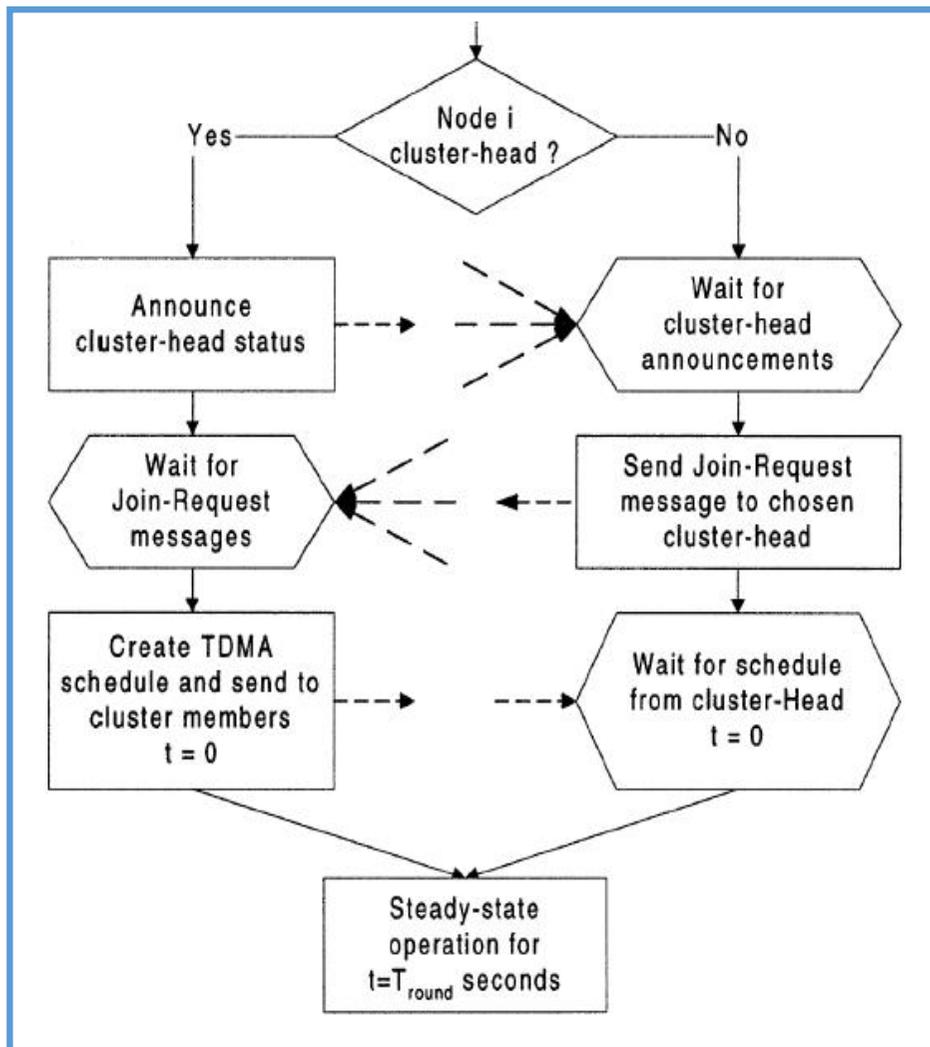


Fig.II.5. Organigramme de la phase Setup du protocole leach.

Chapitre II : Les protocoles de routage dans les RCSE_s

➤ Le diagramme de la phase steady :

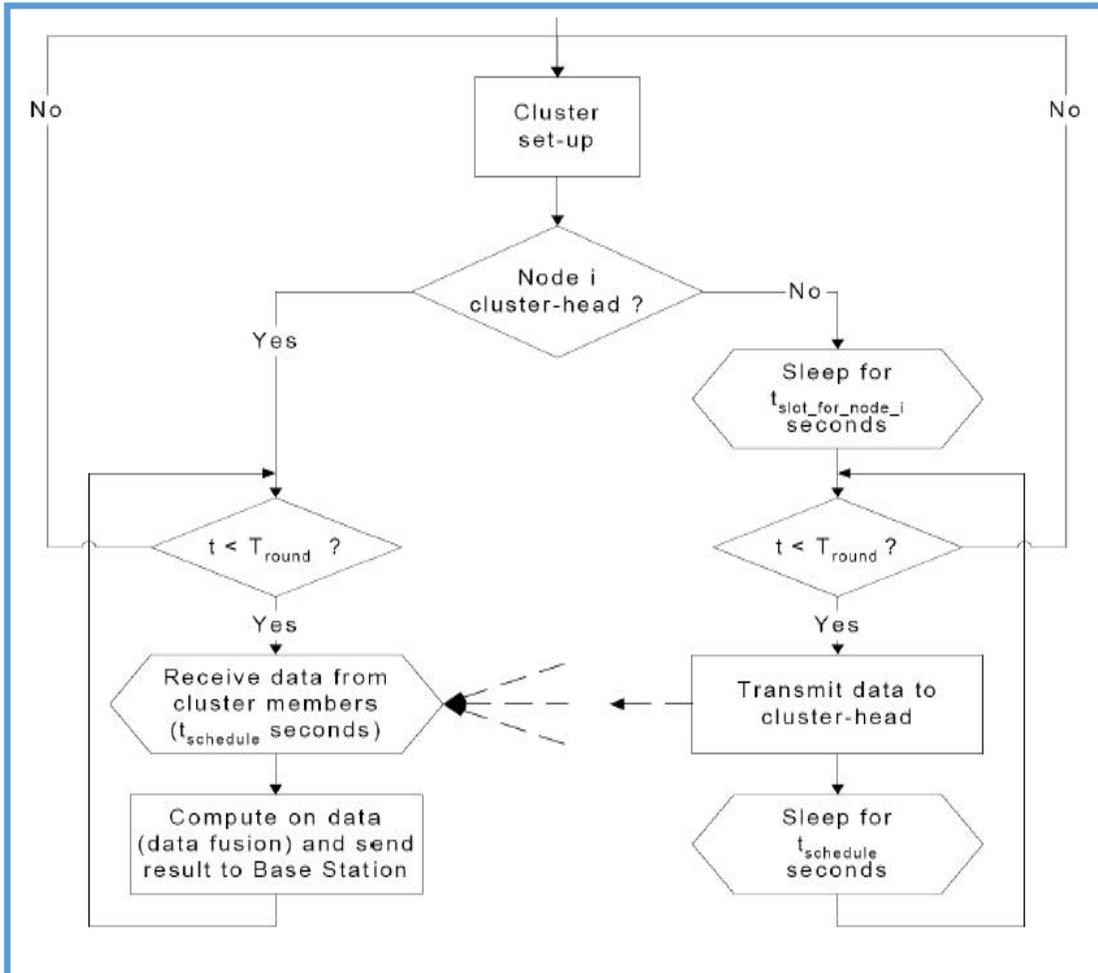


Fig.II.6. Organigramme de la phase steady du protocole leach.

Avantage :

Le protocole LEACH présente les avantages suivants :

- ✓ L'auto-configuration des clusters se fait indépendamment de la station de base.
- ✓ Les données sont fusionnées pour réduire la quantité d'informations transmise vers la

Chapitre II : Les protocoles de routage dans les RCSE_s

station de base. Vu que LEACH préconise une agrégation de données au niveau des CHs pour plus de conservation d'énergie.

- ✓ La consommation d'énergie est partagée sur l'ensemble des nœuds prolongeant ainsi la durée de vie du réseau.
- ✓ LEACH suppose que chaque nœud du réseau peut communiquer directement avec le puits; alors que, les nœuds non-ClusterHead ne peuvent communiquer qu'avec leurs ClusterHead choisi, en utilisant la technique TDMA instaurée par ce dernier. Cette technique permet de minimiser les collisions en allouant à chaque nœud un temps privé pour transmettre ses données vers son CH.

Cependant, plusieurs critiques sont apportées au protocole LEACH :

- La possibilité de communiquer avec le puits à travers n'importe quel nœud (les CHs sélectionnés) du réseau exige une consommation d'énergie importante des nœuds lointains. Ce qui rend le protocole moins apte au passage à l'échelle.
- L'agrégation des données est centrée au niveau des CHs, ce qui les rend les maillons faibles du réseau. Ainsi les CHs les plus éloignés de la station de base meurent rapidement par rapport à ceux qui sont proches de la station.
- LEACH ne garantit pas une distribution homogène des CHs sur le réseau, car le seul critère d'élection du CH est une probabilité aléatoire. Cela n'empêche pas une concentration des CHs dans une région limitée au détriment de l'ensemble du réseau.
- Sans justifier leur choix, les auteurs fixent le pourcentage optimal de CHs pour le réseau à 5% du nombre total des nœuds. Néanmoins, la topologie, la densité et le nombre de

Chapitre II : Les protocoles de routage dans les RCSE_s

nœuds peuvent être différents dans d'autres réseaux.

- Aucune suggestion n'est faite à propos du temps de réélection des CHs (temps des itérations).

II.4.2.2. PEGASIS (Power-Efficient Gathering in Sensor Information Systems) [13]:

Il est considéré comme une optimisation de LEACH [13], proposée par Lindsey et autres en 2002 [14]; PEGASIS regroupe les nœuds du réseau sous forme d'une longue chaîne en se basant sur le principe qui stipule qu'un nœud ne peut communiquer qu'avec le nœud le plus proche de lui. De sorte que les nœuds reçoivent et communiquent juste des données des nœuds voisins. Les données collectées sont transmises d'un nœud à un autre jusqu'à l'arrivée à un nœud particulier qui va à son tour les transmettre vers la station de base. Ainsi, chaque nœud ajuste sa radio pour une communication très courte pour conserver son énergie. Pour communiquer avec le puits, le processus est organisé en rounds; au cours de chaque round un seul nœud est autorisé à communiquer avec le puits directement. Ce privilège est accordé à l'ensemble des nœuds du réseau à tour de rôle. Une meilleure conservation d'énergie est obtenue, également, en agrégeant les données sur chaque nœud du réseau.

Les nœuds particuliers sont choisis à tour de rôle selon la politique Round-robin pour le but d'économiser l'énergie. Contrairement au protocole LEACH, PEGASIS ne fixe pas le rôle à un seul nœud mais ils affectent d'autres rôles en fonction de leurs énergies restantes.

