

N° D'ORDRE

REPUBLIC ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE D'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE  
« HOUARI BOUMEDIENE »  
FACULTE D'ELECTRONIQUE ET D'INFORMATIQUE

# THÈSE

Présentée pour l'obtention du diplôme de

**MAGISTER**

En : INFORMATIQUE

Spécialité : Programmation et Systèmes

Par : ZEGHILET HOUDA

SUJET

*La Diffusion Dirigée avec Conservation d'Énergie dans les Réseaux de Capteurs*

Soutenu le 13/12/2006, devant le Jury composé de :

Mr. AHMED NACER M.	Professeur, USTHB	Président
Mr. BADACHE N.	Professeur, USTHB	Directeur de thèse
Melle. BENSAOU N.	Maître de conférence, USTHB	Examineur
Mr. NOUALI O.	Docteur, CERIST	Examineur
Mr. TANDJAQUI D.	Chargé de recherche, CERIST	Examineur

*Je dédie ce travail à tous ceux qui me sont chers...*

*Ma mère;*

*Mon père;*

*Mes frères et sœurs;*

*Tous mes amis.*

*Houda*

## **Remerciements**

*Je tiens à remercier mon Dieu, le tout puissant, de m'avoir donné le courage et la patience jusqu'à l'achèvement de ce travail.*

*J'exprime ma profonde reconnaissance et mes vifs remerciements à mon directeur de thèse Pr Nadjib BADACHE, de m'avoir fait confiance en me proposant ce sujet. Je le remercie également pour ses lectures attentives et pour ses critiques et suggestions qui ont été d'un grand apport pour la finalité de ce travail.*

*Je remercie M<sup>r</sup> M. Ahmed Nacer, M<sup>elle</sup> N. Bensaou, M<sup>r</sup> O. Nouali et M<sup>r</sup> D. Tandjaoui d'avoir accepté de juger ce travail.*

*J'adresse également mes sincères remerciements à Madame NADIA NOUALI, responsable du laboratoire des Systèmes Informatiques (C.E.R.I.S.T), pour sa compréhension, et pour tous les moyens qu'elle a mis à ma disposition.*

*Un grand MERCI, aux membres de ma famille et à mes très chers amis pour leurs encouragements, leur patience et leur amour.*

## **Résumé :**

La technologie des réseaux de capteurs est une technologie clé pour les années futures. Des composants d'un coût raisonnable, intelligents (capacités de traitement), avec plusieurs capteurs à bord, interconnectés à travers des liens sans fil sont déployés en grand nombre dans différents endroits. Cette interconnexion a un effet profond et des opportunités sans précédents dans beaucoup d'applications militaires et civiles.

Pour assurer la communication entre les nœuds dans un réseau de capteurs, plusieurs protocoles de routage ont été proposés. Une nouvelle approche de routage orienté données est apparue pour ce type de réseaux. La diffusion dirigée est un exemple de protocole de routage orienté données. Il applique une identification des données sous forme de paires attribut/valeur et utilise des traitements intra-réseau (l'agrégation, le cache, ...etc.) dans le routage de données.

En l'absence d'un système d'adressage global dans les réseaux de capteurs, les protocoles de routage, notamment, la diffusion dirigée font recours au mécanisme d'inondation. Ce dernier est très coûteux en terme de consommation d'énergie. Celle-ci constitue, dans les réseaux de capteurs, une ressource limitée qui doit être prise en compte dans tous les niveaux de l'architecture.

Dans cette thèse, nous avons proposé une amélioration pour le paradigme de la diffusion dirigée en vue de le rendre plus efficace en consommation d'énergie. Pour cela, une description de la diffusion dirigée est d'abord donnée décrivant ses caractéristiques et son principe de fonctionnement. Par la suite, une étude des caractéristiques de dissipation d'énergie vient mettre en évidence les niveaux de consommation et de conservation d'énergie dans les réseaux de capteurs.

Après ce constat, un état de l'art des travaux de conservation d'énergie dans le paradigme de la diffusion dirigée est élaboré. Cet état de l'art présente l'ensemble des solutions proposées dans ce domaine ainsi que des discussions décrivant les avantages et les limitations des différentes solutions. Basées sur les discussions et critiques effectuées, des améliorations à ce paradigme ont été proposées, elles consistent essentiellement en:

- L'amélioration d'un algorithme de clustering afin de prendre en compte les états énergétiques des nœuds lors de la formation et la maintenance de la topologie.
- Ensuite appliquer cet algorithme avec la diffusion dirigée dans le but de prolonger la durée de vie des nœuds et ainsi de tout le réseau.

Une évaluation de performances de la solution proposée est réalisée à l'aide du simulateur NS 2.

## **Mots clés :**

Diffusion Dirigée, Conservation d'énergie, Clustering Passif, Simulation.

# SOMMAIRE

<b>INTRODUCTION.....</b>	<b>4</b>
<b>1. LES RESEAUX DE CAPTEURS ET LA DIFFUSION DIRIGEE .....</b>	<b>6</b>
1.1 INTRODUCTION .....	6
1.2 LES RESEAUX MOBILES AD HOC .....	7
1.3 LES RESEAUX DE CAPTEURS.....	8
1.3.1 Architecture d'un nœud capteur.....	8
1.3.2 Le modèle d'un réseau de capteurs.....	9
1.3.3 Le matériel.....	9
1.3.4 Les systèmes d'exploitation .....	10
1.3.5 Application des réseaux de capteurs .....	10
1.3.6 Caractéristiques d'un réseau de capteurs.....	11
1.3.7 Comparaison entre les réseaux de capteurs et les réseaux Ad hoc.....	11
1.4 LA DISSEMINATION DE DONNEES DANS LES RESEAUX DE CAPTEURS .....	12
1.4.1 Classification des protocoles de routage dans les réseaux de capteurs.....	12
1.4.1.1 Les protocoles géographiques.....	13
1.4.1.2 Les protocoles hiérarchiques .....	13
1.4.1.3 Les protocoles orientés qualité de service .....	13
1.4.1.4 Les protocoles centrés données.....	13
1.5 LA DIFFUSION DIRIGEE.....	14
1.5.1 Terminologie .....	14
1.5.2 L'identification des données.....	15
1.5.3 La propagation des intérêts.....	15
1.5.4 Etablissement des gradients .....	16
1.5.5 La propagation des données exploratrices.....	16
1.5.6 Le renforcement.....	16
1.5.7 La dissémination des données .....	17
1.5.8 La programmation d'un réseau de capteurs (APIs de publication/souscription) .....	17
1.5.9 Les algorithmes de dissémination .....	18
1.5.9.1 La diffusion Pull à deux phases .....	18
1.5.9.2 La diffusion Push.....	18
1.5.9.3 La diffusion Pull à une phase.....	18
1.5.10 Discussion :.....	19
1.6 CONCLUSION : .....	19
<b>2. LA CONSOMMATION ET LA CONSERVATION D'ENERGIE DANS LES RESEAUX DE CAPTEURS.....</b>	<b>21</b>
2.1 INTRODUCTION .....	21
2.2 LA CONSOMMATION D'ENERGIE DANS UN NŒUD CAPTEUR.....	21
2.2.1 La consommation d'énergie dans l'unité centrale de traitement .....	22
2.2.2 La consommation d'énergie au niveau des capteurs.....	22
2.2.3 La consommation d'énergie dans le système de communication .....	22
2.2.4 Consommation d'énergie et modes de fonctionnement .....	22
2.2.5 Les caractéristiques de la batterie.....	24
2.2.6 Le convertisseur DC-DC.....	25
2.3 LA CONSERVATION D'ENERGIE DANS LES RESEAUX DE CAPTEURS .....	25
2.3.1 Optimisation d'énergie au niveau d'un nœud capteur .....	26
2.3.1.1 L'étape de calcul.....	26
2.3.1.2 Les logiciels .....	26
2.3.1.3 Gestion de puissance de la radio .....	27
2.3.1.4 La coordination au niveau du nœud.....	28
2.3.2 Conservation de l'énergie dans le réseau .....	29

2.3.2.1	Le contrôle de la puissance de transmission .....	29
2.3.2.2	Gestion de la couche MAC .....	30
2.3.2.3	Les protocoles de contrôle de topologie : entre le MAC et le routage.....	32
2.3.2.4	Le routage .....	33
2.4	CONCLUSION.....	34
<b>3. LA CONSERVATION D'ENERGIE DANS LA DIFFUSION DIRIGEE .....</b>		<b>35</b>
3.1	INTRODUCTION .....	35
3.2	LA CONSOMMATION D'ENERGIE DANS LA DIFFUSION DIRIGEE.....	35
3.3	LES PROTOCOLES DE CONSERVATION D'ENERGIE DANS LA DIFFUSION DIRIGEE .....	36
3.3.1	<i>Le Rumor routing</i> .....	36
3.3.2	<i>Routage à base de gradient</i> .....	37
3.3.3	<i>La diffusion dirigée avec routage géographique (GEAR)</i> .....	38
3.3.4	<i>Le routage efficace en consommation d'énergie (Energy Aware Routing)</i> .....	40
3.3.5	<i>La diffusion dirigée avec énergie solaire</i> .....	41
3.3.6	<i>La diffusion dirigée avec routage multiple</i> .....	42
3.3.6.1	Solution1.....	43
3.3.6.2	Solution 2.....	44
3.3.7	<i>La diffusion dirigée avec le clustering passif</i> .....	45
3.3.7.1	Discussion et critiques .....	46
3.4	CONCLUSION.....	47
<b>4. PRESENTATION D'UNE AMELIORATION D'UN ALGORITHME DE CLUSTERING POUR LA CONSERVATION D'ENERGIE DANS LA DIFFUSION DIRIGEE .....</b>		<b>48</b>
4.1	INTRODUCTION .....	48
4.2	POURQUOI LE CLUSTERING ? .....	48
4.3	LE CLUSTERING DANS LES RESEAUX DE CAPTEURS .....	49
4.3.1	<i>Le clustering passif</i> .....	49
4.3.1.1	Les éléments du clustering passif .....	50
4.3.1.2	Le principe de fonctionnement .....	51
4.4	LE CLUSTERING PASSIF EFFICACE EN CONSOMMATION D'ENERGIE: ENERGY AWARE PASSIVE CLUSTERING (EAPC) .....	52
4.4.1	<i>Les éléments du protocole :</i> .....	53
4.4.1.1	Les états des noeuds.....	53
4.4.1.2	L'entête des messages manipulés .....	53
4.4.2	<i>Les règles de fonctionnement</i> .....	53
4.4.3	<i>Les étapes de l'algorithme</i> .....	54
4.4.3.1	L'étape d'initialisation.....	54
4.4.3.2	L'étape de formation et de maintenance :.....	55
4.4.4	<i>La définition et la réévaluation des paramètres</i> .....	57
4.5	L'APPLICATION DU EAPC AVEC LA DIFFUSION DIRIGEE .....	57
4.5.1	<i>Les hypothèses de travail</i> .....	58
4.5.2	<i>Le principe</i> .....	58
4.5.3	<i>La structure des messages</i> .....	59
4.5.4	<i>L'implémentation du protocole</i> .....	60
4.6	CONCLUSION.....	60
<b>5. EVALUATION DE PERFORMANCES DE LA SOLUTION PROPOSEE .....</b>		<b>62</b>
5.1	INTRODUCTION .....	62
5.2	ENVIRONNEMENT DE SIMULATION .....	62
5.2.1	<i>Le simulateur NS (Network Simulator)</i> .....	63
5.2.2	<i>Architecture de NS</i> .....	63
5.2.3	<i>La diffusion dirigée dans NS</i> .....	64
5.2.4	<i>Le modèle d'énergie dans NS</i> .....	65
5.2.5	<i>Implémentation de la solution dans NS</i> .....	65

5.3	LES ELEMENTS DE LA SIMULATION.....	66
5.3.1	<i>La version de la diffusion dirigée.....</i>	66
5.3.2	<i>La charge du réseau :.....</i>	66
5.3.3	<i>Le modèle d'énergie.....</i>	67
5.4	LES SCENARIOS.....	67
5.4.1	<i>Génération des scénarios.....</i>	68
5.4.2	<i>Les paramètres de la simulation.....</i>	68
5.5	LES METRIQUES.....	69
5.5.1	<i>1<sup>er</sup> cas : une quantité d'énergie suffisante.....</i>	69
5.5.2	<i>2<sup>ème</sup> cas : une quantité d'énergie insuffisante.....</i>	71
5.6	RESULTATS ET INTERPRETATION DES RESULTATS.....	71
5.6.1	<i>Une quantité d'énergie suffisante.....</i>	71
5.6.1.1	Effet de la topologie du réseau.....	71
5.6.1.2	Effet du trafic.....	73
5.6.2	<i>Une quantité d'énergie insuffisante.....</i>	75
5.6.2.1	Effet de la topologie du réseau.....	75
5.6.2.2	Effet du trafic.....	76
5.6.3	<i>Une quantité d'énergie variable.....</i>	78
5.7	CONCLUSION.....	79
<b>6.</b>	<b>CONCLUSION.....</b>	<b>81</b>
<b>7.</b>	<b>BIBLIOGRAPHIE.....</b>	<b>83</b>

# Introduction

L'essor des technologies des systèmes micro-electro-mécaniques, offre aujourd'hui de nouvelles perspectives dans plusieurs domaines d'applications. L'évolution dans ce domaine a donné naissance à des micro-composants, appelés micro-capteurs ayant plusieurs fonctionnalités telles que le captage, le traitement et la transmission/réception de données. Cette évolution conjuguée à celle des technologies de communications sans fil a fait du déploiement des réseaux de capteurs une vision pragmatique.

La technologie des réseaux de capteurs est une technologie clé pour les années futures. Elle a été annoncée comme une des plus importantes technologies du 21<sup>ème</sup> siècle. Des composants d'un coût raisonnable, intelligents (capacités de traitement), avec plusieurs capteurs à bord, interconnectés à travers des liens sans fil sont déployés en grand nombre dans différents endroits. Cette interconnexion a un effet profond et des opportunités sans précédents dans beaucoup d'applications militaires et civiles telles que la détection d'intrusions, la surveillance de l'environnement, les applications de tactique, l'habitat intelligent, la télémédecine, la détection des conditions ambiantes (la température, le mouvement, la lumière), et la gestion des catastrophes.

A la différence des réseaux traditionnels et même Ad hoc, les réseaux de capteurs sont caractérisés par un déploiement dense des noeuds, des changements très fréquents de topologie, et des contraintes de calcul, de mémoire et de puissance très sévères. Ces caractéristiques uniques soulèvent beaucoup de nouveaux défis dans la réalisation des réseaux de capteurs. En particulier, les noeuds capteurs sont habituellement alimentés par des batteries de petites tailles et devraient fonctionner sans assistance pendant des périodes relativement longues. Dans la plupart des cas, il est difficile, voire impossible de changer ou recharger les batteries des noeuds capteurs.

La prolongation de la durée de vie d'un réseau de capteurs est un objectif partagé par tous les travaux destinés à ce type de réseaux du fait qu'elle est équivalente à celle du processus de captage que le réseau est censé mettre en œuvre. De là, l'efficacité en consommation d'énergie vient s'imposer comme considération dominante dans toutes les solutions proposées dans ce type de réseaux indépendamment du problème traité. Notamment dans les communications entre les noeuds qui constituent une étape très consommatrice en terme d'énergie.

Pour assurer la communication entre les noeuds dans un réseau de capteurs, plusieurs protocoles de routage ont été proposés. Une nouvelle classe de routage orienté données est apparue pour ce type de réseaux dans laquelle l'identification est appliquée sur les données communiquées plutôt que sur les noeuds du réseau. La diffusion dirigée est un exemple de protocole de routage orienté données. Il applique une identification des données sous forme de paires attribut/valeur et utilise des traitements intra-réseau (l'agrégation, la fusion, le cache, ...etc.) dans le routage de données.

Dans son fonctionnement, la diffusion dirigée utilise des données de niveau application et des informations locales transportées dans les messages échangés. En l'absence d'un système d'adressage global dans les réseaux de capteurs, les protocoles de routage, en général, et la diffusion dirigée, en particulier, font recours au mécanisme d'inondation. Ce dernier est très coûteux en terme de consommation d'énergie.

Dans cette thèse, nous nous sommes intéressés au problème de conservation d'énergie dans les réseaux de capteurs et plus particulièrement dans la diffusion dirigée. Ainsi, nous

proposons une amélioration de ce paradigme dans le but de le rendre plus efficace en consommation d'énergie. Notre contribution consiste essentiellement en :

- L'amélioration d'un algorithme de clustering afin de prendre en compte les états énergétiques des nœuds lors de la formation et la maintenance de la topologie.
- Ensuite appliquer cet algorithme avec la diffusion dirigée dans le but de prolonger la durée de vie des nœuds et ainsi de tout le réseau.

Cette thèse est composée de cinq chapitres :

Le chapitre 1 introduit les réseaux de capteurs et la diffusion dirigée. Du fait que les réseaux de capteurs ont émergé comme un cas distinctif des réseaux Ad hoc conventionnels, ce chapitre présente d'abord une brève définition des réseaux Ad hoc. Ensuite nous introduisons les réseaux de capteurs et leurs caractéristiques logicielles et matérielles, ainsi qu'une comparaison avec les réseaux Ad hoc. Le problème de dissémination de données et une classification des travaux proposés dans ce domaine sont donnés ultérieurement. La diffusion dirigée étant le cœur de notre travail, nous présentons et expliquons en détail ce paradigme dans la dernière partie de ce chapitre.

Le chapitre 2 présente les niveaux de consommation et de conservation d'énergie dans les réseaux de capteurs. Le chapitre commence par une analyse de la consommation d'énergie d'un nœud capteur dans ses différentes étapes de fonctionnement. Ensuite nous présentons les niveaux de conservation d'énergie d'un nœud capteur. En dernier lieu, nous présentons les solutions et les techniques proposées dans la conservation d'énergie au niveau du réseau dans les couches liaison et réseau.

L'importance du facteur de la consommation d'énergie dans les réseaux de capteurs a orienté beaucoup de travaux menés sur le paradigme de la diffusion dirigée. Le chapitre 3 est un état de l'art des solutions proposées pour conserver l'énergie dans la diffusion dirigée. Nous donnerons tout au long de ce chapitre des critiques et discussions des solutions présentées.

Le chapitre 4 présente notre contribution pour la conservation d'énergie dans la diffusion dirigée. Nous proposons, d'abord, une amélioration d'un protocole de clustering dans le but de le rendre plus efficace en consommation d'énergie. Ensuite nous proposons sa combinaison avec la diffusion dirigée.

Le chapitre 5 est consacré à l'évaluation de performances de la solution proposée à l'aide de simulation. Cette dernière est faite avec le simulateur de réseau NS 2. Nous justifions tout au long de ce chapitre nos choix de conception et nos paramètres de simulation. Les graphes qui schématisent nos résultats et leurs interprétations seront exposés à la fin du chapitre.

# Chapitre 1

## Les réseaux de capteurs et la diffusion dirigée

### 1.1 Introduction

L'évolution rapide de la technologie dans le domaine de la communication sans fil et les technologies « MEMS » (Micro-electro-mechanical systems) ont permis le développement des micro-composants qui intègrent des dispositifs de captage, de communication sans fil et de calcul. Ces micro-composants sont appelés micro-capteurs ou capteurs. Ils ont une taille très réduite et un coût de déploiement raisonnable.

Les micro-capteurs peuvent mesurer les conditions ambiantes de l'environnement telles que la température ou l'humidité et les transforment en des signaux électriques. Le traitement de tels signaux est d'une grande importance dans un large spectre d'applications. De ce fait, un grand nombre de micro-capteurs sont interconnectés entre eux formant un réseau appelé, réseau de capteurs. Les applications des réseaux de capteurs peuvent inclure : les applications militaires et de tactique, de santé, de surveillance de l'environnement, d'habitat intelligent, et plus simplement les applications commerciales.

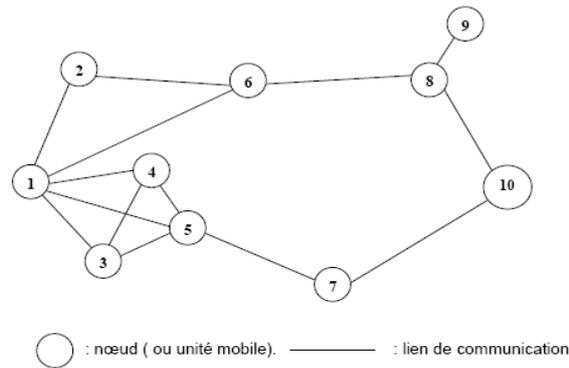
Les réseaux de capteurs peuvent être considérés comme un type particulier des réseaux Ad hoc. Ils héritent alors toutes les caractéristiques de ces derniers et en engendrent d'autres comme : le grand nombre des nœuds, l'absence d'adressage global, ...etc. Un nœud capteur peut jouer le rôle de capteur d'information ou de collecteur. Dans ce dernier cas, il est appelé puits. Les données captées sont envoyées par les nœuds capteurs en direction des nœuds puits. Pour réaliser cela, des mécanismes de dissémination de données ou routage sont utilisés pour assurer la communication entre les nœuds du réseau.

Plusieurs classes de protocoles de routage ont été considérées dans les réseaux de capteurs. Une nouvelle classe de protocoles orientés données est apparue dans ce type de réseaux. Dans cette catégorie, l'identification est faite sur les données du réseau plutôt que sur ses nœuds. La diffusion dirigée est un paradigme de routage appartenant à cette catégorie. Ce paradigme constitue une solution très prometteuse dans les réseaux de capteurs.

Le but de ce chapitre est de présenter les principaux concepts liés aux réseaux de capteurs et à la diffusion dirigée. Du fait que les réseaux de capteurs ont émergé comme un cas distinctif des réseaux Ad hoc conventionnels, ce chapitre va présenter d'abord une brève définition des réseaux Ad hoc. Ensuite nous introduisons les réseaux de capteurs et leurs caractéristiques logicielles et matérielles, ainsi qu'une comparaison entre eux et les réseaux Ad hoc. Ceci sera suivi d'une introduction au problème et aux différentes classes de routage dans ce type de réseaux. En dernier lieu, nous présentons et expliquons en détail le paradigme de la diffusion dirigée, un exemple de protocole de routage orienté données dans les réseaux de capteurs.

## 1.2 Les réseaux mobiles ad hoc

Un réseau mobile Ad Hoc, appelé généralement MANET (Mobile Ad Hoc NETwork), comprend des plates-formes mobiles (par exemple, un routeur interconnectant différents hôtes et équipements sans fil) appelés noeuds qui sont libres de se déplacer sans contrainte. Un réseau Ad hoc est donc un svstème autonome de noeuds mobiles.



**Figure 1.1 :** La modélisation d'un réseau ad hoc.

Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes au travers des passerelles. Dans ce dernier cas, un réseau Ad hoc est un réseau d'extrémité [1]. Un réseau Ad hoc peut être modélisé par un graphe  $G_t = (V_t, E_t)$  où  $V_t$  représente l'ensemble des noeuds ( i.e. les unités ou les hôtes mobiles) du réseau et  $E_t$  modélise l'ensemble des connexions qui existent entre ces noeuds. Si  $e = (u,v) \in E_t$ , cela veut dire que les noeuds  $u$  et  $v$  sont en mesure de communiquer directement à l'instant  $t$  [2] (voir la figure 1.1). Les caractéristiques d'un réseau Ad hoc peuvent être résumées en [3]:

- **La mobilité :** La mobilité des noeuds constitue à l'évidence une caractéristique très spécifique des réseaux Ad hoc. Cette mobilité est intrinsèque au fonctionnement du réseau. Dans un réseau Ad hoc, la topologie peut donc changer rapidement, de façon aléatoire et non prévisible.
- **L'équivalence des noeuds du réseau :** Dans un réseau classique, il existe une distinction nette entre les noeuds terminaux (stations, hôtes) qui supportent les applications et les noeuds internes (routeurs par exemple) du réseau. Cette différence n'existe pas dans les réseaux Ad hoc car tous les noeuds peuvent être amenés à assurer les mêmes fonctions.
- **Les liaisons sans fil :** Les technologies de communication sans fil sont indispensables à la mise en place d'un réseau Ad hoc. Malgré des progrès très importants, leurs performances restent et resteront en deçà de celles des technologies des réseaux filaires. Une de leurs caractéristiques primordiales est l'utilisation d'un médium de communication partagé. Ce partage fait que la bande passante réservée à un hôte soit modeste
- **L'autonomie des noeuds :** La consommation d'énergie constitue un problème important pour des équipements fonctionnant grâce à une alimentation électrique autonome.
- **La vulnérabilité :** Les réseaux sans fil sont par nature plus sensibles aux problèmes de sécurité. Pour les réseaux Ad hoc, le principal problème ne se situe pas tant au

niveau du support physique mais principalement dans le fait que tous les noeuds sont équivalents et potentiellement nécessaires au fonctionnement du réseau. Les possibilités de s'insérer dans le réseau sont plus grandes.

### 1.3 Les réseaux de capteurs

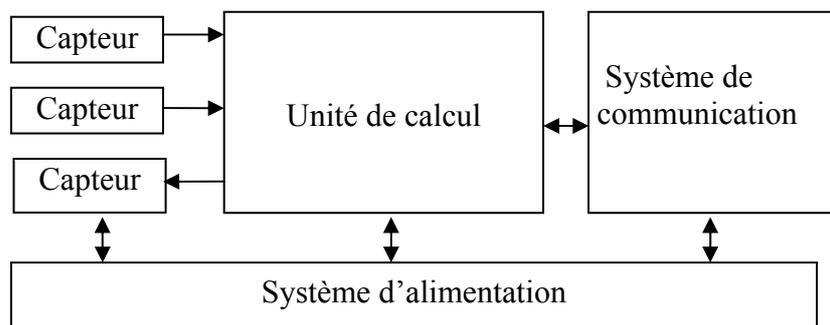
Les réseaux de capteurs sont considérés comme un type particulier des réseaux sans fil Ad hoc. Ce sont des réseaux de nœuds dédiés à une application. Un réseau de capteurs consistait en un petit nombre de nœuds capteurs reliés par des liens filaires à une station de traitement centrale. Cependant, de nos jours, l'intérêt est porté sur les réseaux de capteurs sans fil regroupant un grand nombre de nœuds.

Un réseau de capteurs sans fil se compose d'un grand nombre de nœuds capteurs qui sont distribués d'une manière dense à travers une zone d'observation [4]. Dans cette dernière, les capteurs se coordonnent entre eux pour réaliser la surveillance d'un phénomène précis. Chaque capteur est capable de communiquer et a un certain niveau d'intelligence lui permettant de traiter des signaux et de communiquer des données. Un capteur peut envoyer une donnée ou une combinaison de données (somme, moyenne), sauvegarder une donnée à son niveau pour une utilisation ultérieure ou bien réagir par l'envoi d'un message suite au déclenchement d'un évènement particulier.

#### 1.3.1 Architecture d'un nœud capteur

Les prototypes de nœuds capteurs qui ont été développés et sont actuellement employés par différents groupes de recherche sont en principe très semblables. Les exemples pour de tels nœuds incluent : les nœuds WINS et MEDUSA II de UCLA [5], COTS Motes (projet smart-dust) de Berkeley [6], WEbS [7], PicoRadio [8], uAMPS[12], GNOMES[11], les nœuds du projet européen EYES [9], et des nœuds développés localement dans des universités [10].

L'architecture générale d'un nœud capteur est schématisée dans la figure suivante (figure 1.2). Le nœud est constitué de quatre sous systèmes : une unité de calcul, un système de communication pour la transmission et la réception des données, un système de capture, et un système d'alimentation en énergie [13].



**Figure 1.2 :** Architecture générale d'un nœud capteur

#### *a. L'unité de calcul (microcontrôleur ou microprocesseur MCU)*

C'est la partie intelligente du nœud capteur, elle est responsable du contrôle des capteurs, de l'exécution des protocoles de communication et celle des algorithmes de

traitement du signal appliqués sur les données captées. Les microprocesseurs utilisés généralement sont StrongARM de Intel<sup>1</sup> et AVR de Atmel<sup>2</sup>.

### **b. Les capteurs**

Un nœud peut contenir plusieurs capteurs capables de transformer les phénomènes physiques captés en des signaux électriques analogiques ou numériques selon le type de résultat qu'ils produisent. Il existe une diversité de capteurs qui mesurent les paramètres de l'environnement tels que la température, l'intensité lumineuse, le son, les images, etc.

### **c. Le système de communication**

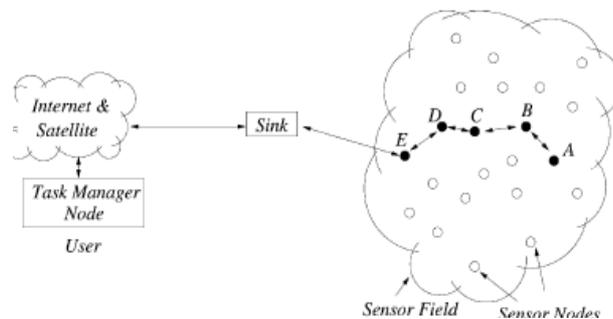
Ce système est constitué d'un support de communication sans fil (infrarouge, optique, fréquences radio, ...) qui sert à relier le nœud à ses voisins et au reste du réseau. La majorité des prototypes des nœuds capteurs construits utilisent les radiofréquences comme moyen de communication. Par exemple, les nœuds uAMPS [12] utilisent une unité de transmission 2.4 GHZ compatible bluetooth.

### **d. Le système d'alimentation**

Le système d'alimentation loge la batterie et le convertisseur<sup>3</sup> dc-dc [14]. Son rôle est d'alimenter le reste des composants du nœud.

## **1.3.2 Le modèle d'un réseau de capteurs**

Dans le modèle classique, un nœud spécial, appelé puits (sink), est supposé avoir plus de capacités relativement aux autres nœuds [15]. Il peut ou pas communiquer à l'aide d'une seule diffusion avec tous les composants du réseau. Il constitue l'origine des demandes des opérations et les données (réponses) sont envoyées en sa direction. Les approches en cours, par contre, sont orientées vers des réseaux autonomes où un ou plusieurs puits se connectent pour distribuer des tâches ou collecter des informations. Les puits peuvent également jouer le rôle de passerelles vers d'autres types de réseaux.



**Figure 1.2 :** Le modèle d'un réseau de capteurs

## **1.3.3 Le matériel**

L'architecture MOTES présentée dans [6] constitue un état de l'art sur le matériel utilisé dans les réseaux de capteurs. Les Motes sont des nœuds capteurs développés et utilisés réellement. Ils constituent un champ de test pratique dans ce domaine. Un Mote est

<sup>1</sup> [http://www.intel.com/design/pca/applicationsprocessors/1110\\_brf.htm](http://www.intel.com/design/pca/applicationsprocessors/1110_brf.htm)

<sup>2</sup> <http://www.atmel.com/products/AVR/>

<sup>3</sup> D'un point de vue général, tout dispositif qui modifie les caractéristiques d'une source d'énergie s'appelle un convertisseur

composé de deux éléments : le MICA board et le sensor board. Le premier forme la partie calcul, il est doté d'un processeur de vitesse 4/8 Mhz. Ce processeur a 32 registres généraux, une mémoire de programmes de 128 Kb (512 pages avec 256 octets dans chacune) et une mémoire pour les données de 4 Kb. Il contient aussi une mémoire de capacité de 4 Mb utilisée comme mémoire externe. Le jeu d'instructions est composé de 121 instructions. Concernant l'énergie, le processeur peut être dans les états suivants : actif, veille (avec la CPU en arrêt et le timer et la mémoire en marche), à un état d'énergie faible ou en état de sauvegarde d'énergie. Cet élément est aussi équipé d'un transmetteur radio d'une très petite puissance avec une portée de transmission dépendant des conditions de l'environnement dans lequel il est employé. Le deuxième, le sensor board (similaire au MTS310CA (figure 1.3 (b)) offre plusieurs facilités de captage comme le taux de lumière, le son, et l'accélération. Des efforts sont entrain de se faire dans le processus de standardisation de IEEE 802.15.4 dans le but de créer une forme d'unification des travaux proposés dans ce domaine.