Dans PEGASIS chaque nœud nécessite une connaissance actualisée de la topologie pour s'informer de l'état énergétique de ses voisins afin de router ses données efficacement. Cela génère une surconsommation d'énergie importante surtout pour les réseaux à grande échelle.

En plus, PEGASIS stipule que tous les nœuds du réseau peuvent atteindre le puits ce qui nécessite une transmission réglable avec un surcoût énergétique non négligeable.

Chapitre II : Les protocoles de routage dans les RCSE_s

En plus, le délai de livraison des données est très important lorsque la chaîne formée est très longue. Et, le nœud qui transmet les données vers le puits peut devenir un point de congestion du réseau.

II.4.2.3. HEED (Hybrid, Energy-Efficient, Distributed approach):

Younes et Fahmy [15] ont proposé un algorithme de clustering distribué appelé HEED pour les réseaux de capteurs, il ne dépend pas de la topologie du réseau ni de sa taille mais il suppose que les capteurs sont intelligents (ils ont la possibilité de modifier leur puissance de transmission). HEED construit les cluster-Head selon un critère hybride regroupant le degré des nœuds et leurs énergies restantes. Il vise à réaliser une distribution uniforme des clusters dans le réseau et à générer des clusters équilibrés en taille. Dans HEED, un nœud n est élu comme cluster Head avec une probabilité P_{ch} égale à :

$$P_{ch} = C_{prob} * (E_n / E_{total})$$

Où:

E_n est l'énergie restante du nœud n , E_{total} l'énergie globale dans le réseau, et C_{prob} est le nombre optimal de clusters.

Dans HEED, les cluster-Head communiquent directement avec la station de base ce qui permet de minimiser la consommation d'énergie. Cependant, dans HEED, la topologie en clusters ne réalise pas la consommation minimale d'énergie dans les communications intra-cluster et les clusters générés ne sont pas tellement équilibrés en taille.

II.4.2.4. TEEN (Threshold-sensitive Energy Efficient sensor Network protocol) [16]:

Manjeshwar et Agrawal ont proposé une technique de clustering appelée TEEN [16] pour les applications critiques où le changement de certains paramètres peut être brusque tels que la température. L'architecture du réseau est basée sur un groupement hiérarchique à plusieurs

Chapitre II : Les protocoles de routage dans les RCSE_s

niveaux où les nœuds les plus proches forment des clusters. Puis ce processus de clustering passe au deuxième niveau jusqu'à ce que la station de base soit atteinte.

TEEN n'est pas bon pour les applications qui nécessitent des rapports périodiques

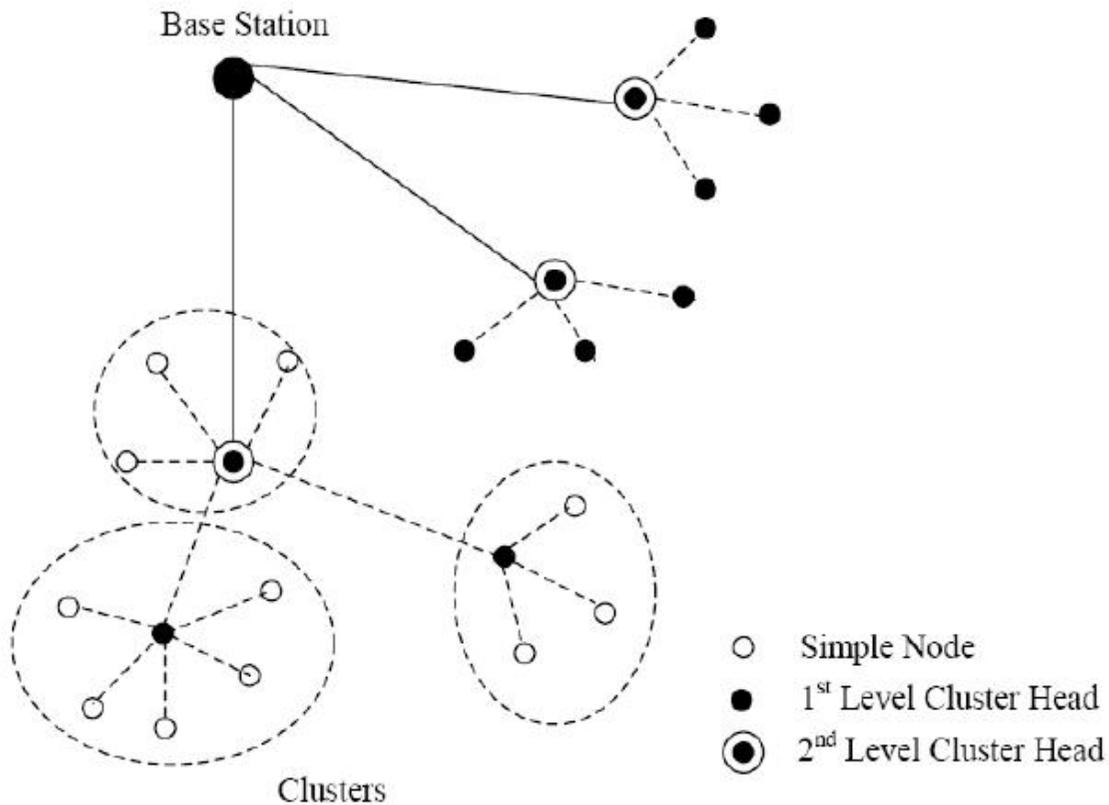


Fig.II.7.Fonctionnement du protocole TEEN [10].

II.4.2.5. APTEEN (Adaptive Threshold-sensitive Energy Efficient sensor Network protocol):

Pour remédier aux limitations du protocole TEEN, les auteurs ont proposé une extension de TEEN appelée APTEEN [17]. APTEEN est un protocole hybride qui change la périodicité et les valeurs seuils utilisées dans TEEN selon les besoins de l'utilisateur et le type d'application.

APTEEN est une extension de TEEN qui fait à la fois la collection des captures

Chapitre II : Les protocoles de routage dans les RCSF_s

périodique de données et qui réagit aux événements critiques. Quand la station de base forme des clusters, les clusters Head diffusent les attributs, les valeurs des seuils, ainsi que le calendrier de transmission à tous les nœuds.

Le cluster-Head effectue également l'agrégation de données afin d'économiser l'énergie.

II.4.2.6. SPEED [18]:

SPEED est un protocole de routage avec QoS portant sur l'utilisation des RCSF dans des applications temps-réel.

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

L'objectif de SPEED est de supporter un service de communication temps-réel mou (soft) avec une vitesse de relais bornée des paquets à travers le RCSF, de sorte que le délai de bout-en-bout soit proportionnel à la distance séparant la source de la destination (nœud collecteur) du paquet.

SPEED fournit trois types de services de communication temps-réel :

- 1) Unicast permettant à une zone du réseau détectant une certaine activité d'alerter une station de base distante,
- 2) Multicast dans une zone définie (*area-multicast*) donnant la possibilité à une station de base d'envoyer une commande ou requête à un secteur du réseau
- 3) Anycast dans une zone définie (*area-anycast*) permettant l'interrogation d'un seul nœud situé dans un secteur donné du réseau.

Pour son fonctionnement, SPEED maintient seulement l'information du voisinage immédiat du nœud. En se basant sur une table de position SPEED estime le délai sur chaque saut en calculant le délai d'aller-retour(en retranchant le temps de traitement coté récepteur). $HopDelay(i,j)=T_{rec} - T_{send} - T_{ack}$, où T_{rec} est temps réponse, T_{send} est le temps de l'envoi, T_{ack} est le temps du traitement du paquet d'acquittement (ACK : *Acknowledgment*) au niveau du

Chapitre II : Les protocoles de routage dans les RCSEs

récepteur .

Autrement dit il fournit une vitesse de relais uniforme des paquets à travers le réseau.

Il présente par ailleurs les caractéristiques suivantes :

- 1) l'emploi d'un mécanisme de contre-pression pour annoncer et faire éviter aux paquets les liens congestionnés du réseau.
- 2) l'utilisation d'une expédition non-déterministe pour assurer un équilibrage de charge entre chemins de routage concurrents.
- 3) la possibilité de passage à l'échelle du fait que toutes ses opérations sont distribuées et basées sur l'information locale au nœud.
- 4) la prise en charge des vides de routage.

Les conclusions de SPEED se basent aussi sur le fait que le réseau fonctionne de manière continue. Dans un cas réel, cela ne peut être le cas. En effet, les événements se produisent d'une manière aléatoire en temps et en espace et, par conséquent, des régions du réseau peuvent consommer plus d'énergie que d'autres. Du fait qu'il n'est pas chargé, un nœud candidat ayant une énergie minimale peut très bien être choisi comme nœud du prochain saut du paquet dans SPEED.

Chapitre II : Les protocoles de routage dans les RCSEs

II.5.Comparaison entre ces différents protocoles :

	Classification	Mobilité	Basé sur la négociation	Agrégation	Mise à l'échelle	Multi-chemin	L'utilisation d'énergie
SPIN	Plat	Possible	Oui	Oui	Limité	Oui	Limité
Diffusion Dirigée	Plat	Possible	Oui	Oui	Limité	Oui	Limité
LEACH	Hiérarchique	SB fixe	Non	Oui	Bonne	Non	Max
PEGASSIS	Hiérarchique	SB fixe	Non	Non	Bonne	Non	Max
TEEN, APTEEN	Hiérarchique	SB fixe	Non	Oui	Bonne	Non	Max
HEED	Hiérarchique	SB fixe	Non	Oui	Bonne	Non	Max

Tab. II.1. Comparaison entre ces différents protocoles.

Chapitre II : Les protocoles de routage dans les RCSEs

II.6.Conclusion :

Le routage dans les réseaux de capteurs est un problème complexe. Car nous devons assurer la fiabilité de livraison de données tout en consommant le minimum d'énergie. Plusieurs protocoles de routage ont été proposés pour les réseaux de capteurs dont l'objectif principal est de trouver la façon la plus simple d'acheminer les paquets de données d'une source vers une destination tout en économisant l'énergie.

La topologie plate du protocole de routage fournit de bonnes performances pour un réseau d'une petite taille. Mais lorsque la taille du réseau augmente, ces performances se dégradent vu le trafic de données qui augmente entre les nœuds, pour pallier à cette inconvénient, les chercheurs ont proposé l'architecture hiérarchique qui vise à augmenter la scalabilité du réseau et optimiser l'utilisation des ressources énergétique,

Dans ce chapitre nous avons résumé les protocoles de routage dans les réseaux de capteurs qui ont comme objectif commun le prolongement de la durée de vie du réseau et Nous nous sommes concentrés sur un de ces protocoles de routage hiérarchique qui est le protocole LEACH. Egalement dans ce chapitre nous avons classé les approches en 02 catégories principales : les protocoles hiérarchiques, et les protocoles data-centric. Nous avons choisi d'utiliser les protocoles hiérarchiques. Grâce à la création de grappes qui agrègent et fusionnent les données, le réseau devient capable de supporter une charge supplémentaire.

Conclusion générale

III.1. Introduction :

Le routage est un service très important dans les réseaux de capteurs. Il doit permettre l'arrivée des données à la station de base avec le minimum de pertes et de dissipation d'énergie.

La plupart des protocoles hiérarchiques n'adressent que le routage à un seul saut des chefs de groupes (cluster Head) à la station de base. En effet, ils supposent que les chefs de grappes peuvent communiquer directement avec la station de base. Cela devient impossible quand l'étendue du réseau augmente et que les chefs de groupes sélectionnés sont loin de la station de base.

Pour améliorer d'avantage la consommation d'énergie, et pour remédier à ce problème nous proposons le routage à plusieurs sauts qui consiste à utiliser un ou/et des nœuds intermédiaires pour l'acheminement des données.

III.2. Protocoles de routage à multi-saut:

Nous présentons ici trois protocoles qui adoptent une communication inter-Clusters à multi-sauts au lieu d'une communication directe entre les CHs et la BS.

III.2.1. Le protocole MCR (Multi-hop Clustering Routing Protocol): [19]

C'est un protocole de Clustering, baptisé Multi-hop Clustering Routing Protocol (MCR) il est proposé en 2011 par Q. Yang, Yuxiang ZHUANG, et Hui LI [19], basé sur l'utilisation des nœuds Gateway pour parvenir à une livraison de données vers une station de base éloignée avec un coût raisonnable en énergie.

Le protocole MCR utilise un principe de transmission inter-cluster à deux sauts, c'est-à-dire les nœuds CHs ne communiquent pas directement avec le puits, mais, ils utilisent un nœud intermédiaire (Gateway) qui se trouve dans une zone couverte par la station de base. Pour les

nœuds non couverts par le puits, le protocole propose d'utiliser le même principe que le protocole HEED pour la construction des clusters. Initialement, le puits envoie un message SinkMsg vers la zone de captage, les nœuds qui reçoivent ce message répondent par un message ChildReq qui indique leurs candidatures pour devenir nœud Gateway. Ensuite, le puits choisit les nœuds Gateway en répondant par un message d'acceptation. Les autres nœuds (non choisis par le puits comme Gateway) exécutent l'algorithme de Clustering relatif à HEED.

Une fois les nœuds CHs définis, chacun d'eux choisit son nœud Gateway en se basant sur le cout de la communication. Afin de garantir une meilleure utilisation des nœuds Gateway, l'algorithme exige qu'un nœud CH ne puisse communiquer qu'avec un et un seul nœud Gateway. Une fois la topologie construite, les nœuds simples envoient leurs données vers leurs ClusterHeads, qui les agrège et les envoie vers son nœud Gateway pour atteindre le puits.

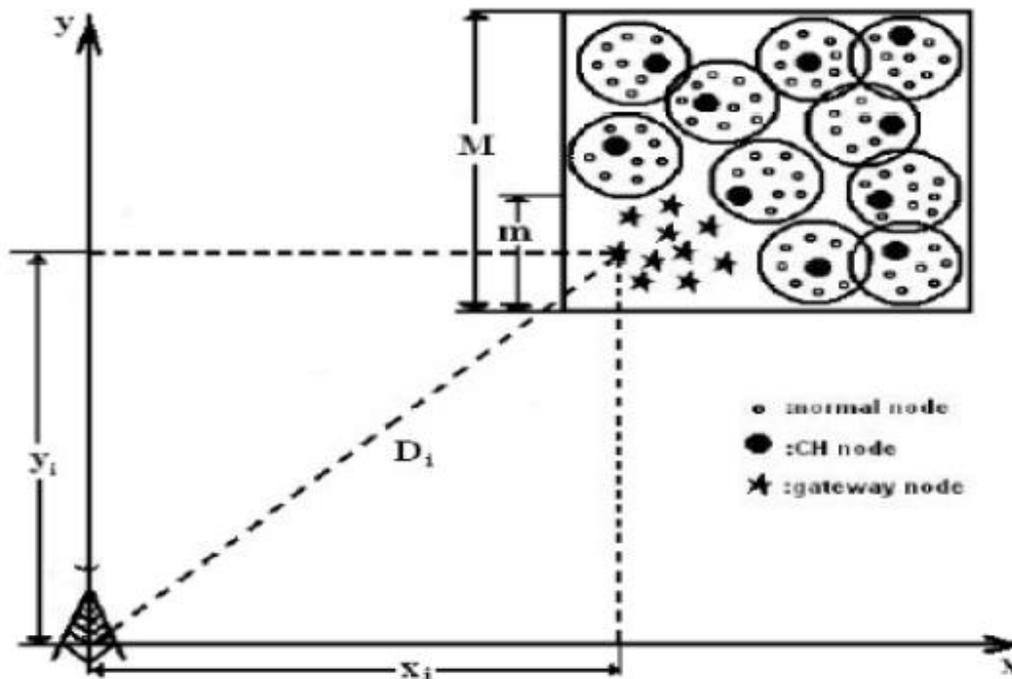


Fig.III.1.Exemple du protocole MCR [19].

D_i : distance entre la SB et les Gateway.

m : zone où se trouve les Gateway.

M : le champ de captage.

Le protocole MCR prolonge la durée de vie des nœuds CHs, et par conséquent celle du réseau, en adoptant une communication inter-cluster en deux sauts et une réélection basée sur l'énergie résiduelle. D'autre part, les nœuds Gateway sont déchargés de la phase Clustering ce qui limite leurs responsabilités à acheminer les paquets vers le puits. Cela minimise considérablement leurs dépenses énergétiques.

Le protocole MCR se focalise uniquement sur la communication inter-cluster et ignore celle de l'intra-cluster qui est responsable de la surconsommation en énergie. La solution hérite, également, les défauts du protocole HEED sur lequel elle se base, à savoir, le nombre important des clusters génères vu l'utilisation des communications intra-cluster a un saut.

III.2.2. Protocole LCH [20] :

Ce protocole est proposé en 2007 par Y. Wang et B. Ramamurthy, un protocole de Clustering basé sur une hiérarchisation en forme de couches dites LCH qui fixe comme objectif la conservation de l'énergie et la scalability du réseau WSN. Chaque couche contient un nombre prédéfini de Clusters sont construits en utilisant l'algorithme de Clustering du protocole LEACH. LCH vise la réduction de la consommation de l'énergie et le passage à l'échelle en adoptant une communication en plusieurs sauts, à travers les couches construites, entre les CHs et la BS.

Dans le protocole LCH les nœuds sont organisés en couches selon leurs distances, en nombre de sauts, par rapport au puits; chaque couche exécute un algorithme distribué LEACH-like ou un nombre défini de ClusterHeads est choisi. Ensuite, les nœuds non CH choisissent leurs clusters en se basent sur le coût de la communication. Les nœuds CH reçoivent les données envoyées par les membres de leurs clusters, les agrègent et les envoient vers la couche supérieure.

Une rotation des CH est effectuée pour équilibrer la consommation de l'énergie et garantir une durée de vie plus longue du réseau.

Les nœuds CHs sont choisis en se basant sur le calcul d'une probabilité p de la forme suivante:

$$P_j(t) = \begin{cases} \frac{K_i}{N_i - K_i \times (r \times \text{mod}(N_i/K_i))} & :C_j(t)=1 \\ 0 & :C_j(t)=0 \end{cases}$$

Ou:

K_i est le nombre de CH dans la couche i ;

N_i est le nombre totale des nœuds sur la couche i ;

r est le nombre de rounds;

$C_j(t) = 0$ si le nœud j a été désigné comme CH dans les rounds les plus récentes.

Une fois le CH est choisi, il annonce son état en diffusant un message avec le maximum de la puissance de transmission pour attirer des nœuds membres. Ensuite, il attend les messages d'adhésion émanant des autres nœuds.

Par la suite, chaque nœud non CH choisi aléatoirement son cluster, parmi les 3 meilleurs CHs qui nécessitent le minimum d'énergie de communication. Si un nœud reçoit des annonces provenant des CHs d'autres couches, il rejoint ceux qui appartiennent à la même couche que lui.

Les nœuds CHs recevant des annonces des couches supérieures mettent à jour leurs informations afin de parvenir à les rejoindre. Après que les messages d'adhésion des nœuds soient reçus, chaque CH crée et diffuse une planification TDMA pour les membres de son cluster.

Le protocole LCH permet des performances meilleures que LEACH en éliminant la contrainte de la possibilité de communication directe entre le puits et l'ensemble des nœuds du réseau, en adoptant une communication en multi-sauts.

L'organisation en couches permet de réduire le délai de livraison de données vers le puits car les données sont envoyées d'une couche inférieure vers une couche supérieure, en se rapprochant du puits.

Cependant, le protocole LCH ne garantit pas une distribution homogène des nœuds CHs sur le réseau car il se base sur l'algorithme LEACH. Il n'accorde aucune considération à l'énergie des capteurs lors du choix des nœuds CHs ou bien lors de leurs réélections.

LCH donne de meilleurs résultats lorsque le réseau est d'une densité plus importante.

III.2.3. Le routage multi-sauts MTE (Minimum Transmission Energy) [21] :

MTE est un protocole de routage à topologie plate, les nœuds capteurs routent les données à destination de la station de base à travers des nœuds intermédiaire, dans ce protocole tous les nœuds du réseau agissent comme des routeurs, les nœuds intermédiaires sont choisis de telle sorte que l'énergie de l'émission du paquet soit minimisée.

Pour router les données, un nœud choisit le prochain saut parmi ses voisins, en choisissant le plus proche de la station de base, de cette manière des routes seront construites. Quand un nœud reçoit des données de l'un de ses voisins en amont, il transmet ces données vers le voisin du saut suivant. Cela continue jusqu'à ce que les données atteignent la station de base.