(a) : Le MICA board



(b) : Le sensor board

**Figure 1.3** : Les composants de l'architecture MOTES

#### **1.3.4 Les systèmes d'exploitation**

Les plateformes logicielles varient selon le type du matériel. Ainsi pour chaque plateforme matérielle proposée est associé un système d'exploitation spécialisé. Habituellement, le rôle d'un système d'exploitation consiste à faire une abstraction du matériel. La conception pourra varier d'une simple abstraction au-dessus d'un environnement d'exécution comme le TinyOS (pour le projet MOTES) à un véritable gestionnaire de tâches (scheduler) comme dans le cas de EYESOS (pour le projet EYES).

Dans l'architecture MOTES, le système d'exploitation utilisé est le TinyOS [16]. Ce dernier a été conçu pour assurer un minimum de fonctionnalités en raison des contraintes serrées de l'architecture (taille, alimentation) et parce que beaucoup d'aspects comme la gestion des connexions et celle des capteurs sont fortement liés aux types d'applications. Le système est composé d'une routine d'initialisation (172 octets), une routine en C (82 octets) et un gestionnaire d'événements (178 octets) qui constitue le noyau du système. Une application complète est un graphe de composants chacun représentant un module logiciel composé : d'un ensemble de tâches et de gestionnaires d'événements, de mémoire et de commandes. Les applications sont écrites en langage C spécialisé, appelé nesC.

#### **1.3.5 Application des réseaux de capteurs**

Les nœuds développés pour les réseaux de capteurs ne sont pas coûteux. Ils sont petits, composés de plusieurs capteurs et dotés d'interface de communication sans fil. Ils sont

déployables sur le sol, dans l'air, sous la mer, dans les corps, dans les véhicules et dans les immeubles.

Les réseaux de capteurs peuvent être composés, suivant leur utilisation, de différents types de nœuds capteurs tels que les capteurs sismiques, thermiques, visuels, infrarouges, acoustiques et radar, ils sont capables de surveiller une grande variété de phénomènes ambiants (température, humidité, mouvement des véhicules, accélération,...). Ils peuvent donc être utilisés pour la surveillance continue d'un phénomène, la détection et l'identification d'un évènement, la surveillance d'un terrain et le contrôle local d'objets déclencheurs d'évènements (actionneurs). Ceci peut être fait dans plusieurs domaines tels que : le domaine militaire, la santé, la gestion et la surveillance de l'environnement et les applications commerciales [4].

### 1.3.6 Caractéristiques d'un réseau de capteurs

Les réseaux de capteurs sont comme les réseaux Ad hoc caractérisés par l'absence de l'infrastructure fixe, une topologie dynamique, des capacités matérielles limitées, des liens de communication non fiables, des risques de sécurité...etc. A toutes ces contraintes héritées des réseaux Ad hoc viennent s'ajouter d'autres contraintes spécifiques aux réseaux de capteurs [17]:

- **Les contraintes matériels** : Les capteurs sont trop petits et donc très limités en capacité de calcul et de mémorisation (mémoire, CPU). Ils sont alimentés par des sources d'énergie non renouvelables.
- **Le nombre des nœuds capteurs** : Pour capter ou surveiller un phénomène, les nœuds capteurs sont employés en grand nombre. Ceci est aussi encouragé par leur prix qui est de plus en plus bas.
- **Le déploiement aléatoire** : Afin de surveiller un phénomène, les nœuds capteurs sont distribués d'une façon très aléatoire dans la zone d'observation.
- **La densité** : Les nœuds capteurs sont nombreux et déployés d'une manière aléatoire dans l'environnement. Les zones d'observations présentent donc des densités différentes par rapport au nombre des nœuds.
- **L'identification des nœuds** : Dans un réseau de capteurs, les données collectées sont plus importantes que leurs sources et donc les nœuds constituant le réseau ne sont pas tous identifiés.
- **La durée de vie** : La durée de vie constitue un facteur important dans l'utilisation des réseaux de capteurs. La maximisation de ce facteur est équivalente à celle du processus de surveillance que le réseau est censé mettre en œuvre. Cela fait de sa maximisation un objectif de priorité.
- **Les objectifs de déploiement** : Les réseaux de capteurs sont dédiés applications et l'objectif de les mettre en œuvre est ciblé dans la plupart des cas à l'avance. Les nœuds collaborent donc pour remplir cet objectif.
- **Le changement rapide de topologie** : La topologie d'un réseau de capteurs peut changer fréquemment notamment dans le cas où le déplacement des nœuds est influencé par les conditions de l'environnement (par exemple le vent).

### 1.3.7 Comparaison entre les réseaux de capteurs et les réseaux Ad hoc

Le tableau suivant illustre les différences majeures entre les réseaux de capteurs et les réseaux Ad hoc [17]:

<b>Réseaux de capteurs</b>	<b>Réseaux Ad hoc</b>
Objectif ciblé	Générique/communication
Nœuds collaborent pour remplir un objectif	Chaque nœud a son propre objectif
Flot de données : many-to-one ou one-to-many	Flot de données : any-to-any
Très grand nombre de nœuds n'ayant pas tous un identificateur	Notion d'ID
Energie est un facteur déterminant	Débit est majeur

## **1.4 La dissémination de données dans les réseaux de capteurs**

Généralement, le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste pour un réseau à déterminer un acheminement optimal des paquets (de messages, de produits ...etc.) à travers le réseau au sens d'un certain critère de performance. Le problème consiste à trouver l'investissement de moindre coût en capacités nominales et de réserves qui assure le routage du trafic nominal et garantit sa survivabilité en cas de n'importe quelle panne de canal ou de nœud [2].

Dans les réseaux traditionnels, les mécanismes de routage sont utilisés pour assurer la communication entre les composants du réseau. Ces mécanismes sont souvent basés sur l'utilisation des schémas d'adressage global et le calcul des métriques. Dans les réseaux de capteurs le mot routage est souvent remplacé par *dissémination de données* pour faire ressortir l'absence de l'adressage global, la dépendance aux rôles locaux des nœuds et l'utilisation du traitement spécifique à l'application [18].

### **1.4.1 Classification des protocoles de routage dans les réseaux de capteurs**

Les travaux traditionnels proposés dans le domaine du routage semblent être difficiles à adapter dans les réseaux de capteurs en raison des limitations rencontrées. La construction des routes pour un réseau de capteurs doit être faite avec un minimum de contrôle et de consommation de ressources prenant en compte l'absence d'un adressage global des nœuds. Les solutions apportées à cette dernière contrainte varient d'un protocole à un autre et constitue un critère de classification des protocoles proposés. Ainsi, ces derniers peuvent être classées en trois grandes catégories : les protocoles orientés ou centrés données, les protocoles hiérarchiques et ceux géographiques (à base de localisation).

La plupart des protocoles proposés peuvent être classés sous une des catégories citées. Cependant, quelques autres protocoles suivent d'autres principes de conception comme les protocoles basés sur la maintenance de la qualité de service. D'autres classifications peuvent contenir : les protocoles à chemins multiples et les protocoles à base de requêtes. Quoique, les protocoles de routage puissent être hybrides appartenant ainsi à plusieurs catégories à la fois.

#### **1.4.1.1 Les protocoles géographiques**

Les protocoles de routage appartenant à cette catégorie exigent l'information de localisation relative aux nœuds capteurs. Cette information est utilisée pour calculer la distance entre une paire de nœuds de sorte que la consommation d'énergie puisse être déterminée. En l'absence d'un système d'adressage pour les nœuds et leur déploiement aléatoire, l'information de localisation est aussi employée pour router les données à une région particulière [25, 26]. Ceci peut minimiser le nombre de transmissions d'une manière significative. Plusieurs protocoles, appartenant à cette catégorie, sont principalement conçus pour les réseaux mobiles Ad hoc [27]. Des apports concernant la consommation d'énergie ont été utilisés afin de les adapter aux réseaux de capteurs.

#### **1.4.1.2 Les protocoles hiérarchiques**

Les protocoles hiérarchiques de leur part tentent de réaliser le passage à l'échelle (scalabilité) en présence d'un grand nombre de nœuds dans le réseau. Le but principal du routage hiérarchique est de minimiser la consommation d'énergie au niveau des nœuds capteurs. Ceci est basé sur des techniques de clustering et traitements de données. Ainsi, à chaque cluster (groupe) formé, on associe un représentant responsable du routage. Ce dernier utilise des techniques comme l'agrégation et la fusion pour collecter les données dans son groupe et diminuer ainsi le nombre de transmissions. Aussi, l'identification ou l'adressage peut être appliqué aux têtes de clusters plutôt qu'à tous les nœuds du réseau pour pallier au problème d'adressage. LEACH [28] est une des premières approches du routage hiérarchique dans les réseaux de capteurs. L'idée proposée dans LEACH est à la base de beaucoup de protocoles hiérarchiques proposés ultérieurement [23, 29], bien que quelques protocoles aient été indépendamment développés [30].

#### **1.4.1.3 Les protocoles orientés qualité de service**

Les protocoles orientés qualité de service emploient des métriques de cette dernière (délai, énergie, bande passante, ...etc.) au moment de l'établissement des chemins entre les nœuds. Dans le cas d'un réseau de capteurs, celui-ci doit effectuer un équilibre entre la consommation d'énergie et la qualité de données au moment de leur délivrance [31, 32].

#### **1.4.1.4 Les protocoles centrés données**

Dans un protocole de routage plat tous les nœuds du réseau effectuent les mêmes tâches vis à vis de la fonction du routage. En raison du grand nombre de nœuds dans un réseau de capteurs, cette approche est peu performante en l'absence d'un système d'adressage global. Un tel manque d'identification ajouté au déploiement aléatoire des nœuds rendent la sélection d'un sous-ensemble de nœuds spécifiques une tâche irréalisable. En conséquence, une nouvelle classe de protocoles de routage orienté données est apparue. Dans les approches orientées données l'identification est appliquée sur les données et non pas sur les nœuds du réseau. Ainsi, un nœud de capteurs peut ne pas avoir besoin d'une identité c'est-à-dire, les applications des réseaux de capteurs sont peu susceptibles de poser la question: Quelle est la température au nœud n° 12 ? Au lieu de cela, les applications se concentrent sur les données produites par les capteurs. Les données sont identifiées par des attributs et les applications demandent celles conformes à certaines valeurs de ces attributs. De ce fait, la requête peut être de la forme : quelles sont les zones dans lesquelles la température a récemment excédé les 30 degrés? Cette approche découple les données des capteurs qui les produisent. Ceci mène à une conception plus robuste des applications: même si le capteur n° 12 tombe en panne, les données qu'il produit peuvent être (probablement) cachées dans d'autres capteurs voisins pour une récupération ultérieure.

Le protocole SPIN [19] est le premier protocole orienté données proposé pour les réseaux de capteurs. Il considère la négociation de données entre les noeuds afin d'éliminer des données superflues et économiser ainsi plus d'énergie. Plus tard, la diffusion dirigée [20] a été développée et est devenue un paradigme dans le routage ou la dissémination de données dans ce type de réseau. Beaucoup d'autres protocoles basés sur ce paradigme ont été proposés tels que : [21, 22]. D'autres approches suivent un concept similaire tels que [23, 24].

*Notre travail étant la conservation d'énergie dans la diffusion dirigée, nous consacrerons la prochaine section à une description détaillée de ce paradigme.*

## **1.5 La diffusion dirigée**

Les algorithmes distribués ont beaucoup de propriétés souhaitables dans le contexte des réseaux de capteurs [33]. Ces algorithmes peuvent supporter les changements fréquents de topologie, l'accroissement du nombre des noeuds et même les pannes des différents éléments du réseau. Cependant, il est difficile de concevoir ce type d'algorithmes en raison du rapport potentiellement compliqué entre le comportement local et celui global des nœuds du réseau. Une solution à ce problème serait de définir un ensemble d'abstractions qui décrivent le modèle de communication pour ce type d'algorithmes. Dans cette partie, nous explorons un tel ensemble nommé, la diffusion dirigée [20].

La diffusion dirigée est un paradigme de dissémination de données dans les réseaux de capteurs. Il adopte des APIs (Application Programming Interface) de publication/souscription qui isolent les producteurs et les consommateurs de données des détails algorithmiques de la dissémination. L'abstraction principale consiste en l'identification des données par un ensemble d'attributs, les producteurs de données (ou les sources) produisent les données et les publient, les consommateurs (ou les puits) souscrivent aux données, et c'est à l'algorithme de diffusion d'assurer l'échange efficace de ces données entre les éditeurs et les abonnés.

La diffusion dirigée est un paradigme orienté-données où toutes les communications sont faites pour des données identifiées. Les nœuds dans le réseau sont sensibles aux besoins des applications en manipulant des codes et des données qui leur sont spécifiques. Ceci permet à la diffusion dirigée de mettre en cache, d'agréger et de traiter les données dans le réseau. L'utilisation de ces techniques aide à éliminer les redondances et à minimiser le nombre de transmissions ; et donc à conserver l'énergie et prolonger la durée de vie du réseau.

### **1.5.1 Terminologie**

Un message d'intérêt ou intérêt est une requête ou une interrogation qui indique ce que veut un utilisateur. Chaque intérêt contient une description d'une tâche de captage soutenue par le réseau de capteurs pour acquérir des données. Le nœud diffusant l'intérêt explorateur initial est appelé puits. Ce dernier forme la destination finale des données requises. Les données dans les réseaux de capteurs sont les informations rassemblées ou traitées d'un phénomène physique. Dans la diffusion dirigée, l'adjectif "explorateur" fait référence à une diffusion non optimisée qui va disparaître si elle n'est pas renforcée. Un gradient est un état de direction créé dans chaque nœud recevant un intérêt [34]. Les données sont diffusées vers les puits suivant des chemins multiples en utilisant les gradients. Un ou plusieurs chemins sont renforcés et seront utilisés pour la dissémination de données.

### 1.5.2 L'identification des données

Le schéma d'identification ou représentation de données utilisé est basé sur l'utilisation des attributs. Cette identification est appliquée pour toutes les données du réseau : les intérêts et les réponses. Cette approche d'identification est souple et présente beaucoup d'avantages : D'abord, elle est conforme à l'interface applicative du mécanisme de publication/souscription et à la communication entre les nœuds. Aussi, cette identification est orientée données, permettant aux applications de se concentrer sur les données demandées plutôt que sur les différents nœuds capteurs. En plus, l'utilisation des attributs fournit une certaine structure aux messages. En identifiant des champs séparément, les algorithmes de diffusion peuvent employer diverses informations et filtres pour influencer le routage. En outre, le traitement des messages en tant qu'ensembles d'attributs simplifie les extensions des applications et des protocoles d'une part et la comparaison des données de l'autre. Si les attributs dans la souscription d'un puits correspondant à ceux se trouvant dans la publication d'une source, les données sont envoyées des sources vers le puits. Supposons une application de détection des animaux dans une zone géographique. Une description de l'intérêt sera sous la forme [18]:

```
Type = animal à quatre pattes // détecter la localisation d'un animal
Intervalle = 20 ms // envoyer les évènements chaque 20 ms
Durée = 10 secondes // pour les 10 prochaines secondes
Rect = [-100, 100, 200,400] // à partir des capteurs se trouvant dans un rectangle
```

Un nœud capteur qui détecte un animal, envoie une réponse représentée sous la même forme suivante:

```
Type = animal à quatre pattes // type de l'animal détecté
Instance = éléphant // instance du type
Localisation = [125,220] // localisation du noeud
Intensité = 0.6 // mesure de l'amplitude du signal
Fidélité = 0.91 //degrés de correspondance
Estampille = 01 :20 :40 // temps de génération de l'évènement
```

### 1.5.3 La propagation des intérêts

C'est la première étape du paradigme dans laquelle le nœud puits commence par diffuser le message d'intérêt à tous ses voisins. Contrairement, aux intérêts suivants, le premier message d'intérêt a une valeur d'intervalle plus grande. Il est donc considéré comme message d'exploration qui sert à identifier les capteurs concernés par la tâche demandée. L'intérêt est périodiquement rafraîchi par le puits. Pour ce faire, le puits renvoie le même intérêt avec un attribut durée incrémenté.