Quand un nœud meurt, les voisins qui lui ont envoyés des données vont envoyer directement au prochain saut, de cette manière de nouvelles routes seront créées.

Les nœuds ajustent leur puissance d'émission au minimum nécessaire pour atteindre le prochain saut. Ceci réduit les interférences avec d'autres transmissions et réduit la perte d'énergie au niveau du nœud, et quand des collisions se produisent, les données seront supprimées.

Si le message envoyé traverse n nœuds pour arriver à la station de base, alors l'énergie totale consommée par ce message sera égale à n fois l'énergie à l'émission, plus n fois l'énergie de la réception.

Les nœuds qui sont proches de la station de base vont acheminer un grand nombre de paquets, ainsi ces nœuds vont s'éteindre rapidement, en d'autres termes plus le nœud est proche de la station de base plus il va mourir plus rapidement.

III.3.Synthèse des protocoles a multi-saut:

Nous avons présenté, trois protocoles qui adoptent une communication inter-Clusters à multi-sauts au lieu d'une communication directe entre les CHs et la BS. Nous avons présente les protocoles: LCH [20], MTE [21] et MCR [19] comme échantillon des protocoles à routage multi-sauts.

Avec le protocole LCH adopte une hiérarchisation du réseau en couches, et sur chaque couche un nombre prédéfini de Clusters sont construits en utilisant l'algorithme de Clustering du protocole LEACH. LCH vise la réduction de la consommation de l'énergie et le passage à l'échelle en adoptant une communication en plusieurs sauts, à travers les couches construites, entre les CHs et la BS.

Avec MTE les nœuds capteurs ne transmettent pas seulement leurs propres données, mais ils servent aussi de routeurs pour les autres nœuds capteurs, par conséquence, l'énergie des nœuds capteurs qui sont à proximité de la station de base est rapidement consommée tous les nœuds du réseau agissent comme des routeurs, les nœuds intermédiaires sont choisis de telle sorte que l'énergie de l'émission du paquet soit minimisée.

Avec le protocole MCR supposant que la station de base ne couvre que partiellement le champ de captage, il utilise les nœuds couverts par la BS comme Gateway lors de la livraison de données vers cette dernière. Les nœuds non couverts par la BS exécutent l'algorithme de Clustering du protocole HEED pour construire des Clusters à un saut. Le protocole MCR prolonge la durée de vie des nœuds CHs, et par conséquence celle du réseau, en adoptant une communication inter-cluster en deux sauts et une réélection basée sur l'énergie résiduelle. D'autre part, les nœuds Gateway sont déchargés de la phase Clustering ce qui limite leurs responsabilités à acheminer les paquets vers la BS. Cela minimise considérablement leurs dépenses énergétiques. Cependant, MCR hérite de HEED le nombre important de Clusters génères, ce qui le rend moins favorable au passage à l'échelle.

III.4.Conclusion :

Nous avons présenté dans ce chapitre l'approche du routage à plusieurs sauts à travers trois protocoles étudiés dans la littérature, nous avons opté pour l'adaptation de Leach au routage à plusieurs sauts ainsi les nœuds CHs ne communiquent pas directement avec le puits, mais, ils utilisent un nœud intermédiaire en se basant sur le principe utilisé dans le protocole MCR. Le résultat est appelé LEACH+. Pour tester l'efficacité et montrer les performances et les améliorations de l'approche proposée, nous avons réalisé une étude comparative avec le protocole de routage existant LEACH, les résultats de la simulation sont présentés dans le chapitre suivant.

» Chapitre IV «

*Etude de la solution
proposée.*

IV.1. Introduction:

Dans ce chapitre, nous allons présenter la conception de notre solution pour la conservation d'énergie dans les RCSF. Nous allons décrire l'objectif de notre solution et le mécanisme qu'on va utiliser pour l'atteindre. Nous présentons ensuite le simulateur que nous avons choisis et nous discuterons les résultats obtenus.

IV.2. Problématique et motivations :

Les contraintes imposées par les réseaux de capteurs sans fil ont poussé la communauté scientifique à proposer de nouvelles solutions pour minimiser l'énergie totale consommée par les capteurs tout en distribuant la consommation énergétique sur l'ensemble des capteurs. Notamment, la réalisation d'une méthode de communication à faible coût énergétique a été jusqu'à présent l'un des défis les plus importants dans les WSN (comparé à la réduction des délais dans les réseaux sans fil traditionnels). De nombreuses propositions ont ainsi vu le jour pour adresser ces problèmes.

Les techniques conçues pour les réseaux ad hoc traditionnels ne sont pas bien adaptées aux réseaux de capteurs. De nombreuses contraintes doivent être résolues pour les réseaux de capteurs sans fil. Nous nous intéressons, dans ce mémoire, aux contraintes posées par le routage et par la gestion de l'énergie dans ce type de réseaux pour prolonger la durée de vie de ces réseaux. Pour cela, notre objectif est de proposer un algorithme de routage basé sur la conservation de l'énergie, notre contribution est de concevoir un protocole de routage en prenant comme base le protocole Leach, dont l'objectif est de traiter le problème de gestion de ressources afin maximiser la durée de vie du réseau.

Dans ce protocole le réseau est divisé en clusters et chaque cluster possède un nœud maître appelé cluster-Head. Ce dernier communique directement avec la station de base ce qui permet de minimiser la consommation et réduire la quantité d'informations envoyées à la station

de base. Cependant, parmi l'une des critiques qui sont apportées au protocole LEACH : La possibilité de communiquer avec le puits à travers n'importe quel nœud du réseau exige une consommation d'énergie importante des nœuds lointains. Ce qui rend le protocole moins apte au passage à l'échelle.

IV.3. Notre objectif :

Notre objectif est de développer le protocole Leach dans le but de réduire la consommation d'énergie des nœuds dans un réseau de capteurs sans fil ainsi nous proposons d'utiliser le principe du protocole MCR basé sur l'utilisation des nœuds Gateway pour parvenir à une livraison de données vers une station de base éloignée avec un coût raisonnable en énergie. Ainsi les nœuds CHs ne communiquent pas directement avec le puits.

Initialement, le puits envoie un message **SinkMsg** vers la zone de captage, les nœuds qui reçoivent ce message répondent par un message **ChildReq** qui indique leurs candidatures pour devenir **nœud Gateway**.

Ensuite, le puits choisit les nœuds Gateway en répondant par un message d'acceptation. Les autres nœuds qui ne sont pas choisis par le puits comme Gateway exécutent l'algorithme de Clustering du protocole Leach.

Après la sélection des nœuds CHs chacun d'eux choisit son nœud Gateway en se basant sur le coût de la communication. Un nœud CH ne peut communiquer qu'avec un et un seul nœud Gateway.

Une fois la topologie construite, les nœuds simples envoient leurs données captées vers leurs ClusterHeads, qui les agrègent et les envoient vers son nœud Gateway pour atteindre le puits. La rotation des nœuds CH se fait sur la base du protocole Leach.

IV.4. Etude de la solution proposée:

IV.4.1.Choix de simulateur:

La simulation d'un système réel devient nécessaire dès lors que les modèles analytiques deviennent, soit trop complexes en termes de calcul et de temps de résolution. On fait recours à la simulation dans de nombreux cas par exemple quand on ne peut pas facilement observer les états du système ou quand le système réel est inobservable ou difficilement observable pour toutes sortes de raisons (dimension, sécurité, coût...)

Une simulation est le fait d'imiter sur un support informatique un phénomène physique, ce qui est possible à l'aide d'un simulateur.

Un simulateur est un dispositif technique permettant de reproduire de façon virtuelle un comportement d'un appareil en vue d'étudier les réactions de celui-ci.

Le type de simulation que nous avons utilisé est la simulation à événements discrets. Dans ce qui suit, nous décrirons le principe de fonctionnement de ce type de simulation.

4.1.1. La Simulation à événements discrets:

Une simulation à événements discrets [22] est une modélisation informatique où l'état d'un système est représenté par une séquence chronologique d'événements discrets (une suite d'évènement dans le temps). Chaque événement arrive à un instant donné et modifie l'état du système. C'est à dire que la simulation à événements discrets désigne la modélisation d'un système réel tel qu'il évolue dans le temps, par une représentation dans laquelle les grandeurs caractérisant le système (variables) ne changent qu'en un nombre fini ou dénombrable de points isolés dans le temps. Ces points sont les instants où se passent les événements.

➤ Exemples de simulateurs à événements discrets dédiés aux RCSFs :

Nous citons ici quelque exemples d'outils de simulation les plus populaires des réseaux de capteurs sans fils, ainsi nous présenterons celui que nous avons choisis d'utiliser pour tester et évaluer les performances de la solution proposé.

1. TOSSIM [23]:

TOSSIM est le simulateur dédié à TinyOS. **TinyOS [24]** est un système d'exploitation dédié aux réseaux de capteurs. TOSSIM essaye de tirer parti du mode d'exécution de TinyOS pour proposer un simulateur efficace et fiable. Le mode d'exécution de TinyOS est dirigé par les évènements, ce mode d'exécution se calque bien sur un simulateur a évènements discrets.

TOSSIM contient un modèle abstrait de chaque composant du matériel d'un nœud. Pour une simulation TOSSIM, on utilise le même code que celui destiné au nœud cible mais cette fois-ci TOSSIM simule le comportement du matériel en utilisant les modèles des composants. Simuler exactement le code qui tournera sur les nœuds permet de tester l'implémentation finale des algorithmes. Cette notion d'abstraction du matériel est tout à fait intéressante, cependant TOSSIM dans sa première version ne permet pas d'estimer l'énergie consommée.

2. PowerTOSSIM [25] :

PowerTOSSIM est l'extension de TOSSIM qui contient un modèle de consommation d'énergie. Pour les valeurs de consommation, les auteurs se sont basés sur leMica2 (nœud développé à l'université de Berkeley). Les auteurs connaissent les consommations des différents composants de ce nœud suivant leurs états. Il faut donc connaître l'état de chaque composant d'un nœud pendant la simulation. Grâce au modèle de simulation basé sur TinyOS, on connaît immédiatement l'état des composants autres que le microcontrôleur puisque les changements d'états correspondent à des événements dans TinyOS et donc dans TOSSIM. Plusieurs

composants du nœud sont parfois abstraits dans TOSSIM par un seul composant. L'estimation de la consommation du microcontrôleur est plus délicate : il faut instrumenter le code pour être capable de compter le nombre d'exécution de chaque bloc d'instruction, et il faut faire correspondre chaque bloc d'instruction avec son code en assembleur. Lors de la simulation, on note le nombre de passage, d'exécution de chaque bloc d'instructions. Sachant combien d'instruction élémentaire contient chaque bloc de base, on en déduit le nombre d'instructions effectuées par le microcontrôleur et donc sa consommation. Cette approche est intéressante mais elle ne permet pas de varier la précision du modèle de consommation. Enfin, les simulateurs TOSSIM et PowerTOSSIM ne conviennent que pour des applications écrites en TinyOS.

3. Aurora [26]:

Aurora est un simulateur pour réseaux de capteurs. Pour obtenir une simulation fiable, ce simulateur est cycle accurate ce qui signifie qu'il simule pour chaque nœud toutes les instructions qui s'exécutent sur ce nœud. Certes, cette approche offre une modélisation particulièrement précise mais on ne peut pas espérer qu'elle passe à l'échelle. Même dans le domaine des systèmes sur puce, une modélisation cycle-précis s'avère trop détaillée pour être utilisable en pratique. De plus, Aurora est écrit en Java et chaque nœud est implémenté par un thread (processus léger) Java ce qui impose une difficulté supplémentaire : il faut synchroniser les threads pour s'assurer qu'un nœud ne reçoive pas un paquet avant que son voisin ne l'ait envoyé.

4. WSNNet [27]:

WSNNet est un simulateur de réseaux de capteurs à événements discrets. Il se différencie des autres simulateurs dans sa capacité à modéliser très précisément le medium radio sans toutefois empêcher la simulation de réseaux de plusieurs centaines de nœuds. WSNNet fonctionne sous Linux et il a l'avantage d'être gratuit

5. J-Sim [28]:

J-Sim, autrefois connu sous le nom de JavaSim, est un simulateur réalisé en Java. Il offre une bibliothèque de simulation de réseaux de capteurs; il est conçu suivant une approche orientée composants appelée *autonomous component architecture (aca)*. ces composants communiquent les uns les autres via l'envoi/réception de données. Chaque composant est représenté par une classe. Un composant est une entité indépendante qui représente un objet physique (une batterie, une couche logicielle, un module radio...) ou logique (un protocole de routage, un module d'énergie, un module de mobilité...).

Son architecture se subdivise en trois types de composants : un premier pour contenir la représentation des nœuds, un deuxième pour la représentation du canal de captage et un troisième pour la représentation du canal radio.

La simulation du fonctionnement du réseau exige la définition des composants, leur mise en relation est réalisée par le **tcl** il s'agit d'un langage de script qui est complètement supporté par le simulateur J-Sim il permet de spécifier l'architecture du réseau et les paramètres de simulation.

6. OPNET [29]:

OPNET (*Optimum NETWORK Performance*) est un simulateur à événements discrets. C'est un outil très puissant pour la simulation et l'évaluation de performance de réseaux. Il permet aussi à l'utilisateur de construire ses propres modèles des plus simples aux plus complexes. Il possède trois niveaux d'abstraction pour construire les modèles. Les trois niveaux d'abstraction sont : le réseau, les nœuds, les processus.

Le réseau est composé de plusieurs nœuds, le nœud est composé de plusieurs processus et le processus est décrit avec un automate et du code en C.

OPNET permet de décrire les processus à l'aide des automates et d'intégrer ces processus dans les nœuds formant un réseau ou un dispositif informatique. Le langage de base d'OPNET est le C.

Ses avantages:

Il permet de concevoir et d'étudier des réseaux de communications, des nouvelles technologies, des protocoles et des applications avec facilité et évolutivité.

Son inconvénient :

- il est payant mais ce problème est résolu avec la version académique. OPNET (version d'apprentissage : IT Guru 9.0)
- apprentissage long.

7. OMNET [30]:

1. présentation:

Le simulateur OMNET++ est un simulateur à événements discrets dans lequel les différents éléments du réseau communiquent par envoi de messages.

OMNET++ est un projet open source dont le développement a commencé en 1992 par Andras Vargas à l'université de Budapest. C'est une bibliothèque de simulation écrite en C++ pour construire des simulateurs de réseaux au sens large, c-à-d. réseaux filaires et sans fils, mais également des réseaux internes aux machines (BUS de processeur par exemple).

OMNET++ se distingue par son orientation objet et l'utilisation de modules hiérarchisés qui permet une grande flexibilité dans la création de nœuds complexes au sein du réseau. Il peut donc être utilisé pour :

- La modélisation de trafic de réseaux de communication.
- La modélisation de protocoles.
- La modélisation de réseaux de files d'attente et d'autres systèmes distribués.

L'architecture d'OMNET++ est hiérarchique composé de modules. Un module peut être soit **module simple** ou bien un **module composé**. Pour chaque module simple correspond un fichier **.cc** et un fichier **.h**. Un module composé est composé de simples modules ou d'autres modules composés connectés entre eux.

La communication entre les différents modules se fait à travers les échanges de messages. Les messages peuvent représenter des paquets, des trames d'un réseau informatique, des clients dans une file d'attente ou bien d'autres types d'entités en attente d'un service. Les messages sont envoyés et reçus à travers des ports qui représentent les interfaces d'entrer et de sortie pour chaque module.

La conception d'un réseau se fait dans un fichier **.ned** et les différents paramètres de chaque module sont spécifiés dans un fichier de configuration (**.ini**). OMNET++ génère à la fin de chaque simulation deux nouveaux fichiers **omnet.vec** et **omnet.sca** qui permettent de tracer les courbes et calculer des statistiques.

La vue modulaire nous donne la possibilité de modifier des modules déjà implémentés ou d'implémenter des modules indépendants et les intégrer par la suite dans la même architecture en définissant les interfaces de communication entre ces modules.

Vu les avantages que présente le simulateur OMNET++, nous l'avons choisi comme environnement de simulation pour tester et évaluer les performances du protocole que nous avons proposé.

IV.5. Implémentation et simulation sous OMNET:

IV.5.1. Paramètres de la simulation:

IV.5.1.1. modèle de simulation implémenté:

OMNET est basé sur une architecture modulaire flexible, ainsi un modèle OMNET++ consiste en un ensemble de modules hiérarchiquement emboîtés communiquant entre eux par échange de messages.

Afin d'exploiter cette architecture, le réseau est considéré comme un module composé de plusieurs autres modules, la figure (**Fig.IV.3.**) illustre l'architecture du réseau selon le modèle de simulation implémenté.

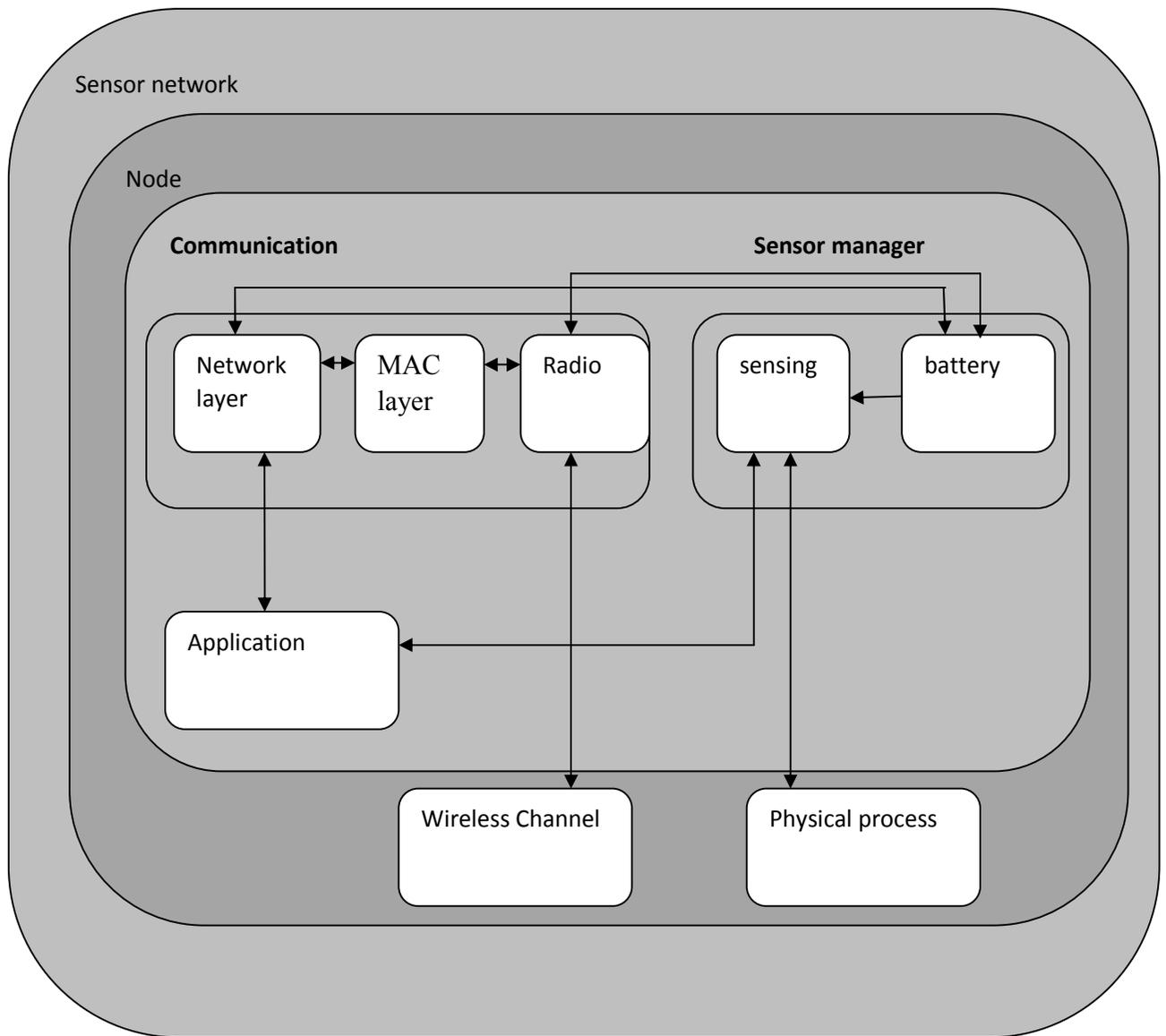


Fig.IV.1. Architecture du réseau selon le modèle de simulation implémenté.