Chaque nœud maintient un cache ayant plusieurs entrées, chacune fait référence à un intérêt particulier. Deux intérêts sont différents s'ils ont deux types (deux tâches différentes) différents ou deux valeurs de rect différentes (zones ciblées différentes). Cette différence est exploitée pour effectuer des traitements sur les intérêts. Par exemple : deux intérêts I1 et I2 avec des informations identiques et des valeurs rect incluse l'une dans l'autre peuvent être représentés par un seul intérêt. Une entrée dans le cache contient

plusieurs champs, notamment un champ estampille qui indique le temps de la dernière réception de l'intérêt et plusieurs champs gradients, un par voisin. Chaque gradient spécifie le taux d'envoi de données exigé par le voisin (calculé à partir de l'attribut intervalle). L'entrée contient aussi un champ durée calculé à partir du champ *durée* indiquant approximativement la durée de vie de l'intérêt.

Quand le nœud reçoit l'intérêt, il vérifie son existence dans son cache. Si aucune entrée n'est trouvée, le nœud crée une entrée et instancie les valeurs des attributs à partir du message d'intérêt. Si l'entrée existe mais sans un champ gradient, ce dernier est ajouté par le nœud. Quand le gradient expire, il va être supprimé du cache. Les gradients relatifs à une entrée n'expire généralement pas en même temps. Quand tous les gradients d'une entrée expirent celle-ci va être supprimée. A la réception d'un intérêt, le nœud peut décider de le renvoyer à un sous-ensemble de ses voisins comme il peut décider de ne pas le renvoyer si par exemple un intérêt identique a déjà été envoyé. Généralement beaucoup de choix peuvent être faits au moment de la propagation des intérêts. Ces derniers peuvent être simplement rediffusés par chaque nœud ce qui est équivalent à une inondation ou bien rediffusés en utilisant des informations géographiques ou des informations relatives aux exécutions précédentes.

#### **1.5.4 Etablissement des gradients**

Les gradients sont établis au moment de la propagation des intérêts. Le gradient spécifie essentiellement un taux d'envoi de données et une direction dans laquelle l'événement sera envoyé. La diffusion dirigée accorde aux concepteurs d'applications la liberté d'attacher différentes sémantiques aux valeurs de gradients. En conclusion, la propagation des intérêts installe des états dans le réseau pour faciliter la propagation des données vers le puits. Les règles de propagation sont locales et différentes d'une tâche à une autre.

#### **1.5.5 La propagation des données exploratrices**

Un nœud capteur qui est dans une zone dont les coordonnées se concordent avec les données se trouvant dans l'attribut *rect* traite les intérêts reçus (devient une source). S'il détecte un événement, il cherche dans son cache une entrée correspondante. Dans ce cas, l'entrée correspondante est celle ayant le même type et une valeur *rect* qui englobe (ou égale à) la localisation du nœud. Si une telle entrée existe, le nœud calcule le plus grand taux d'envoi d'événement des différents gradients. Ce taux est utilisé par la source jusqu'à ce que le puits renforce un ou plusieurs liens. La source envoie alors à chacun de ses voisins, vers lesquels elle a des gradients, une description de l'événement (selon le taux). Un nœud qui reçoit un message de données tente de trouver une entrée pour un intérêt correspondant dans son cache. Si l'intérêt n'existe pas le message est rejeté sinon le nœud vérifie le cache de données de l'entrée existante et applique le même principe.

#### **1.5.6 Le renforcement**

Le renforcement, dans la diffusion dirigée peut être positif ou négatif selon l'objectif ciblé. Cette technique est utilisée pour l'établissement des liens, la réparation des chemins et l'enlèvement des boucles. Quand le puits commence à recevoir les messages de données (explorateurs), il renforce un chemin (voisin) sur lequel il reçoit les données réelles. Le renforcement consiste à l'envoi du message d'intérêt original avec un intervalle plus petit (taux d'envoi d'évènements plus élevé). Quand un nœud voisin reçoit un tel intérêt, il s'aperçoit qu'il a déjà un gradient pour ce voisin et que le taux indiqué est plus élevé. Si ce taux est plus élevé que tous les gradients existants, le nœud doit renforcer à son tour au

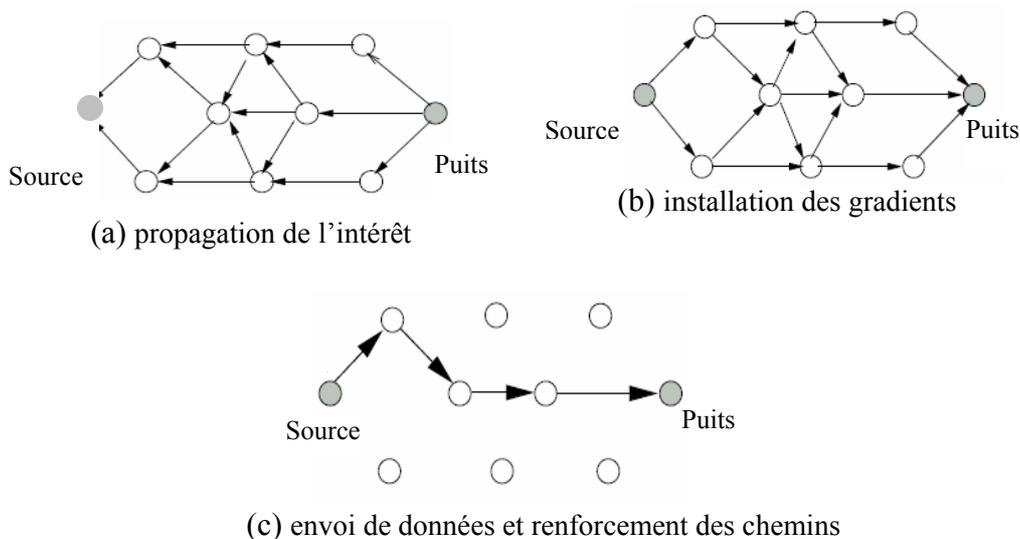
moins un voisin. Le renforcement est aussi utilisé dans l'établissement des liens entre plusieurs sources et puits. Beaucoup de règles peuvent être utilisées dans le renforcement comme : choisir le voisin à partir duquel le nœud a reçu le plus grand nombre de messages de données, ou bien le voisin qui envoie uniformément les données avant les autres nœuds.

La procédure de renforcement peut résulter en plusieurs chemins renforcés. Pour renforcer un seul chemin le nœud fait un renforcement négatif vers le reste des voisins. Le message de renforcement négatif est un intérêt avec un taux d'envoi d'événement inférieur à celui spécifié dans l'intérêt original. Les règles locales qu'utilise un nœud pour renforcer négativement un chemin sont définies par le concepteur de l'application. Un exemple serait de renforcer négativement un voisin qui n'a pas envoyé de nouveaux événements pendant une certaine durée de temps. Le renforcement négatif est aussi utilisé pour l'enlèvement des boucles. Cependant, cette procédure doit prendre en considération la présence des gradients partagés reliant plusieurs sources et puits.

### 1.5.7 La dissémination des données

Dans cette étape les nœuds utilisent le chemin renforcé (les chemins) pour envoyer les données.

Le fonctionnement de la diffusion dirigée est schématisé dans la figure (1.4) suivante :



**Figure 1.4:** Fonctionnement de la diffusion dirigée

### 1.5.8 La programmation d'un réseau de capteurs (APIs de publication/souscription)

La diffusion dirigée adopte des APIs (Application Programming Interface) basées sur le mécanisme de publication/souscription [35]. Pour recevoir des données, les utilisateurs ou les programmes souscrivent à un ensemble particulier d'attributs, devenant ainsi des puits. Les capteurs publient les données qu'ils ont détectées, et deviennent des sources de données. Les fonctions de l'interface de publication/souscription sont utilisées pour : l'initialisation d'une fonction de routage, la création ou la suppression d'une souscription ou d'une publication, l'envoi et la gestion des filtres, ...etc.

## **1.5.9 Les algorithmes de dissémination**

L'implémentation de la diffusion dans ce paradigme est faite en utilisant plusieurs algorithmes. Ces derniers dépendent du type d'application [36]:

### ***1.5.9.1 La diffusion Pull à deux phases***

Les étapes expliquées dans le fonctionnement de la diffusion dirigée constituent le premier algorithme de dissémination dans ce paradigme. La première phase de l'algorithme comporte l'inondation initiale des intérêts ainsi que celle des messages explorateurs de données. Le renforcement des chemins, et la transmission des données le long des chemins renforcés, constituent la deuxième phase de cet algorithme.

### ***1.5.9.2 La diffusion Push***

La diffusion à deux phases fonctionne bien pour les applications où un nombre restreint de puits rassemblent des données, par exemple, un utilisateur demandant des détections de certains objets dépistés. Une autre classe d'applications pourrait avoir besoin de capteurs fonctionnant souvent avec un bas coefficient d'utilisation, mais quand un capteur détecte quelque chose il déclenche les capteurs voisins pour devenir plus actifs et vigilants. La diffusion Push a été motivée par de telles applications. Une caractéristique de cette classe d'applications est qu'il y a beaucoup de capteurs intéressés aux données (déclencheurs), et beaucoup qui peuvent éditer de telles données, mais la fréquence des déclenchements est assez petite. La diffusion à deux phases est inapplicable dans ce cas, parce que dans celle-ci tous les capteurs envoient des intérêts et maintiennent des gradients même si rien n'est détecté.

La diffusion push à une phase (ou juste la diffusion push) a été conçue pour ce type d'application. Bien que les APIs soient identiques à celle de la diffusion à deux phases, les rôles des sources et des puits sont inversés. Les puits deviennent passifs et maintiennent les intérêts localement. Les sources deviennent actives et envoient les messages explorateurs de données dans tout le réseau sans l'utilisation de gradients. Cet algorithme peut également tirer profit de plusieurs types d'informations (géographiques ou relatives aux applications) pour des fins d'optimisation.

### ***1.5.9.3 La diffusion Pull à une phase***

Un avantage de la diffusion Push comparée à celle à deux phases est qu'elle utilise l'inondation dans un seul cas (messages explorateurs de données) plutôt que deux (des intérêts et messages explorateurs). Dans de grands réseaux la réduction de l'inondation peut être d'un avantage significatif. Motivé par l'efficacité du premier algorithme dans plusieurs types d'applications, cet algorithme a été amélioré en éliminant une de ses phases d'inondation.

À la différence du premier algorithme, quand un intérêt arrive à un nœud source, il ne marque pas son premier message de données comme explorateur, mais envoie les données seulement sur le gradient préféré. Le gradient préféré est déterminé par le premier voisin duquel a été reçu l'intérêt conforme aux données collectées, suggérant ainsi le chemin de latence minimale (cas particulier de l'algorithme original). En conséquence, l'algorithme n'exige pas des messages de renforcement, et le chemin de latence minimale est implicitement renforcé.

Cependant, l'algorithme exige que les messages d'intérêt transportent des identificateurs de flux. Bien que la génération d'une telle information soit assez simple, cette exigence rend la taille des intérêts grande avec l'accroissement du nombre des puits.

### **1.5.10 Discussion :**

La diffusion dirigée est un paradigme de routage dans les réseaux de capteurs. C'est une solution très prometteuse dans ce type de réseaux. Sa description fournit une liberté de conception aux utilisateurs selon le type d'applications et les conditions de l'environnement.

Dans la diffusion dirigée, l'adressage des nœuds du réseau n'est pas nécessaire du fait que chaque nœud communique avec son voisin direct. Chaque nœud peut agréger et mettre en cache les données issues des autres nœuds, en plus de sa tâche de captage.

L'utilisation du cache est fournit un grand avantage en terme d'efficacité en consommation de ressources. En outre, la diffusion dirigée est efficace en consommation d'énergie puisqu'elle utilise un schéma à la demande et il n'y a aucun besoin de maintenir la topologie globale du réseau.

La diffusion dirigée peut ne pas être appliquée à toutes les applications des réseaux de capteurs puisqu'elle est basée sur un modèle à base d'interrogation. Les applications qui exigent une livraison continue de données ne fonctionneront pas efficacement (grande consommation d'énergie) avec un modèle à la demande. Par conséquent, elle ne constitue pas un bon choix pour les applications telles que le contrôle de l'environnement.

La maintenance de routes est réalisée par la technique de renforcement. Les nœuds maintiennent les routes avec l'émission des messages d'intérêts en utilisant des petits taux.

Le schéma d'identification des données utilisé dépend fortement de la sémantique des applications et doit être défini a priori. D'autre part, le processus de correspondance entre les données et les attributs des requêtes pourrait exiger une charge de calcul supplémentaire.

La conception du paradigme ne prévoit pas son utilisation dans un réseau de capteurs avec un ou plusieurs puits mobiles, ou en cas où toutes les composantes du réseau sont mobiles [37].

*Plusieurs protocoles de routage basés sur le paradigme de la diffusion dirigée ont été proposés. Nous présenterons certains de ces protocoles visant à le rendre plus efficace en consommation d'énergie dans le troisième chapitre.*

### **1.6 Conclusion :**

Dans ce chapitre nous avons présenté le concept de réseau de capteurs et le problème de dissémination de données ou routage dans cet environnement. Dans la pratique, les réseaux de capteurs connaissent aujourd'hui un essor croissant dans plusieurs domaines d'applications. De façon générale, toutes les applications qui requièrent des opérations de contrôle dans des zones non directement surveillées. Dans plusieurs travaux de recherche, les réseaux de capteurs sont considérés comme un type particulier des réseaux sans fil Ad hoc. Une comparaison entre ces deux concepts a été aussi présentée dans ce chapitre.

Après avoir défini les réseaux de capteurs et décrit ses principales applications et caractéristiques, nous avons parlé du problème de dissémination de données dans ce type de réseaux, c'est à dire le problème de routage. La nature coopérative qui caractérise ces réseaux fait du routage une fonction très importante pour assurer l'échange des messages entre les nœuds capteurs et les puits. Ainsi, une classification des protocoles de routage a été donnée. Notamment les protocoles orientés données qui constituent une nouvelle classe de routage.

La diffusion dirigée étant le cœur de notre travail, nous avons présenté ce paradigme en détail et donné ses caractéristiques. Cette étude nous permettra de détecter les caractéristiques de dissipation d'énergie dans ce paradigme. Le protocole de routage n'étant pas le seul responsable de la consommation d'énergie dans un réseau de capteurs, nous présentons dans le chapitre suivant les niveaux de consommation et de conservation d'énergie dans un tel réseau.

# Chapitre 2

## La consommation et la conservation d'énergie dans les réseaux de capteurs

### 2.1 Introduction

Dans les réseaux Ad hoc classiques, la consommation d'énergie est un facteur important mais ne constitue pas la première considération pour les concepteurs, car les batteries sont supposées toujours remplaçables, les chercheurs ont cependant concentré leurs efforts sur les facteurs de la qualité de service tel que le débit et la tolérance aux pannes. Par contre, dans les réseaux de capteurs, l'efficacité en consommation d'énergie représente une métrique de performance significative, qui influence directement la durée de vie du réseau entier. Pour cela, les concepteurs peuvent au moment du développement de protocoles négliger les autres métriques, au détriment du facteur de consommation d'énergie. Toutes les solutions proposées dans les réseaux de capteurs partagent un objectif commun qui est la maximisation de la durée de vie du réseau [37].