Le module composé **sensorNetwork** comprend :

- les modules simples **wirelessChannel** ,**physicalProcess**.
- le module composé **Node**.

Le module simple wirelessChannel représente le canal de transmission sans fil.

Dans cette architecture les nœuds ne sont pas directement connectés entre eux, mais ils communiquent via ce module simple **wirelessChannel** qui décide du nœud vers lequel le message doit être acheminé selon les informations contenues dans l'entête du message et avec quelle puissance ce dernier doit être acheminé.

Le module simple physicalProcess représente les processus physique et/ou les objets cibles à contrôler .afin de simuler la capture, tous les nœuds sont connectés à ce module.

Le module composé Node représente les capteurs et la station de base.

Les sous modules composant le module Node:

- **module application** : permet de traiter l'information capturée avant de la transmettre à la pile de communication, ce module modélise les applications utilisateur exécutées au niveau du capteur.
- **module communication**: il représente les couches de communication située au-dessous de la couche application dans la pile protocolaire et qui sont :

La couche réseau (**netLayer**), la couche Mac (**macLayer**) et la couche physique (**radio**):

macLayer: simule la couche Mac dans un capteur, dans notre application son rôle est résumé au transfert de message entre la couche physique et la couche réseau.

netLayer: simule les protocoles de routage implémentés au niveau de la couche réseau .donc c'est au niveau de cette couche que sera implémenté notre protocole.

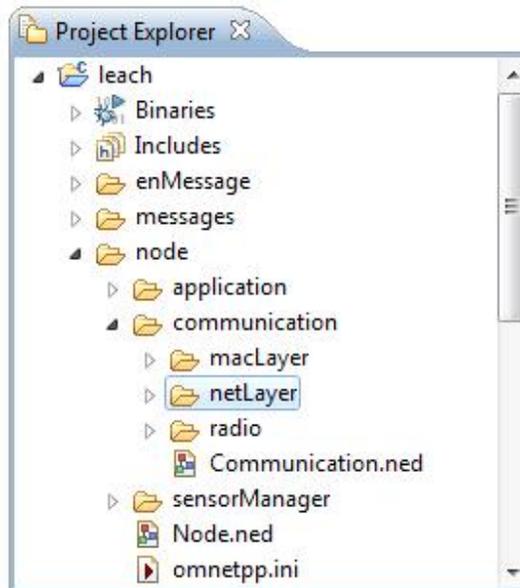


Fig.IV.2. la couche réseau ou sera implémenté notre protocole.

radio: simule un dispositif caractérisant les propriétés sans fil .elle représente la couche physique et l'unité de transmission/réception.

- **module sensorManager:** module composé il représente le plan de gestion de taches de sensation (qui est le module sensing) il représente le module de capture au niveau d'un nœud. et la gestion d'énergie (qui est le module battery Manager) son rôle consiste à mettre à jour l'énergie disponible au niveau du nœud, il permet de déconnecter le nœud du réseau si l'énergie de ce dernier est complètement dissipée.

Création de nouveau projet sous OMNET:

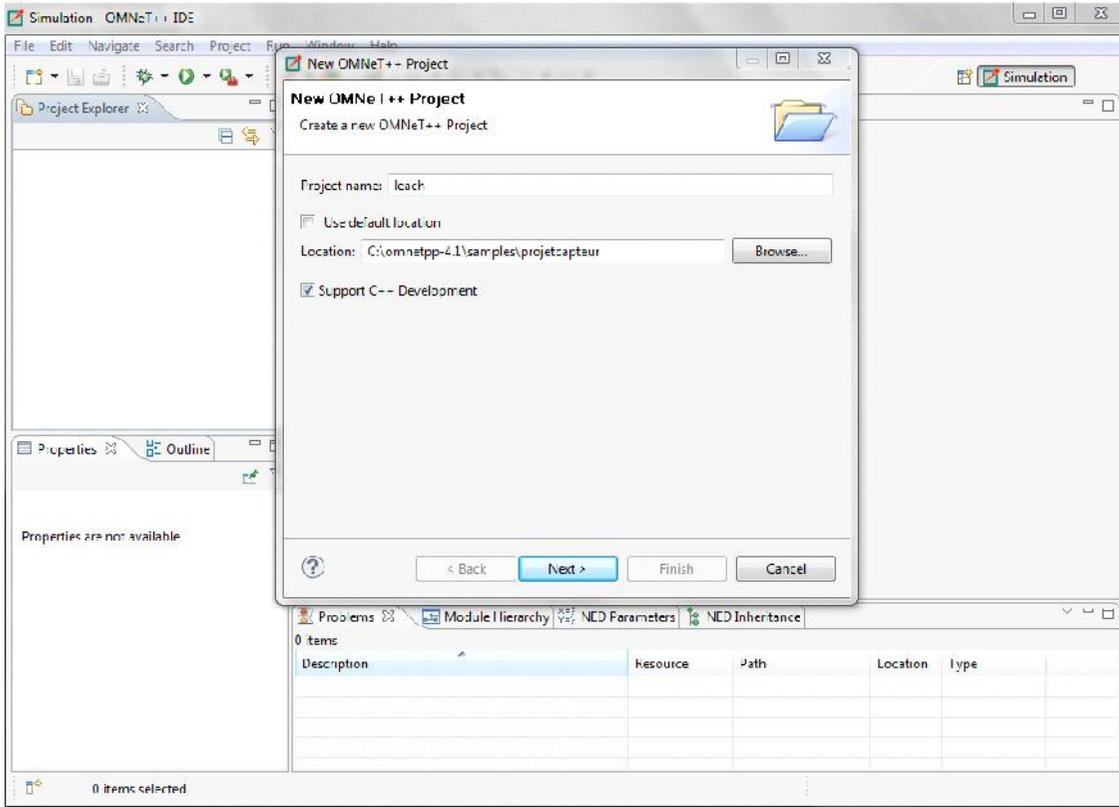


Fig.IV.3. Nouveau projet sous OMNET.

Après la création du nouveau projet nommé Leach on aura la figure suivante :

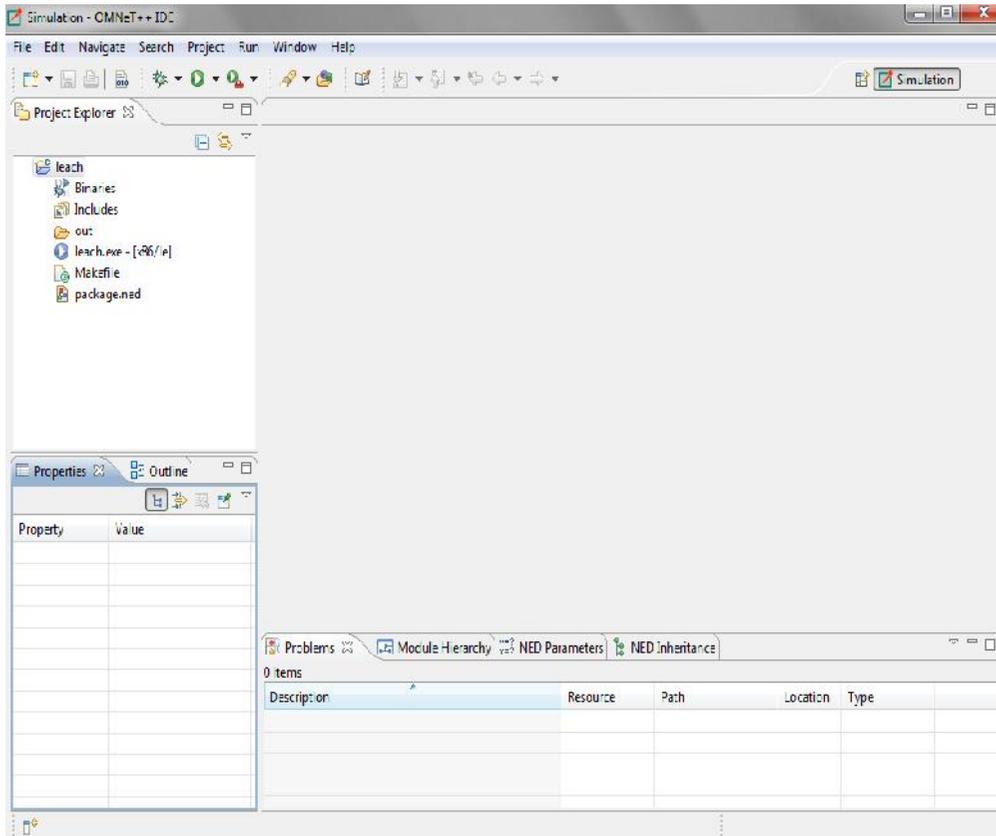


Fig.IV.4. Nouveau projet nommé Leach sous OMNET.