La durée de vie du réseau dépend strictement de celle des nœuds le constituant. Celle d'un nœud est déterminée par sa batterie qui n'est, en général, ni remplaçable ni rechargeable. Elle peut être, en partie, alimentée par une unité génératrice d'énergie comme les cellules solaires. De petite taille, elle fournit donc une quantité d'énergie très limitée à l'échelle de 1 à 2J par nœud. Elle limite ainsi la durée de vie du capteur et influe sur le fonctionnement global du réseau.

La batterie doit alimenter tous les autres composants du nœud capteur. Ces derniers sont basés sur des technologies distinctes et diffèrent dans leurs caractéristiques de dissipation d'énergie. Ainsi, toute solution visant à conserver l'énergie dans un réseau de capteurs doit d'abord prendre en considération tous les facteurs qui caractérisent la consommation d'énergie dans les différentes étapes de fonctionnement. Ensuite déterminer un niveau de conservation qui peut être le nœud capteur ou l'ensemble du réseau.

Dans ce chapitre, nous présentons les niveaux de consommation et de conservation d'énergie dans un réseau de capteurs. Nous donnerons des analyses, en matière de consommation d'énergie, de deux types de nœuds capteurs réellement employés. Ensuite nous abordons les niveaux de conservation d'énergie dans ce type de réseaux tout en exposant les idées principales de quelques solutions proposées dans ce domaine.

### 2.2 La consommation d'énergie dans un nœud capteur

La première étape dans la conception d'un système de capteurs efficace en consommation d'énergie est l'analyse des caractéristiques de dissipation d'énergie d'un nœud capteur. Cette dernière est extrêmement importante pour identifier les maillons faibles du système, qui peuvent donc être le but d'une optimisation agressive.

### **2.2.1 La consommation d'énergie dans l'unité centrale de traitement**

Les performances en puissance des microprocesseurs ont beaucoup été étudiées, et plusieurs techniques ont été proposées pour estimer la consommation d'énergie dans ces processeurs [38, 39]. Le choix du type du microprocesseur est souvent lié au niveau de performance requis. Cependant, il peut significativement avoir un impact sur les caractéristiques de dissipation d'énergie du nœud. Par exemple, le microprocesseur StrongARM consomme approximativement 400 mW au moment de l'exécution des instructions alors que AVR ATmega103L consomme seulement 16.5 mW, mais ne fournit pas les mêmes performances. Donc, le choix du microprocesseur doit être fait selon le scénario de l'application afin de réaliser un compromis entre le niveau de performance offert par le microcontrôleur et celui demandé par l'application. De plus, les microprocesseurs supportent généralement plusieurs modes de fonctionnement : *Active*, *Idle*, et *Sleep*. La quantité d'énergie consommée diffère d'un mode à un autre. Par exemple, le microprocesseur StrongARM consomme 50 mW dans le mode *Idle*, et juste 0.16 mW dans le mode *Sleep*. Cependant, la transition entre les différents modes implique de l'énergie et de la latence supplémentaires.

En conséquence, les niveaux de consommation d'énergie dans les différents modes, les coûts de transitions et la durée de temps que passe le microprocesseur dans chaque mode agissent sur la durée de vie de la batterie et donc du nœud capteur.

### **2.2.2 La consommation d'énergie au niveau des capteurs**

Il y a plusieurs sources de consommation d'énergie dans un capteur comme le prélèvement du signal de l'événement capté, la conversion des signaux physiques en des signaux électriques, leur traitement et leur conversion (analogique vers numérique). Etant donnée la diversité des capteurs, il n'y a pas de modèle typique de consommation d'énergie. En général, les capteurs passifs (thermiques et sismiques) consomment une quantité d'énergie négligeable par rapport aux autres composants du nœud capteur à la différence des capteurs actifs (sonores, capteurs d'images) qui sont, par rapport, de grands consommateurs en terme d'énergie.

### **2.2.3 La consommation d'énergie dans le système de communication**

Plusieurs facteurs affectent les caractéristiques de la consommation d'énergie de la radio. Ces facteurs incluent le type de modulation utilisé, le taux d'envoi de données, la puissance de transmission (déterminée par la portée de communication) et le cycle de fonctionnement. En général, la radio travaille dans quatre modes différents : *transmission*, *réception*, *Idle* et *Sleep*. Une caractéristique importante pour la plupart des radios est que la quantité d'énergie consommée dans le mode *Idle* est importante et plus au moins égale à celle consommée dans le mode de *réception* [40]. Donc, il est préférable d'éteindre complètement la radio au lieu de passer en mode *Idle* quand il n'y a pas de transmission ou réception de données. Un autre facteur qu'il faut prendre en considération est la transition entre les différents modes. Par exemple, quand la radio passe du mode *Sleep* au mode de transmission pour envoyer un paquet, une quantité importante d'énergie est consommée pour mettre en marche le transmetteur [41].

### **2.2.4 Consommation d'énergie et modes de fonctionnement**

A un moment donné, le nœud capteur calcule, capte ou communique des données. La quantité d'énergie consommée diffère d'un mode de fonctionnement à un autre. Le tableau 1 montre les caractéristiques de la consommation d'énergie d'un nœud WINS de Rockwell

[42]. Ce nœud capteur est équipé d'un microprocesseur StrongARM SA-1100 de Intel, un module radio de Conexant Systems<sup>4</sup> et plusieurs capteurs acoustiques et sismiques. Le tableau 2 donne les caractéristiques de MEDUSA-II, un nœud expérimental développé au laboratoire Networked and Embedded Systems Lab, à l'université UCLA [5]. Le nœud MEDUSA est un nœud qui fonctionne à de basses puissances. Il est similaire au nœud COTS Motes développé dans le projet SmartDust [6]. Il est équipé d'un microprocesseur AVR de ATMEL, un module radio RFM et quelques capteurs.

Mode de l'unité de calcul	Mode des capteurs	Mode de la radio	La puissance (mW)
Active	On	Tx (puissance : 36.3 mW)	1080.5
		Tx (puissance : 19.1 mW)	986.0
		Tx (puissance : 13.8 mW)	942.6
		Tx (puissance : 3.47 mW)	815.5
		Tx (puissance : 2.51 mW)	807.5
		Tx (puissance : 0.96 mW)	787.5
		Tx (puissance : 0.30 mW)	773.9
		Tx (puissance : 0.12 mW)	771.1
Active	On	Rx	751.6
Active	On	Idle	727.5
Active	On	Sleep	416.3
Active	On	×	383.3
Sleep	On	×	64.0
Active	×	×	360.0

**Tableau 1 :** Analyse de la consommation d'énergie d'un nœud WINS

Les valeurs présentées dans les tableaux montrent les caractéristiques très différentes de la dissipation d'énergie des deux capteurs (analyse faite en 2002). A partir de ces valeurs, nous pouvons donner les observations suivantes :

- ▶ L'utilisation des composants à basse consommation d'énergie, en diminuant les performances non nécessaires durant la conception du nœud, peut avoir une grande influence sur la conservation d'énergie.
- ▶ La consommation d'énergie est entièrement dépendante des modes de fonctionnement des différents composants. Par exemple, comme le montre le

<sup>4</sup> <http://www.conexant.com/products>

tableau 1, le nœud WINS consomme seulement le un sixième de l'énergie quand le microcontrôleur est dans le mode *Sleep* que lorsqu'il est dans le mode *Active*.

- ▶ A cause des distances de transmission extrêmement petites qui caractérisent les nœuds capteurs, l'énergie de réception peut, dans certains cas, être plus grande que l'énergie de transmission. Donc, les protocoles réseaux conventionnels qui assument une énergie de réception négligeable ne sont pas applicables dans les réseaux de capteurs.
- ▶ L'énergie consommée dans le mode *Idle* est approximativement la même que celle consommée dans le mode de réception. Donc, faire basculer la radio en mode *Idle* ne fournit aucun avantage en terme d'énergie. La radio doit donc être complètement éteinte, quand il est possible, afin de réaliser un gain en énergie.

Mode de l'unité de calcul	Mode des capteurs	Mode de la radio	Schéma de modulation	Taux de données	Puissance (mW)
Active	On	Tx (transmission) (puissance : 0.7368 mW)	OOK (On/Off keying)	2.4 kb/s	24.58
		Tx (puissance : 0.0979 mW)	OOK	2.4 kb/s	19.24
		Tx (puissance : 0.7368 mW)	OOK	19.2 kb/s	25.37
		Tx (puissance : 0.0979 mW)	OOK	19.2 kb/s	20.05
		Tx (puissance : 0.7368 mW)	ASK (Amplitude-shift keying)	2.4 kb/s	26.55
		Tx (puissance : 0.0979 mW)	ASK	2.4 kb/s	21.26
		Tx (puissance : 0.7368 mW)	ASK	19.2 kb/s	27.46
		Tx (puissance : 0.0979 mW)	ASK	19.2 kb/s	22.06
Active	On	Réception	Quelconque	Quelconque	22.20
Active	On	Idle	Quelconque	Quelconque	22.06
Active	On	Off	Quelconque	Quelconque	9.72
Idle	On	Off	Quelconque	Quelconque	5.92
Sleep	Off	Off	Quelconque	Quelconque	0.02

**Tableau 2 :** Analyse de la consommation d'énergie d'un nœud MEDUSA-II

### 2.2.5 Les caractéristiques de la batterie

La batterie fournit de la puissance à l'ensemble des composants du nœud et donc joue un rôle principal dans la détermination de la durée du vie du nœud. Les batteries sont des composantes complexes qui dépendent dans leur fonctionnement de beaucoup de facteurs comme les dimensions, le type du matériel de l'électrode utilisé et le taux de diffusion des

matériaux actifs dans l'électrolyte<sup>5</sup>. De plus, plusieurs caractéristiques de fonctionnement de la batterie elle-même peuvent influencer la durée de vie du système :

### **1. Effet du taux de décharge**

Le facteur le plus important qui affecte la durée de vie du système est le taux de décharge ou bien la quantité du courant issu de la batterie. Chaque batterie a une capacité de décharge spécifiée par le constructeur. Si la quantité du courant déchargé dépasse le taux spécifié, une diminution significative dans la durée de vie de la batterie peut avoir lieu. Si la quantité du courant issu de la batterie est grande, le taux de diffusion des matériaux actifs sera inférieur au taux de leur consommation dans l'électrode. Si un grand taux de décharge est maintenu pour une longue durée, les électrodes vont fonctionner sans les matériaux actifs, ce qui a pour conséquence l'épuisement de la batterie même si les matériaux actifs sont encore présents dans l'électrolyte. Ainsi, pour éviter ce phénomène, la quantité du courant déchargé de la batterie doit être soigneusement vérifiée. Malheureusement, selon le type de batterie (lithium ion, NiMH, NiCd, alcaline, etc.), la consommation du courant requise des nœuds capteurs dépasse souvent le taux de décharge, ce qui résulte en des batteries de durée de vie non optimales.

### **2. Effet de relaxation**

L'effet du taux de décharge peut être allégé à travers la relaxation de la batterie. Si le courant déchargé par la batterie est stoppé ou réduit, le taux de diffusion et de transport des matériaux actifs augmente. Ce phénomène est appelé l'effet de relaxation et permet à la batterie de recouvrer une partie de sa capacité perdue. La durée de vie de la batterie est améliorée d'une manière significative si le courant déchargé par la batterie est fréquemment réduit à une très petite valeur ou complètement coupé [43].

#### **2.2.6 Le convertisseur DC-DC**

Le convertisseur dc-dc est responsable de fournir un voltage constant au reste du nœud capteur en utilisant toute la capacité de la batterie. Le facteur d'efficacité associé au convertisseur joue un rôle important dans la détermination de la durée de vie de la batterie [44]. Si le convertisseur n'est pas efficace, une perte d'énergie importante dans celui-ci peut avoir lieu, réduisant ainsi la quantité d'énergie disponible aux autres composants du nœud. Aussi, le niveau de voltage dans la batterie diminue quand celle-ci est déchargée. Le convertisseur déploie alors de grandes quantités de courant pour maintenir un voltage constant au nœud capteur. Comme résultat, le courant déchargé par la batterie devient progressivement plus grand que celui fourni au reste du nœud. Ceci mène à un épuisement de la batterie (effet du taux de décharge).

### **2.3 La conservation d'énergie dans les réseaux de capteurs**

Après l'analyse des caractéristiques de dissipation d'énergie d'un nœud capteur, nous allons focaliser notre attention sur la minimisation de l'énergie consommée dans le réseau. La conservation d'énergie peut être faite à différents niveaux soit au niveau de chaque nœud ou bien dans le réseau entier. Ce dernier cas implique un travail coopératif entre tous les nœuds.

---

<sup>5</sup> Dans une batterie, l'électrolyte est le matériel qui permet à l'électricité de découler d'un pôle à l'autre (entre les électrodes positives et négatives).

### **2.3.1 Optimisation d'énergie au niveau d'un nœud capteur**

Une première étape pour incorporer l'efficacité en énergie dans le réseau serait de développer des méthodologies de conception (matériel et logiciel) et des architectures systèmes qui permettent un fonctionnement et une conception efficace en consommation d'énergie des nœuds individuels dans le réseau.

#### **2.3.1.1 L'étape de calcul**

Les microprocesseurs intégrés dans les capteurs ne nécessitent généralement pas de grandes quantités d'énergie dans leur fonctionnement. En plus de l'utilisation de ces composants matériels durant la conception d'un nœud capteur, le fonctionnement des différents ressources systèmes, d'une façon efficace, à travers l'utilisation d'une gestion de puissance dynamique (DPM) [45] peut réduire davantage la consommation d'énergie, prolongeant ainsi la durée de vie de la batterie. Un schéma de gestion de puissance, fréquemment utilisé, est basé sur l'extinction des composants inactifs. Dans ce schéma, le nœud capteur (ou certaines de ses parties) est éteint ou mis dans des états dans lesquels il ne consomme pas beaucoup d'énergie, quand il n'y a pas des événements intéressants qui arrivent. Le principal défi dans ces approches est la détermination d'une politique de transition [45]. Notamment que les différents états sont caractérisés par une consommation d'énergie différente et les transitions d'état nécessitent une puissance et un temps non négligeables.

En plus de cette approche, une conservation d'énergie est possible dans l'état *actif* en utilisant un voltage dynamique [46]. La plupart des systèmes des microprocesseurs ont des charges de calcul variables dans le temps et donc l'utilisation des performances maximales du système n'est pas toujours essentielle. Les méthodes proposées dans cette catégorie adaptent dynamiquement le voltage fourni au processeur en sacrifiant les performances. Ces méthodes ont prouvé de meilleurs résultats par rapport aux méthodes DPM [46]. Plusieurs microprocesseurs modernes tels que StrongARM de Intel supportent le voltage dynamique.

Par exemple, dans une application de surveillance, un nœud peut être éteint pour une certaine durée de temps. Une conservation d'énergie peut être réalisée au prix de quelques valeurs manquantes. Cette méthode dépend de la qualité de service exigée par l'utilisateur (taux d'envoi).

#### **2.3.1.2 Les logiciels**

La durée de vie d'un réseau de capteurs peut être significativement améliorée si l'ensemble du système incluant le système d'exploitation, la couche application, les protocoles réseau sont tous conçus pour être efficaces en consommation d'énergie.

Le système d'exploitation est idéalement conçu pour supporter le voltage et la gestion dynamiques du fait qu'il a une connaissance globale des exigences de performances et de fidélité<sup>6</sup> de toutes les applications. Il peut donc contrôler directement les ressources matérielles. Au cœur du système d'exploitation, il y a un gestionnaire de tâche, qui est responsable de gérer un ensemble de tâches en s'assurant que les contraintes temporelles sont satisfaites. La durée de vie du système peut être considérablement augmentée par l'incorporation de la gestion d'énergie dans le gestionnaire de tâches [47, 48].

---

<sup>6</sup> La *fidélité* est définie par le degré auquel une copie de données présentée s'accorde avec la copie de référence (donnée originale). La *fidélité* a plusieurs dimensions qui sont utilisées comme des axes naturels pour l'adaptation. La consistance est une des dimensions les plus connues [52].

L'algorithme proposé dans [47] exploite deux caractéristiques des systèmes de communications sans fil pour fournir une puissance adaptative en fonction de la fidélité. La première est qu'ils sont conçus pour assurer différents niveaux de fidélité en permettant les pertes de données et les erreurs du médium sans fil. Cette capacité d'adaptation est donc utilisée dans la conservation d'énergie. La deuxième caractéristique réside dans le fait que ces systèmes présentent beaucoup de variations dans la charge de calcul et de communication dues aux variations des phénomènes physiques dans le temps. Cette caractéristique est exploitée dans la gestion proactive de l'énergie par la prédiction des exigences du système. Le voltage est spécifié selon les exigences prévues des tâches individuelles, et un contrôle adaptatif est effectué pour garder la fidélité du système dans les spécifications.

Le compromis fidélité-énergie peut être exploité en concevant la couche application pour être efficace en consommation d'énergie. Ceci peut être réalisé par la transformation des logiciels applicatifs tels que les calculs prioritaires seront effectués en premier. Donc, l'arrêt de l'algorithme prématurément pour des contraintes énergétiques n'aura pas un impact considérable sur les résultats.

Par exemple, dans une application de surveillance, plusieurs algorithmes de filtrage sont utilisés. Une conservation d'énergie peut être réalisée au prix de quelques précisions dans le filtrage de données.

### ***2.3.1.3 Gestion de puissance de la radio***

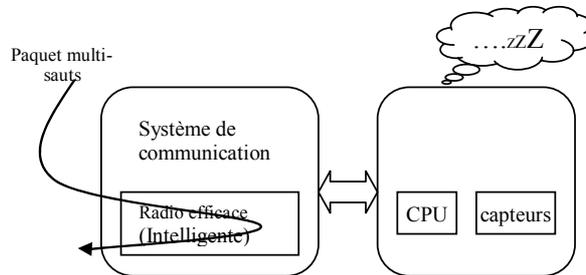
La gestion de la radio est extrêmement importante du moment que les communications sans fil sont très coûteuses en terme d'énergie. Pour considérer l'importance de ce problème, le ratio entre l'énergie consommée dans l'envoi d'un bit d'information et celle consommée dans l'exécution d'une instruction est calculé. Cependant, ce ratio ne peut être clairement calculé sans la normalisation de la portée de transmission, de la probabilité du bit d'erreur et de la complexité des instructions (8 ou 32 bits). Des exemples de valeurs pour ce ratio sont : de 1500 à 2700 pour le nœud WIN de Rockwell, de 220 à 2900 pour le nœud MEDUSA II et 1400 pour le nœud WINS NG 2.0 (ce nœud est utilisé par la plupart des chercheurs [5]).

Deux principaux composants sont responsables de la consommation d'énergie dans la radio [13]: le composant RF (radio frequency), qui dépend de la distance de transmission et des paramètres de modulation, les composants électroniques (les circuits) qui effectuent la synthèse de la fréquence, le filtrage et la conversion, etc. Les techniques utilisées telles que le voltage et la gestion dynamique de la puissance ne sont pas directement applicables dans le cas de la radio. Ces techniques conservent de l'énergie au prix de la latence du calcul. Dans ce cas, par contre, les puissances des deux composants sont comparables. Ainsi, ralentir la radio peut mener à une augmentation dans la consommation d'énergie. D'autres facteurs spécifiques à l'architecture peuvent compliquer le problème tels que la puissance d'initialisation des transmetteurs [41].

#### ***a. La retransmission efficace des paquets***

En plus du captage et de la communication de ses propres données aux autres nœuds du réseau, le nœud capteur agit dans la plupart des cas comme un routeur retransmettant les paquets à d'autres nœuds. En effet, pour des scénarios typiques des réseaux de capteurs, une grande portion (environ 65%) des paquets reçus ont besoin d'être retransmis à d'autres destinations [49]. Les architectures des réseaux de capteurs implémentent la plupart des fonctionnalités des protocoles dans l'unité de calcul centrale. Donc, quelque soit la destination finale d'un paquet, celui-ci traverse tout le chemin menant à l'unité de calcul

où il est traité ce qui implique une consommation d'énergie importante. L'utilisation d'un matériel radio intelligent, comme montré dans la figure suivante (figure 2.1), permet l'identification des paquets à retransmettre et leur redirection directe. Ceci permet à l'unité de calcul de rester dans un état inactif et donc de consommer moins d'énergie [49].



**Figure 2.1 :** La retransmission efficace

### ***b. Les schémas de modulation***

En plus de l'architecture matérielle, la technologie de la radio utilisée dans les liens sans fil entre les nœuds capteurs joue un rôle principal dans les considérations de l'énergie. Le choix du schéma de propagation influence le compromis entre l'énergie, la fidélité et la latence. L'équation suivante exprime le coût d'envoi d'un bit d'information, comme une fonction de la taille du paquet  $L$ , la taille de l'entête  $H$ , une charge (overhead) fixe  $E_{start}$  associé à l'initialisation de l'antenne, un taux  $R_s$  pour un schéma de modulation de vecteur de taille  $M$  [41, 50].  $P_{elec}$  représente la consommation en puissance des composants électroniques (filtrage, conversion, ...). La puissance délivrée par l'amplificateur  $P_{RF}$ .

$$E_{bit} = \frac{E_{start}}{L} + \frac{P_{elec} + P_{RF}(M)}{R_s * \log_2 M} * \left(1 + \frac{H}{L}\right)$$

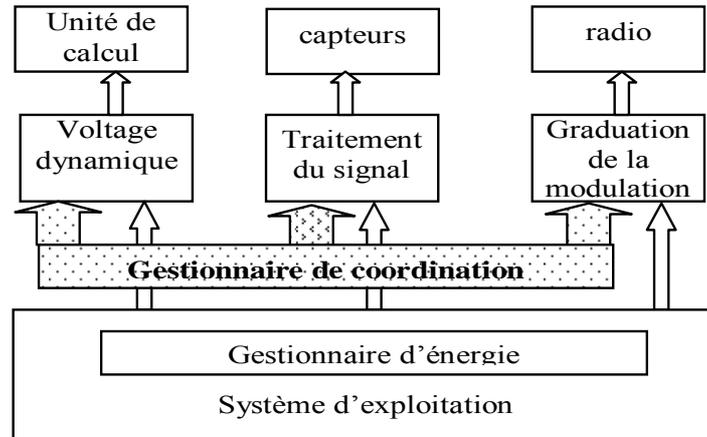
Plusieurs études ont été menées pour déterminer un schéma de modulation qui soit efficace en consommation d'énergie. Pour ce faire, des valeurs réelles (des technologies des réseaux de capteurs) ont été données aux différents paramètres [13]. Les résultats obtenus indiquent les niveaux de modulation optimaux pouvant être utilisés dans la transmission des données. La conservation d'énergie peut alors être faite en :

- ▶ Adaptant le niveau de modulation selon la charge du trafic. Cette technique d'adaptation dynamique du niveau de modulation selon la charge instantanée du trafic, comme partie de la gestion de puissance de la radio, est appelée graduation de la modulation [50].
- ▶ Utilisant de grandes tailles pour les paquets manipulés. Ceci diminue le coût d'initialisation de la radio.
- ▶ Utilisant de petits niveaux de modulation dans le cas où les nœuds capteurs interagissent avec des passerelles avec de longues distances de communications [51].

#### ***2.3.1.4 La coordination au niveau du nœud***

Le nœud capteur peut utiliser toutes les techniques déjà citées à différents niveaux pour réaliser un compromis entre le calcul, les communications et la consommation d'énergie. Dans la figure 2.2, le nœud utilise les techniques déjà citées à différents niveaux

de son architecture dans le but de réaliser des compromis entre la qualité de service et la quantité d'énergie consommée [13].



**Figure 2.2 :** La coordination au niveau des composants d'un nœud

### **2.3.2 Conservation de l'énergie dans le réseau**

L'incorporation de la gestion d'énergie au niveau de chaque nœud seulement ne résout pas le problème de l'efficacité en énergie dans les réseaux de capteurs [13]. Ainsi, la conservation de l'énergie exige la coopération entre tous les nœuds du réseau. Les décisions de gestion de puissance qui prennent en considération l'effet des communications inter-nœuds mènent sûrement à une conservation d'énergie plus importante. Ces décisions permettent de transporter l'efficacité en consommation d'énergie d'un nœud individuel à un ensemble de nœuds ce qui permet l'augmentation de la durée de vie de régions entières dans le réseau.

Dans cette optique, plusieurs solutions ont été proposées. Nous allons présenter dans cette partie quelques solutions suggérées au niveau des couches liaison et réseau.

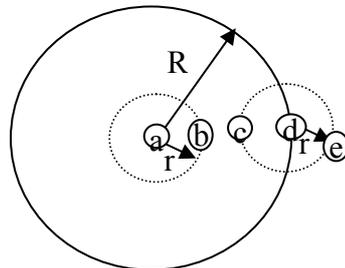
#### **2.3.2.1 Le contrôle de la puissance de transmission**

Le contrôle de la puissance de transmission est intégré au niveau du protocole MAC (Medium Access Control) ou au niveau du protocole de routage, ou parfois placé externe au système. Cette technique est d'une grande importance pour différentes raisons : premièrement, l'ajustement de la puissance est efficace pour garantir la connectivité. Deuxièmement, le contrôle de puissance est essentiel pour la gestion de la topologie et la réutilisation spatiale du spectre. Finalement, réduire la puissance de transmission peut réduire la consommation d'énergie, directement, en exigeant moins de puissance pour envoyer, et indirectement, en réduisant les conflits de transmissions [53].

Il existe beaucoup de travaux qui tentent d'analyser et de concevoir des protocoles pour le contrôle de puissance dans l'environnement sans fil. Ranathan et Rosales-Hain [54] ont montré comment déterminer la puissance de transmission pour réaliser un équilibre entre la connectivité et l'énergie consommée. Beaucoup approchent le problème du point de vue MAC. Un protocole MAC (PCMA) représentatif qui considère le contrôle de puissance est présenté dans [56], il se base sur l'optimisation de la réutilisation spatiale du canal et l'utilisation d'une puissance de transmission variable. Le protocole fournit environ

50% de throughput<sup>7</sup> de plus quand les noeuds et le trafic sont groupés et le contrôle de la puissance de transmission est utilisé.

Peu de chercheurs se concentrent sur le contrôle de la puissance de transmission pour réduire la consommation de l'énergie. L'intérêt est plus souvent porté sur la connectivité et réutilisation spatiale. Ces facteurs constituent des approches cruciales dans la conception des systèmes de communication, en particulier ceux de longues distances. Kaiser et Pottie [59] ont suggéré l'utilisation de la transmission de courte distance afin de réduire le coût de communication qui est proportionnel à la distance. La puissance de transmission radio peut être une partie significative de la consommation d'énergie sur des petites distances, mais d'autres facteurs peuvent dominer s'ils ne sont pas pris en compte. Cependant, le coût fixe de l'écoute du médium rend les différences de puissance de transmission non significatives. Par exemple, si 2% du temps sont dépensés dans la transmission, la conservation d'énergie maximale est de 8% seulement [53]. L'évitement des collisions par la réutilisation spatiale conserve des quantités d'énergie considérables parce qu'en cas de collisions les deux participants (récepteur et transmetteur) doivent retransmettre. La figure (2.3) suivante illustre ces concepts en considérant deux puissances de transmissions,  $r$  et  $R$  où  $R \approx 3r$ . Pour la communication du nœud  $a$  au nœud  $d$ , les deux peuvent transmettre soit directement (un saut) en utilisant la puissance  $R$  ou bien indirectement (trois sauts) en utilisant la puissance  $r$ . En utilisant un modèle d'énergie simple (coût proportionnel au carré de la distance), les coûts relatifs à ces transmissions sont  $1 \cdot 3^2 = 9$  pour un saut avec  $R$  et  $3 \cdot 1^2 = 3$  pour trois sauts avec  $r$ . Ceci démontre la conservation d'énergie qui peut se faire avec une communication plus courte et multi-sauts. Cet exemple montre aussi la possibilité de la réutilisation spatiale et la réduction des conflits fournies par la transmission à basse puissance.



**Figure 2.3 :** Le contrôle de la puissance de transmission

Le contrôle de puissance de transmission est souvent vu comme une partie de la couche du routage. Un exemple de protocole dans ce domaine est LEACH (protocole hiérarchique) [28]. Au lieu d'envoyer directement l'information à un site central, les nœuds forment des clusters et les données sont envoyées par un routage multi-sauts. En effectuant une rotation des têtes de clusters dans le temps, la consommation d'énergie est réduite et éventuellement distribuée.

Les études des interactions entre le contrôle de puissance et les protocoles de routage indiquent leur importance dans la réalisation d'un système global fiable et efficace [60].

### **2.3.2.2 Gestion de la couche MAC**

Les travaux effectués au niveau de la couche MAC supposent généralement une puissance de transmission fixe. Ceci laisse quatre sources de consommation d'énergie à

<sup>7</sup> Le throughput est le taux de sortie du système.

explorer : les collisions, l'écoute passive du médium (*idle listening*), l'écoute des messages envoyés entre d'autres nœuds (*overhearing*) et la communication des paquets de contrôle qui ne transportent pas de l'information utile. Plusieurs approches ont été proposées pour réduire ces coûts : l'approche TDMA, les protocoles à base de contention avec des périodes de contention programmées, les schémas asynchrones, les canaux numérotés (paging channels), et l'écoute à basse puissance [53].

Les protocoles de multiplexage temporel TDMA (Time-division multiple-access) peuvent réduire les coûts des collisions, de l'écoute passive et de l'*overhearing* en programmant les périodes de transmission et de l'écoute. Ces protocoles exigent une synchronisation stricte du temps, souvent fournie par une infrastructure telle que la station de base. La couche MAC de IEEE-802.11 incorpore un intervalle à contention libre qui adopte une structure TDMA gérée par le point d'accès [61], réduisant chacun des trois coûts cités. Les communications et les interférences inter-groupes sont manipulées en utilisant la technique CDMA (Code Divison Multiple Access). Sohrabi et al. [51] ont proposé un protocole d'organisation de transmission point-à-point dans les réseaux de capteurs. Leur approche élimine les stations de base, mais elle se base sur l'allocation de différents canaux CDMA ou FDMA (Frequence Divison Multiple Access) pour n'importe quels liens interférents pour permettre les transmissions concurrentes ce qui résulte en une utilisation non optimale des canaux.

Les protocoles à base de contention constituent une autre classe des protocoles MAC. Ces protocoles essaient d'alléger les contraintes de synchronisation serrées des protocoles TDMA. Ils emploient une technique CSMA (Carrier Sense Multiple Access) pour l'accès au canal. Cependant, ces protocoles n'évitent pas les collisions du fait que les nœuds du réseau accèdent concurrentement au canal. Ainsi, les coûts de l'écoute passive et de l'*overhearing* ne sont pas forcément réduits. Le protocole IEEE-802.11 Ad hoc est un exemple dans cette catégorie très employé actuellement. Il utilise une technique CSMA et des paramètres aléatoires pour réduire la probabilité des collisions [61]. Pour réduire l'écoute passive, il utilise un mode d'économie d'énergie (PSM), permettant aux nœuds d'entrer périodiquement dans l'état *Sleep*. Le PSM suppose un réseau à un seul saut et donc la synchronisation est plus au moins facile. Dans un réseau multi-sauts plusieurs problèmes peuvent survenir tels que la synchronisation d'horloge et la découverte de voisins [62]. Pour résoudre le problème de l'*overhearing*, PAMAS propose l'utilisation de deux canaux, le premier pour le contrôle de trafic et le deuxième pour les données [63]. La solution consiste à garder le canal de données éteint quand les autres nœuds s'échangent des paquets de contrôle.