Cette figure représente le protocole Leach sous OMNET:

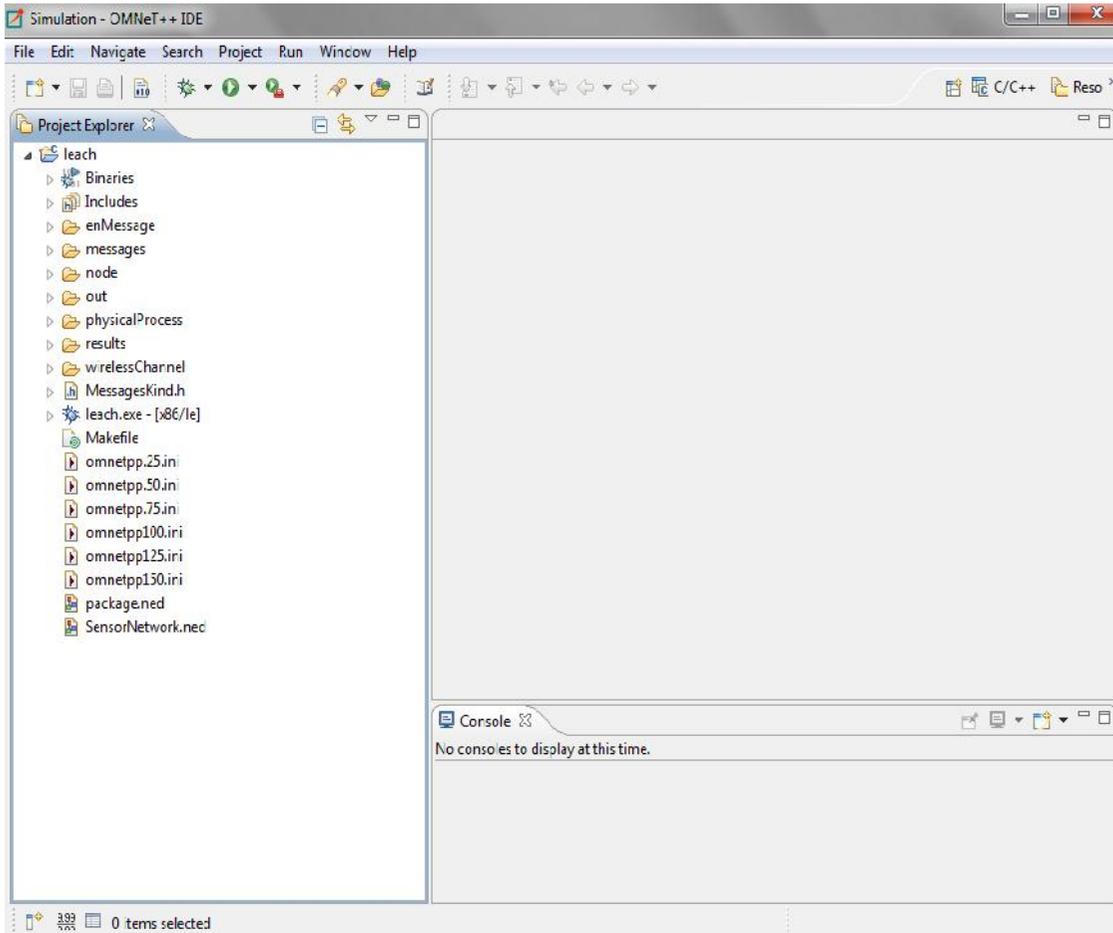


Fig.IV.5.le protocole Leach sous OMNET.

IV.5.1.2. Type de messages échangés:

Type de paquet	Taille de l'entête du paquet	Fonction
NewRoundMsg	32	Envoyé par la station de base pour lancer le nouveau round.
EnergyUpdate	80	Paquet envoyé par tous les nœuds pour diffuser leur energie.
ChAdv	16	Message d'annonce envoyé par les CH pour informer les autres nœuds de son statut et former son cluster.
JoinClusterMsg	32	Message envoyé par les nœuds vers le CH pour l'informer qu'ils désirent joindre son cluster.
SetUpMsg	176	Initié par la SB et relayé par les CHs pour échanger le coût des chemins vers la SB et construire les tables de routage.
DAtaMsg	128	Paquet de données pour les transmissions entre les nœuds membre et leur CHs, inter-cluster et entre les CHs et la SB

Tab.IV.1. messages échangé dans LEACH.

IV.5.1.3.Scénario utilisé:

Paramètres	Valeurs
Nombre de nœuds	25, 50, 75, 100, 125,150
Durée de simulation	100s
Taux de paquet généré par seconde	1/s
Nombre de station de base	1
Energie initiale	1000mj
Seuil de réception du signal	-98dbm
Taux de transmission	19.66kps
La probabilité d'être CH (P)	0.5

Tab.IV.2.les paramètres et valeurs utilisés sous OMNET.**IV.6. Résultats:**

Dans cette section, nous allons présenter les résultats que nous avons obtenus jusqu'à maintenant selon ces différents critères de performances :

IV.6.1. Critères de performances:

Pour la comparaison de notre protocole avec le protocole Leach, nous prenons en compte les métriques suivantes:

- **Moyenne d'énergie consommée:**

$$\text{m.e.c} = \frac{\sum E_i (i=1, n)}{n}$$

E_i: l'énergie consommée pour n capteur i, elle représente la différence entre l'énergie initiale et son

énergie résiduelle.

n: le nombre de capteurs dans le réseau.

- **Durée de vie du réseau:**

L'objectif du développement de notre protocole est de prolonger la durée de vie du réseau. Cette dernière est définie par le temps écoulé avant que tous les nœuds du réseau épuisent leur énergie.

- **Ecart type de la consommation:**

$$\text{Ecartype} = \sqrt{\frac{\sum_{(i=1, n)} (\text{m.e.c} - E_i)^2}{n}}$$

m.e.c: la moyenne de l'énergie consommée.

E_i : l'énergie consommée par le nœud i du réseau.

n : nombre de nœuds du réseaux.

IV.5.2. Discussion des résultats:

Dans cette section nous allons analyser les résultats des simulations réalisées suivant les critères de performance discutés précédemment :

- **Moyenne d'énergie consommée:**

Le tableau (Tab.IV.3.) représente les résultats de la moyenne d'énergie consommée (**m.e.c**) obtenus par l'énergie consommée pour n capteur i (E_i)/ le nombre de capteurs n .

Protocoles Nombre de Nœuds	LEACH	LEACH+
25	56.622	54.567
50	83.898	74.683
75	106.211	83.235
100	119.608	97.232
125	130.455	120.395
150	138.396	129.315

Tab. IV.3.Moyenne d'énergie consommée.

Le graphe de la figure (Fig.IV.6) représente les résultats de la moyenne d'énergie consommée (m.e.c) en fonction du nombre de nœuds du réseau. Nous pouvons remarquer sur le graphe que le protocole Leach + consomme moins d'énergie en le comparant avec le protocole LEACH de base.

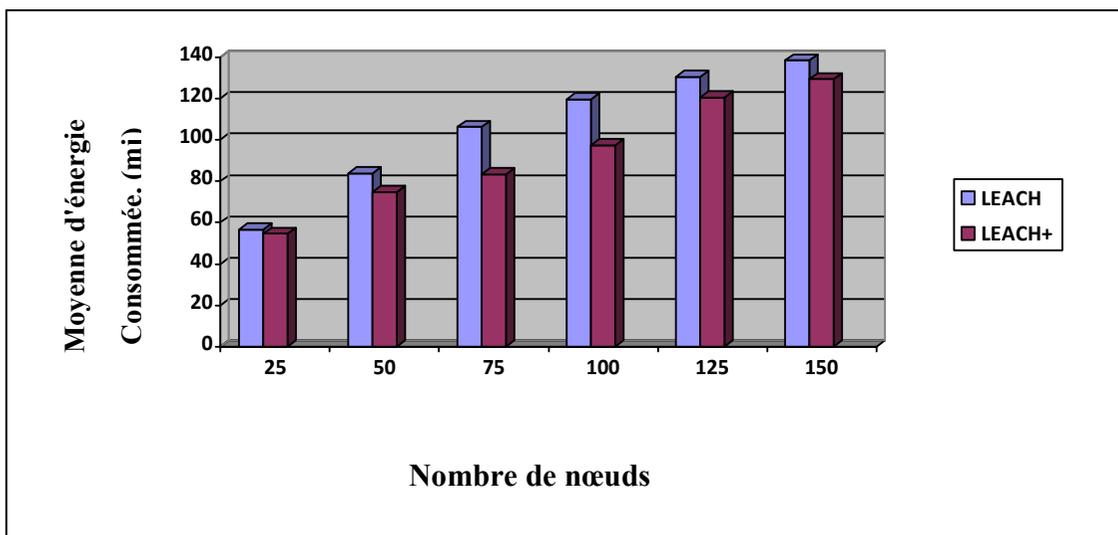


Fig.IV.6.Moyenne d'énergie consommée.

- **Durée de vie du réseau:**

Le graphe de la figure (Fig.IV.7.) illustre la durée de vie du réseau au niveau des deux protocoles. Le protocole LEACH+ présente une amélioration considérable de la durée de vie du réseau par rapport au protocole LEACH.

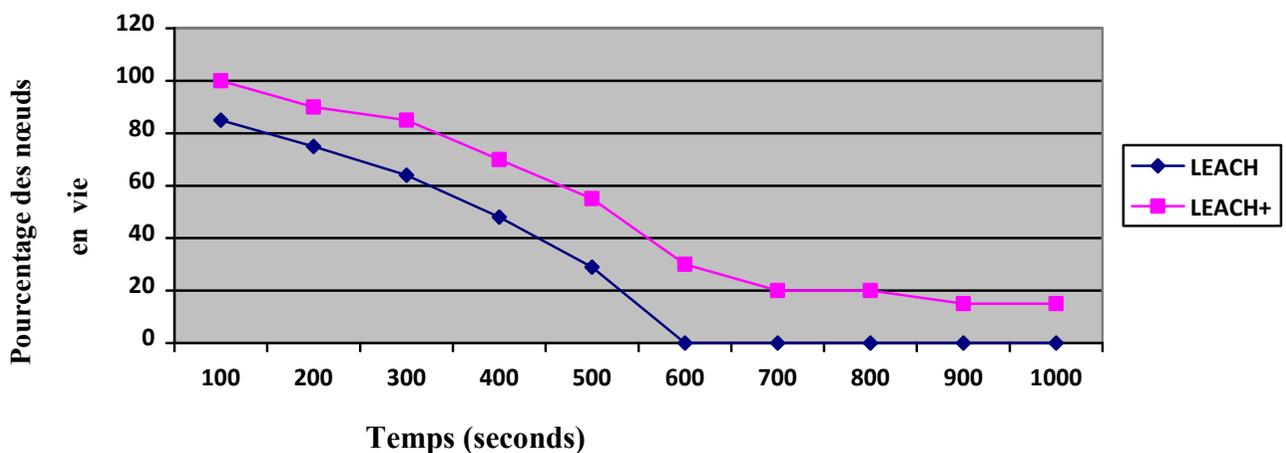


Fig.IV.7. Durée de vie du réseau.

- **Ecart type de la consommation:**

L'écart type sert à mesurer la dispersion, ou l'étalement, d'un ensemble de valeurs autour de leurs moyennes.

Dans notre cas nous avons calculé l'écart type de la moyenne de l'énergie consommée pour évaluer l'équilibrage de la consommation de l'énergie dans le réseau.

Protocoles Nombre de nœuds	LEACH	LEACH+
25	19.0163	17,514
50	23.1448	18,131
75	25.016	16,594
100	29.900	16,204
125	25.6811	19,600
150	28.857	20,881

Tab. IV.4. Ecart type de l'énergie consommée.