Les protocoles à contention programmés constituent une sous classe des protocoles à contention. En plus du PSM en mode 802.11 Ad hoc, le S-MAC (Sensor MAC) est un deuxième protocole dans cette classe [64]. Dans S-MAC, chaque nœud adopte un cycle d'écoute/sommeil. La contention se produit seulement pendant une courte période d'écoute, réduisant le coût d'écoute passive. Pendant la transmission de données, les nœuds autres que la source et les destinations entrent dans des périodes de sommeil afin d'éviter les coûts d'*overhearing*. Récemment, l'écoute adaptative [65] et T-MAC [66] ont été proposés dans le but d'améliorer la transmission multi-sauts avec des protocoles MAC à cycle de sommeil.

Les schémas asynchrones forment une autre classe des protocoles MAC. Ils se basent sur des modèles de réveil et de sommeil asynchrones [62]. Quelques protocoles viennent étendre les protocoles MAC déjà existant. Ces protocoles éliminent complètement les exigences de la synchronisation. Leur inconvénient majeur est l'inefficacité de la diffusion (broadcast), puisque tous les nœuds se réveillent indépendamment.

Les canaux numérotés forment une autre approche proposée pour réduire la consommation d'énergie : la radio primaire est laissée éteinte quand il n'y a pas de trafic, et une autre radio à basse puissance (le canal numéroté) est utilisée pour réveiller les nœuds quand il y a des données à envoyer. L'avantage d'employer une deuxième radio est la capacité d'éviter complètement les interférences dans les transmissions sur la radio principale. STEM [67] est un protocole de réveil à la demande appartenant à cette catégorie.

L'écoute à basse puissance a été généralisée pour fonctionner comme mécanisme de conservation d'énergie avec une seule radio [68]. Un nœud en sommeil se réveille périodiquement et vérifie le médium. Il reste en mode actif seulement quand une activité est détectée. L'expéditeur réveille le récepteur en envoyant des paquets avec un préambule aussi long que la période de vérification. L'avantage d'une telle approche est que des périodes de vérification trop petites peuvent être utilisées, aussi petites que 3 ms sur des Motes Mica2 [68]. L'inconvénient est que les préambules peuvent être extrêmement longs.

En résumé, les protocoles MAC tels que TDMA, évitent des collisions, et peuvent facilement réduire les coûts de l'écoute passive et *l'overhearing*. Cependant ils peuvent être un mauvais choix dans les réseaux multi sauts en raison de l'utilisation inégale d'énergie au niveau des nœuds et au besoin de synchronisation stricte. Les protocoles à base de contention n'ont pas ces inconvénients, mais ils ne réduisent pas vraiment les coûts de collisions, de l'écoute passive et de *l'overhearing*. Des extensions de ces protocoles peuvent réduire le coût de l'écoute passive en utilisant des techniques comme : les périodes de contention programmées, les schémas asynchrones, les canaux numérotés et l'écoute à basse puissance, chacune avec ses propres avantages et inconvénients.

### **2.3.2.3 Les protocoles de contrôle de topologie : entre le MAC et le routage**

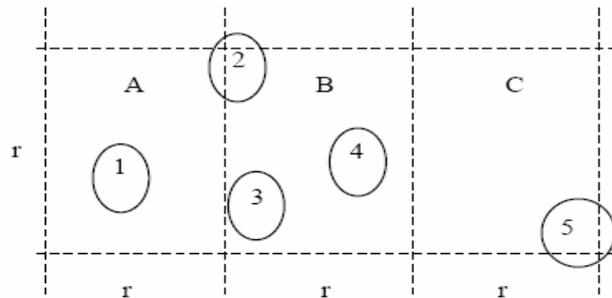
Les protocoles de contrôle de topologie sont une classe de protocoles situés entre la couche MAC et la couche du routage qui proposent d'éteindre les nœuds pour de longues périodes de temps. Ils suggèrent d'arrêter le plus grand nombre de nœuds possible pour conserver l'énergie. Les données peuvent circuler dans le réseau et les différents nœuds peuvent se connecter au réseau pour envoyer des données. Cependant, les nœuds éteints ne peuvent pas être des destinations finales de données.

Les protocoles de contrôle de topologie viennent compléter les protocoles de la couche MAC. Ils fonctionnent similairement en utilisant des intervalles de temps beaucoup plus grands, faisant des cycles de temps de l'ordre de minutes plutôt que des secondes. Cette granularité réduit les coûts de mise en marche de l'interface radio et n'exige pas une synchronisation stricte.

L'inconvénient principal de ces protocoles est que des nœuds particuliers vont être éteints et donc non accessibles pour de longues périodes de temps. Cet inconvénient est à considérer si les nœuds sont explicitement adressés. Cependant, cette restriction aura peu d'impact dans un réseau de capteurs où toutes les communications sont orientées données. Une requête dans un tel réseau peut être satisfaite par n'importe quel ensemble de nœuds en marche.

Les protocoles proposés dans cette catégorie visent à construire un réseau de connexion à partir du réseau initial. Chaque solution diffère dans le choix des nœuds qui formeront un tel réseau. Le protocole GAF (geographic Adaptive Fidelity) utilise la localisation des nœuds pour déduire le réseau final [69]. L'idée est d'éteindre les nœuds inutiles dans le réseau pour conserver leur énergie. L'algorithme forme une grille virtuelle en se basant sur la portée de communication (zone de couverture). Chaque nœud utilise

son information de localisation pour s'associer avec un point dans la grille virtuelle. Les noeuds associés au même point sur la grille sont considérés équivalents en terme de coût du routage d'un paquet. Une telle équivalence est exploitée en maintenant quelques noeuds situés dans une région particulière de la grille dans un état de sommeil. Une telle situation est décrite dans la figure (2.4) suivante. Dans cette figure, le noeud 1 peut atteindre les noeuds 2, 3 et 4. Les noeuds 2, 3, et 4 peuvent atteindre le noeud 5. Par conséquent les noeuds 2, 3 et 4 sont équivalents et deux d'entre eux peuvent être éteints.



**Figure 2.4 :** Grille virtuelle dans GAF

Une autre solution consiste à calculer directement la connectivité du réseau et élire ainsi des coordonneurs pour garantir la couverture. Dans SPAN [70], un noeud devient un coordonneur si n'importe quel noeud de ses voisins ne peut atteindre un autre voisin directement ou via un ou deux coordonneurs. Cet algorithme d'élection exige l'échange de l'information de voisinage entre les noeuds du réseau. La connectivité est alors calculée par chaque noeud en se basant sur cette information. CEC [71] est une extension de SPAN qui prend en considération la mobilité des noeuds.

Les protocoles de contrôle de topologie peuvent interagir ou être indépendants des protocoles MAC et routage. GAF et CEC sont indépendants des deux couches, mais le problème se pose quand ces protocoles mettent un noeud sur un lien de routage en état de sommeil. Ces problèmes peuvent être évités en adaptant la couche du routage ou échangeant explicitement des signalisations entre le protocole de contrôle de topologie et les autres couches [67].

#### **2.3.2.4 Le routage**

Le protocole de routage définit les interactions entre plusieurs noeuds dans un réseau multi-sauts. Ainsi, plusieurs objectifs différents peuvent être réalisés par le routage. Singh et al. décrivent [73] cinq objectifs possibles. Nous récapitulons quatre ici : (1) minimiser l'énergie totale consommée, (2) maximiser le temps avant le partitionnement du réseau, et (3) minimiser la variance d'énergie entre les noeuds, ou (4) minimiser le 'coût' par paquet. Minimiser directement l'énergie consommée peut être contradictoire avec les deux buts 2 et 3, du fait que le chemin d'énergie minimale peut concentrer le trafic au niveau de certains noeuds. Le dernier objectif cherche à réaliser un équilibre, en définissant le 'coût' comme une fonction de la durée de vie restante de la batterie. Cependant, les auteurs [73] ont indiqué que l'optimisation des coûts est un problème NP-complet. (Chang et Tassiulas l'ont formulé à l'aide d'un problème de programmation linéaire dont la résolution est sous forme polynomiale [74].)

Les algorithmes de routage traditionnels tels que Bellman-Ford ou Dijkstra optimisent une certaine métrique, telle que le chemin le plus court. Dans les réseaux sans fil, cette métrique peut être le nombre de sauts ou une autre mesure de latence, comme dans DSDV [75] et DSR [76]. Ces métriques peuvent être généralisées pour minimiser l'énergie

consommée dans chaque saut (ensemble de sauts). En considérant les coûts de chaque saut comme fonction de la durée de vie restante des nœuds, il devient possible de gérer le routage dans des réseaux hétérogènes où les nœuds ont des quantités d'énergie restantes différentes. Djenouri et al. [72] ont considéré les métriques de routage comme une fonction de l'énergie résiduelle des nœuds et du coût de communication.

Un autre facteur qui affecte la consommation d'énergie dans le routage est la qualité des liens. Les retransmissions au niveau liaison peuvent augmenter le coût énergétique d'une manière significative. Un chemin court avec beaucoup de retransmissions est plus mauvais qu'un plus long chemin avec plus de sauts mais peu de retransmissions. Ce problème peut être résolu en excluant les mauvais chemins du choix des routes. Une autre solution serait d'inclure la qualité des liens dans la métrique du routage. Banerjee et Misra [77] ont considéré la puissance de transmission variable et la qualité de liens dans chaque saut. Le coût d'un chemin est une combinaison de la distance et du taux d'erreur. Woo et al. [78] ont examiné, de leur part, l'interaction entre le routage et la qualité des liens sur des nœud capteurs (Motes Mica2). Ils ont proposé d'employer le nombre total de transmissions (y compris les retransmissions) comme métrique de routage.

En résumé, les recherches menées dans le domaine du routage efficace en consommation d'énergie sont principalement orientées sur deux aspects : minimiser le coût énergétique par paquet et équilibrer la consommation d'énergie dans le réseau. Les protocoles de contrôle de topologie et ceux de la couche MAC peuvent fortement influencer la conception de la couche du routage.

## **2.4 Conclusion**

Dans ce chapitre, nous avons analysé la consommation d'énergie des différents composants d'un nœud capteur dans leurs différentes étapes de fonctionnement. Un aperçu des différentes solutions à plusieurs niveaux de l'architecture a aussi été donné. Nous pouvons dire que l'efficacité en consommation d'énergie doit constituer un objectif dans toutes les étapes de conception. S'étendant ainsi du niveau nœud au niveau réseau où tous les nœuds collaborent dans le processus de conservation, dans ce dernier cas, la conservation d'énergie vise en premier lieu les communications entre les nœuds.

Les réseaux de capteurs sont des réseaux orientés données où toutes les communications sont centrées sur les données captées. Un protocole de routage qui soit efficace en consommation d'énergie doit obligatoirement maximiser la durée de vie du réseau. Dans le chapitre suivant, nous concentrons notre attention sur les solutions de conservation d'énergie dans la phase de routage. Particulièrement, les protocoles qui visent à réaliser une efficacité en consommation d'énergie dans la diffusion dirigée.

# Chapitre 3

## La conservation d'énergie dans la diffusion dirigée

### 3.1 Introduction

Le paradigme de la diffusion dirigée est à la base de plusieurs solutions proposées ultérieurement dans les réseaux de capteurs. Ce paradigme a été instancié pour produire beaucoup de protocoles dans plusieurs types d'applications. Pour ce faire, les changements proposés sont basés sur différents paramètres et visent différentes étapes du paradigme.

L'importance du facteur de la consommation d'énergie dans les réseaux de capteurs a orienté beaucoup de travaux menés sur la diffusion dirigée. Ainsi, plusieurs solutions ont été proposées essayant de réaliser une conservation d'énergie dans ce paradigme. Ces solutions visent à réaliser une efficacité en énergie dans les différentes étapes du protocole. Ainsi, les changements proposés s'étendent de la phase d'inondation jusqu'à la phase de routage en essayant de minimiser la consommation d'énergie dans ces étapes. Dans ce chapitre nous présentons un état de l'art de ses solutions. Nous essayons durant notre étude de données les idées principales de chaque solution, ses apports, et les phases de la diffusion dirigée visées.

### 3.2 La consommation d'énergie dans la diffusion dirigée

La diffusion dirigée est un exemple de protocole de routage efficace en consommation d'énergie [93]. Son efficacité résulte de deux points importants : (1) La nature réactive dans la découverte des chemins et la dissémination de données. (2) L'utilisation des mécanismes d'agrégation et de cache sur les données routées.

Cependant, le schéma du protocole souffre encore de plusieurs limitations dans la consommation d'énergie. Le protocole consiste en une phase d'inondation suivie d'une phase de renforcement de chemins et d'une phase de routage. L'inondation porte sur les intérêts et les messages explorateurs de données et est considérée comme la phase la plus coûteuse en terme d'énergie. La phase de renforcement a pour objectif de maintenir la topologie du réseau. Elle consiste en l'envoi de messages de données en utilisant de petits taux d'envoi, ce qui est aussi coûteux dépendant du nombre de nœuds dans le réseau et surtout du nombre des gradients au niveau des nœuds. Les nœuds peuvent être emmenés à effectuer une inondation dans tout le réseau quand il y a des défaillances de liens. La consommation d'énergie dans la phase de routage dépend des critères de choix des chemins utilisés ainsi que leur utilisation. Dans la diffusion dirigée, les chemins sont souvent renforcés selon la règle du plus court chemin (latence) ou bien de sa consistance [18]. Le calcul des chemins selon des métriques qui réalisent une efficacité en consommation d'énergie n'a pas été proposé dans la version originale du protocole. Aussi, l'utilisation d'un seul chemin pour router les données vers le puits peut mener à un épuisement des nœuds le constituant et causer alors le partitionnement du réseau.

### 3.3 Les protocoles de conservation d'énergie dans la diffusion dirigée

#### 3.3.1 Le Rumor routing

Le protocole proposé dans [22] est une variante de la diffusion dirigée et est principalement destiné aux contextes où le routage géographique n'est pas applicable soit parce qu'un système de localisation n'est pas disponible ou que les intérêts ne s'intéressent pas aux informations géographiques. Dans la diffusion dirigée, les requêtes sont inondées dans tout le réseau quand il n'y a pas de critères géographiques à suivre. Cependant, dans quelques cas l'inondation devient inutile par rapport à la quantité de données demandées. Une approche alternative est de diffuser les événements si le nombre d'évènements est petit et le nombre de requêtes est grand. Le 'rumor routing' traite alors le cas d'application où le volume de données retournées vers l'utilisateur n'est pas important. Il estime que l'envoi de ces données via des chemins non-optimaux serait plus efficace que l'investissement dans une étape d'initialisation, visant à rechercher les chemins optimaux, via l'inondation d'un grand nombre de requêtes relativement supérieur au petit nombre d'évènements détectés.

Le pseudo-algorithme de ce protocole est le suivant :

- Chaque noeud maintient une liste de ses voisins, et également une table d'évènements. La liste des voisins peut être créée et maintenue d'une manière active en diffusant une requête ou d'une façon passive, par l'écoute des émissions des autres nœuds.
- Quand un noeud distingue un événement, il l'ajoute à sa table d'évènements, avec une distance de zéro à l'évènement. Il produit également d'une façon probabiliste un agent. La probabilité de produire un agent est un paramètre de l'algorithme.
- Un agent est un paquet particulier, qui voyage dans le réseau, propageant des informations sur les événements locaux des noeuds éloignés. Il contient une table d'évènements, semblable à celle des noeuds, qu'il synchronise avec chaque noeud qu'il visite. L'agent voyage dans le réseau pour un certain nombre de sauts ensuite il meurt.
- N'importe quel noeud peut produire une requête, qui devrait être routée à un événement particulier. Le noeud transmet la requête s'il connaît le chemin menant à l'évènement. Sinon, il l'enverra dans une direction aléatoire. Ceci continue jusqu'à ce que la requête expire ou atteint un noeud qui a observé l'évènement ciblé. Dans certains cas le noeud ne transmet pas la requête même s'il connaît la route (détection de boucle).
- Si le noeud originaire de la requête conclut que sa requête n'a pas atteint la destination (après un certain temps, paramètre de l'algorithme), il peut la retransmettre, l'inonder ou même l'abandonner.