Le graphe de la figure (Fig.IV.8.) représente les résultats de l'écart type en fonction du nombre de nœuds du réseau. Nous constatons que l'écart type de l'énergie consommé est plus important dans le protocole LEACH comparé au protocole LEACH+. Ceci revient au nœud ch plus loin de la station de base qui consomme plus d'énergie tandis que dans LEACH+, d'un côté l'énergie est de plus en plus équilibré, d'un autre cote les chs communique avec la station de base a travers un nœud intermédiaire Gateway ceci s'explique par la bonne répartition des tâches sur les différents nœuds.

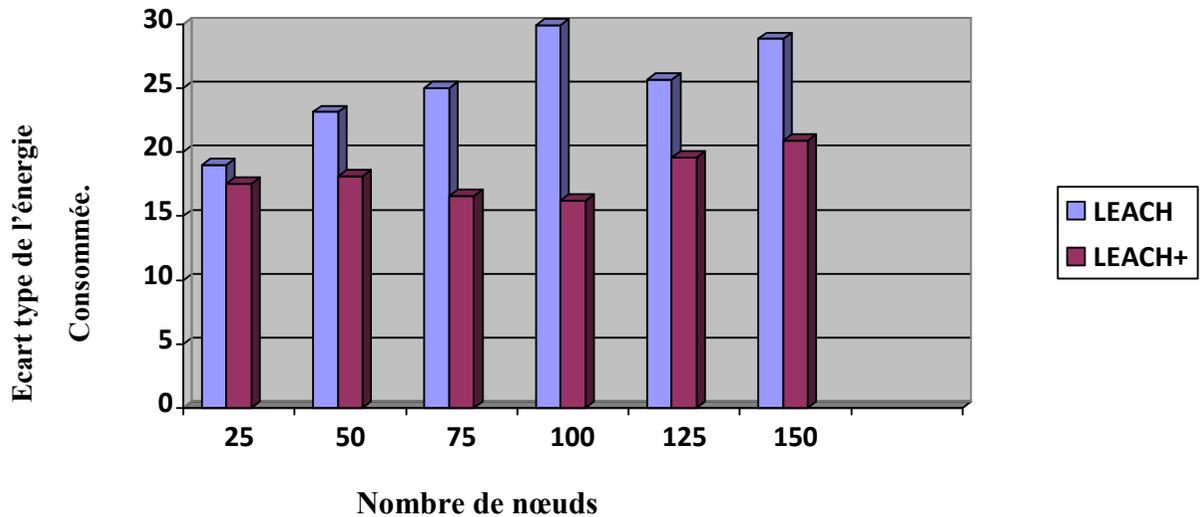


Fig.IV.8. Ecart type de l'énergie consommée.

IV.7. Conclusion:

Dans ce dernier chapitre, nous avons donné un aperçu sur les simulateurs qui existe dans la littérature, nous avons présenté l'environnement de simulation que nous avons choisi OMNET++4.1., son principe de fonctionnement ainsi que la plateforme de simulation utilisée. Nous avons implémenté le protocole LEACH et le protocole LEACH+

Nous avons effectué une étude comparative entre les deux protocoles selon différents métriques les résultats des tests qu'on a obtenus jusqu'à maintenant ont montré que notre solution prolonge la durée de vie de réseau. En effet grâce au routage multi-saut, si le nombre des nœuds du réseau augmente et si les CHs sélectionnés sont situés plus loin de la station de base, les CHs agrègent et acheminent les données vers la station de base à travers des nœuds intermédiaires ceci réduit les consommations de leurs énergie ainsi la durée de vie du réseau augmente.

Conclusion générale

Conclusion générale :

La minimisation de la consommation d'énergie des capteurs et le prolongement de la durée de vie du réseau étaient les objectifs principaux de ce travail, en vue de réaliser ceci, Nous avons présenté les différents Facteurs intervenants dans la consommation d'énergie et les solutions de routage existantes selon une classification par rapport à la topologie du réseau utilisée. Nous avons focalisé, par la suite, notre étude sur la classe des protocoles de routage hiérarchique qui garantit une optimisation de la consommation en énergie. En effet l'organisation du réseau de capteurs en groupes (clusters), permet une distribution de la consommation de l'énergie sur le réseau et une spécification des tâches selon le type de nœud tel que les CHs se chargent de la dissémination et de l'agrégation des données vers la station de base, alors que les autres nœuds se chargent de la collection de données et la dissémination locale vers les CHs.

Quand l'étendue du réseau augmente, le routage hiérarchique devient incapable d'assurer l'arrivée des données des nœuds éloignés à la station de base. Dans ce mémoire, nous avons adapté un protocole hiérarchique au routage à plusieurs sauts afin de résoudre ce problème.

L'évolution des performances de notre protocole a été réalisé par simulation, en utilisant l'environnement OMNeT++4.1. Ce dernier représente un outil modulaire d'une grande flexibilité qui facilite la modélisation de n'importe quel système. Les tests effectués sur notre protocole selon différentes métriques ont prouvé son efficacité en termes de conservation de l'énergie des capteurs ce qui permis d'étendre la durée de vie du réseau.

Notre solution est d'utiliser un nœud intermédiaire entre les CH et SB, donc Comme perspectives pour notre travail, nous envisageons d'étendre notre solution à plusieurs nœud intermédiaire pour économiser encore plus l'énergie des nœuds capteurs, nous envisageons d'apporter une amélioration sur le mécanisme d'élection des nœuds CH, en se basant sur l'optimisation des nombres de clusters formés.

Références

Référence :

- [01]:** S. Zhuang and C. G. Cassandras. "Optimal dynamic voltage scaling for wireless sensor networks with real-time constraints". *Conf. on Intelligent Systems in Design and Manufacturing VI*, October 2005.
- [02]:** Paolo Santi. "Topology Control in Wireless Ad Hoc and Sensor Networks", *Hardcover*, july 2005.
- [03]:** W. Ye, J. Heidemann, and D. Estrin. "An Energy-Efficient MAC Protocol for Wireless Sensor Networks". pp 1567-1576, June 2002.
- [04]:** S. Kumar, D. Shepherd, and F. Zhao. "Collaborative signal and information processing in micro-sensor networks". *IEEE Signal Processing Magazine*, March 2002.
- [05]:** Paolo Santi. "Topology Control in Wireless Ad Hoc and Sensor Networks", *Hardcover*, july 2005.
- [06]:** S.Narayanaswamy, V. Kawadia, R.S. Sreenivas, and P.R. Kumar. "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the Compowprotocol", *European Wireless Conference*, 2006.
- [07]:** H. Namgoog, D. Lee, and D. Nam. "Energy efficient topology for wireless microsensor networks". *ACM, PE-WASUN*, October 2005.
- [08]:** K. Sohrabi and J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Communications*, Vol. 7(5), pp. 16–27, October 2000.
- [09]:** J. N. Al-Karaki and A. E. Kamal. "Routing Techniques in Wireless Sensor Networks: A Survey". Dept. of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011, 2003.
- [10]:** Yacine CHALLAL « réseau de capteur sans fil >>, support de cours, 17/11/2008.
- [11]:** Master Informatique ; Spécialité Systèmes Informatiques ; DEMRI Mohammed, cours 2013/2014.
- [12]:** M.Ali ET S.k.Ravula, "Real-time support and energy efficiency in wireless sensornetworks". technical report, IDE0805, january 2008.
- [13]:** Luis Javier Garcia VILLALBA, Ana Lucila Sandoval OROZCO, Alicia Trivino CABRERA, Claudia Jacy Barenco ABBAS, Routing Protocols in Wireless Sensor

Networks , Sensors 2009, 9, 8399-8421, ISSN 1424-8220, 2009.

[14]:S. LINDSEY, C. RAGHAVENDRA, PEGASIS: Power-Efficient Gathering in Sensor Information Systems , IEEE Aerospace Conference Proceedings, Vol. 3, 9-16 pp.1125-1130, 2002.

[15]: O. Younis, S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing 03 (4). 2004, pp. 366–379.

[16]: Manjeshwar, D.P. Agrawal. TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. Proceedings 15th International Parallel and Distributed Processing Symposium. 2001,

[17]: Manjeshwar, Arati and Agrawal, Dharma P. APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. IPDPS '02, 16th International Parallel and Distributed Processing Symposium. 2003, p. 48.

[18] : Thèse Doctorat Spécialité :Informatique Option :Informatique mobile présenté par : Mohamed AISSANI thème : Optimisation de routage dans les réseaux de capteurs pour les applications temps reel, 2001.

[19]: Qi YANG, Yuxiang ZHUANG, Hui LI, An Multi-hop Cluster Based Routing Protocol for Wireless Sensor Networks , Journal of Convergence Information Technology, Volume 6, Number 3, Mars 2011.

[20]: Yong WANG, Byrav RAMAMURTHY, Layered Clustering Communication Protocol for Wireless Sensor Networks , Proceedings of 16th International Conference on Computer Communications and Networks, 844–849, 2007.

[21]: M. Ettus, "System Capacity, Latency, and Power Consumption in Multihop-Routed SS-CDMA Wireless Networks", In Radio and Wireless Conference (RAWCON '98), pp. 55-58. (1998)

[22]:http://fr.wikipedia.org/wiki/Simulation_%C3%A0_%C3%A9v%C3%A9nements_discrets.

[23]: Philip Levis, Nelson Lee, MattWelsh, and David Culler. Tossim : accurate and scalable simulation of entire tinyos applications. In SenSys '03 : Proceedings of the 1st international conference on Embedded networked sensor systems, pages 126–137, New York, NY, USA, 2003. ACM Press.

[24]: TinyOS, 2001. <http://www.tinyos.net/>.

[25]: Victor Shnayder, Mark Hempstead, Bor rong Chen, Geoff Werner Allen, and Matt

Welsh. Simulating the power consumption of large-scale sensor network applications. In SynSys, pages

[26]: Ben L Titzer, Daniel K Lee, and Jens Palsberg. Avrora : Scalable Sensor Network Simulation with Precise Timing. Proceedings of IPSN, 2005. 188–200, 2004.

[27]: Simulation du protocole de routage binary waypoint sur WSNNet /Travaux d'études et de recherche Par Selma BELHADJ AMOR .

[28] : Jsim: <http://www.j-sim.org/>

[29]: "OPNET Modeler". Software. <http://www.opnet.com/products/modeler/home.html>, 2006.