Les résultats de simulation montrent que cette solution peut atteindre une optimisation considérable en consommation d'énergie grâce à la technique d'inondation des événements qui remplace l'inondation des requêtes dans la diffusion dirigée. Ces performances ne sont, cependant, valables que pour un nombre faible d'évènements. Dans le cas contraire, le coût nécessaire pour la maintenance des agents et des tables d'évènements au niveau des nœuds pourrait ne pas être amorti. Particulièrement, si peu d'intérêts liés à ces événements sont diffusés par le nœud puits. De plus, la minimisation de l'*overhead* nécessaire par

l'ajustement des différents paramètres du protocole reste une question importante à considérer.

*Cette solution vise principalement la phase d'inondation dans la diffusion dirigée en proposant de réduire le nombre des inondations des intérêts. Cependant, elle ne cite aucune modification dans les autres phases du protocole.*

### **3.3.2 Routage à base de gradient**

Schurgers et al. [21] ont proposé une simple instanciation de la diffusion dirigée appelée Gradient based routing (GBR). L'idée est de mémoriser le nombre de sauts au moment de la diffusion des intérêts dans le réseau. Ainsi, chaque noeud peut découvrir le nombre minimum de sauts menant au puits. Ce nombre est appelé taille du noeud. La différence entre la taille d'un noeud et celle de son voisin est considérée comme le gradient sur ce lien. Un paquet est expédié sur un lien ayant la plus grande valeur du gradient.

Les auteurs emploient avec GBR certaines techniques auxiliaires pour l'agrégation et la propagation uniforme de données. La première est faite en définissant des nœuds (entités) combinateurs de données (ECD) afin d'équilibrer le trafic dans le réseau. Ces nœuds compactent l'ensemble des flux de données qui les traversent. Quand un nœud EDC tombe en panne (par exemple panne d'énergie), les paquets prennent automatiquement une route alternative et passent à travers un autre nœud qui peut constituer un EDC. D'autre part, trois techniques différentes de propagation de flux de données ont été proposées :

- Le schéma stochastique : Quand il y a deux ou plus de voisins d'un saut avec le même gradient, le noeud choisit aléatoirement un d'entre eux.
- Le schéma à base d'énergie : Quand l'énergie d'un noeud passe en dessous d'un certain seuil (50% par exemple), il augmente sa taille de sorte que d'autres capteurs soient découragés d'envoyer des données en sa direction. Le changement de la taille d'un nœud doit être propagé dans tout le réseau pour garder un état consistant du gradient.
- Le schéma à base de flux : L'idée est de détourner les nouveaux flux loin des noeuds qui font déjà partie des chemins sur lesquels passent d'autres flux. Pour ce faire, un nœud qui reçoit un flux informe tous ses voisins, sauf celui duquel il l'a reçu, que sa taille a augmenté. En gardant toujours un état consistant du gradient, le nœud va continuer à recevoir le flux original alors que les autres flux suivront d'autres chemins.

Les schémas de propagation de données proposés dans cette solution visent à réaliser une distribution uniforme du trafic dans tout le réseau. Ceci aide à équilibrer la charge sur les noeuds capteurs et augmente la durée de vie du réseau. Les techniques utilisées pour l'équilibrage des charges et la fusion de données sont également applicables à d'autres protocoles pour améliorer leurs performances.

Les résultats de simulation montrent que cette solution offre de meilleures performances en terme de consommation d'énergie en appliquant les deux propositions (l'utilisation des ECDs et les schémas de propagation).

*Cette solution ne vise pas l'étape d'inondation. Celle-ci est utilisée, telle que proposée dans la version originale de la diffusion dirigée, comme étape d'initialisation pour déduire les tailles des nœuds. Cependant, elle vise l'étape de renforcement en proposant des renforcements basés sur les gradients des liens et l'étape de routage en proposant*

*d'utiliser plusieurs chemins selon un schéma de propagation. Les entités combinatoires de données sont aussi une source d'efficacité en énergie dans cette solution.*

### **3.3.3 La diffusion dirigée avec routage géographique (GEAR)**

Yu et al. [25] ont suggéré l'utilisation de l'information de localisation dans la diffusion dirigée dans le but de conserver l'énergie requise pour l'étape d'inondation des intérêts. Le nombre de ces derniers est limité en considérant seulement une certaine région plutôt que le réseau entier. Le protocole, appelé Geographic and Energy Aware Routing (GEAR) complète la diffusion dirigée de cette façon et conserve ainsi plus d'énergie. Il suppose que les requêtes incluent toujours des attributs géographiques (le champ rect.) et donc la dissémination sera faite dans les régions adéquates en utilisant les valeurs de ces attributs. Cette solution utilise des heuristiques de sélection de voisins en se basant sur le facteur énergie pour router un paquet vers une région ciblée, et fait appel à la diffusion géographique récursive pour router le paquet à l'intérieur de la région. Le processus de transmission d'un paquet à tous les nœuds se trouvant dans une région passe par deux étapes :

#### **1. La transmission du paquet vers la région ciblée**

Dans cette étape GEAR utilise les informations de localisation en plus de celles de l'énergie relatives à ses voisins. Pour ce faire, chaque nœud  $N$  maintient un état  $h(N, R)$  appelé le coût d'apprentissage vers la région  $R$ . Si le nœud  $N$  n'a pas l'état  $h(N_i, R)$  pour un voisin  $N_i$ , il calcule un coût estimatif  $c(N_i, R)$  comme une valeur par défaut pour  $h(N_i, R)$ . Le coût estimatif est défini par :  $C(N_i, R) = \alpha d(N_i, R) + (1-\alpha)e(N_i)$

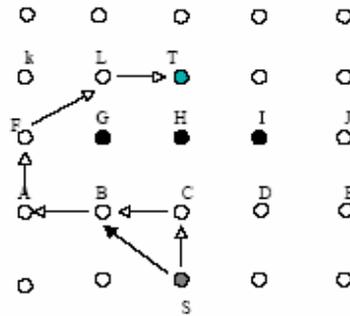
Où  $\alpha$  est un poids réglable,  $d(N_i, R)$  est la distance entre le nœud  $N_i$  et le centre de la région  $R$  normalisée par la plus grande distance parmi tous les voisins de  $N$  et  $e(N_i)$  est l'énergie consommée au nœud  $N_i$  normalisée par la plus grande valeur d'énergie consommée par les voisins de  $N$ .

Le nœud  $N$  va choisir un nœud ayant le plus petit coût estimatif, soit  $N_{min}$  et met son  $h(N, R)$  à  $h(N_{min}, R) + C(N, N_{min})$ . Ce dernier est le coût de transmission du paquet du nœud  $N$  vers le nœud  $N_{min}$ . Dans cette étape, deux cas peuvent se présenter :

- a) Un voisin proche de la région existe :** Quand le nœud  $N$  reçoit un paquet, il choisit parmi ses voisins proches de la destination celui du prochain saut tout en minimisant la valeur du coût  $h(N_i, R)$ . Ce processus est répété jusqu'à atteindre la région ciblée. En l'absence des trous, le coût d'apprentissage est une combinaison de l'énergie consommée et la distance.
- b) Tous les voisins sont lointains :** Dans ce cas, le nœud  $N$  sait qu'il est dans un trou. Le coût d'apprentissage et sa règle de mise à jour sont combinés pour éviter le trou. Considérons l'exemple suivant schématisé dans la figure (3.1):

Dans la figure suivante (figure 3.1), le centre de la région ciblée  $R$  est  $T$ . Les nœuds  $G, H, I$  sont presque épuisés en ressources d'énergie. Pour simplifier on prend  $\alpha=1$  et le coût de transmission d'un paquet d'un nœud à son voisin est égal à 1. Initialement au temps 0, les valeurs  $h(S, R)$  sont :  $h(B, T)=c(B, T)=\sqrt{5}$ ,  $h(C, T)=c(C, T)=2$ ,  $h(D, T)=c(D, T)=\sqrt{5}$ . A la réception d'un paquet destiné à  $T$ , le nœud  $S$  va choisir  $C$  qui se retrouve dans un trou. Au niveau de  $C$ , le paquet va être transmis à un nœud avec un coût minimum. Par exemple, il choisit  $B$  et met à jour son  $h(C, T)$  par  $h(C, T) = h(B, T) + C(C, B)$  où  $h(B, T) = \sqrt{5}$  et  $C(C, B)=1$ . Plus tard, les coûts d'apprentissage au niveau du nœud  $S$  vont être les suivants :  $h(B, T) = \sqrt{5}$ ,  $h(C, T) = \sqrt{5} + 1$ ,  $h(D, T) = \sqrt{5}$ . Le nœud  $N$ , à la réception d'un autre

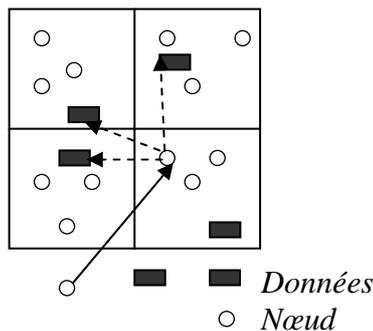
message destiné à T, va choisir directement le nœud B au lieu de C. Le nouveau coût d'apprentissage est propagé d'un saut en arrière quand le premier message atteint la région.



**Figure 3.1 :** Apprentissage des routes dans GEAR

## 2. La transmission du paquet à l'intérieur d'une région

Quand le paquet atteint la région, il est diffusé à l'aide de la diffusion géographique récursive. Dans ce cas, la région est divisée en quatre régions secondaires et quatre copies du paquet sont créées et envoyées dans les quatre sous régions. Ce processus est répété jusqu'à avoir des régions avec un seul nœud (la destination).



**Figure 3.2 :** Diffusion géographique récursive dans GEAR

Ce protocole a été proposé par les mêmes auteurs de la diffusion dirigée. Dans la simulation, GEAR a été comparé avec GPSR [79] et non pas avec la diffusion dirigée. GPSR est un des premiers travaux dans le routage géographique. Il a été conçu pour les réseaux Ad hoc mobiles et exige un service de localisation. GPSR emploie les graphes planaires pour résoudre le problème de trous et déterminer les chemins. GEAR réduit non seulement la consommation d'énergie, mais fonctionne mieux que GPSR en termes de livraison de paquets. Les résultats de simulation montrent que pour une distribution non-uniforme du trafic, GEAR livre de 70% à 80% de paquets plus que (GPSR) et pour une distribution uniforme il livre de 25% à 35% de paquets supplémentaires.

Dans un autre papier [36], GEAR a été comparé avec les deux algorithmes de diffusion déjà cités (Pull à une phase et Push). Cette comparaison a montré que GEAR est meilleur en terme de durée de vie du réseau et du nombre d'évènements livrés.

*Cette solution vise à améliorer les trois étapes de la diffusion dirigée. Elle propose la réduction du nombre d'inondation en utilisant l'information géographique. Aussi, des heuristiques énergétiques dans le renforcement et le routage des données, en se basant sur les valeurs des coûts, ont été utilisées.*

### 3.3.4 Le routage efficace en consommation d'énergie (Energy Aware Routing)

Shal et al. [80] ont proposé un protocole réactif qui utilise un ensemble de chemins semi-optimaux au lieu de maintenir un seul chemin. Ces chemins sont choisis en utilisant des fonctions de probabilité relatives à la quantité d'énergie utilisée sur chaque chemin. L'objectif principal de l'approche est de prolonger la durée de vie du réseau. L'idée est d'éviter l'utilisation du chemin à énergie minimale à chaque fois qu'un paquet est routé puisque ceci peut mener à un épuisement d'énergie des nœuds se trouvant sur un tel chemin. Le choix est porté donc sur d'autres chemins avec une certaine probabilité permettant d'augmenter la durée de vie du réseau. Le protocole utilise un adressage des nœuds basé sur la localisation et le type des nœuds. Il passe par trois phases :

#### *La phase d'installation*

- Dans cette étape, le nœud destination (puits) initie une inondation de l'intérêt. Il met aussi le champ 'coût' à zéro avant d'envoyer la requête.

$$\text{Cost}(N_D)=0$$

- Un nœud intermédiaire n'envoie la requête à un nœud que si celui-ci est plus proche de la source et plus loin de la destination (initiateur de la requête) que le nœud lui même.
- A la réception d'une requête, la métrique d'énergie est calculée et est ajoutée au coût total du chemin.

$$C_{N_j, N_i} = \text{Cost}(N_i) + \text{Metric}(N_j, N_i)$$

- Les chemins ayant des coûts élevés sont rejetés et non ajoutés à la table de routage. Seuls les nœuds  $N_i$  avec des chemins ayant une petite valeur du champ coût sont ajoutés.

$$FT_j = \{i / C_{N_j, N_i} \leq \alpha (\min_k C_{N_j, N_k})\}$$

- Un nœud  $N_j$  affecte une probabilité à chacun de ses voisins  $N_i$  dans la table de routage. Cette probabilité est inversement proportionnelle au coût :

$$P_{N_j, N_i} = \frac{1/C_{N_j, N_i}}{\sum_{k \in FT_j} 1/C_{N_j, N_k}}$$

- Ainsi, chaque nœud  $N_j$  a un nombre de voisins à travers lesquels il peut router les paquets vers la destination.  $N_j$  peut alors calculer le coût moyen pour atteindre la destination en utilisant les voisins se trouvant dans la table de routage :

$$\text{Cost}(N_j) = \sum_{i \in FT_j} P_{N_j, N_i} C_{N_j, N_i}$$

- Ce coût moyen,  $\text{Cost}(N_j)$  est mis dans le champ 'coût' du paquet de la requête et transmis vers la destination.

#### *La phase de communication de données*

- Chaque nœud source transmet le paquet de données en choisissant un nœud dans sa table de routage (ou table de transmission) selon les probabilités calculées.

- Chaque nœud intermédiaire transmet le paquet à un voisin choisi aléatoirement dans sa table de routage.
- Ce processus est répété jusqu'à atteindre le puits (destination).

### ***La phase de maintenance des routes***

Une diffusion classique (inondation) est rarement effectuée pour maintenir les chemins calculés.

Selon les résultats de simulation, cette technique a permis une amélioration de 21.5% dans l'énergie consommée, en plus d'une augmentation de 41% dans la durée de vie du réseau. Toutefois, il est à noter que le protocole requiert l'utilisation des informations de localisation en plus du mécanisme d'adressage des nœuds, ce qui complique davantage l'initialisation des routes par rapport à la diffusion dirigée.

*Cette solution n'est réellement pas proposée pour conserver l'énergie dans la diffusion dirigée. Les auteurs proposent leur solution comme un nouveau protocole de routage efficace en consommation d'énergie destiné pour les réseaux de capteurs. Cependant, ils assurent que beaucoup de similitudes peuvent être trouvées entre les deux solutions. Nous avons jugé que cette solution peut être utilisée dans le cas où le réseau de capteurs utilise déjà la diffusion dirigée dans le routage de données.*

*Les trois étapes du protocole correspondent à celles de la diffusion dirigée. Ainsi nous pouvons conclure que cette solution vise les étapes de renforcement et de routage de données en se basant sur les valeurs des coûts des différents chemins. La phase d'inondation est utilisée telle qu'elle est dans la découverte des routes.*

### **3.3.5 La diffusion dirigée avec énergie solaire**

Le protocole de routage proposé dans [94] est une extension de la diffusion dirigée standard qui emploie des nœuds capteurs alimentés par de l'énergie solaire. Comme dans la diffusion dirigée, la propagation de données est basée sur des interactions localisées, c-à-d des interactions avec les nœuds voisins. La terminologie employée dans ce protocole est la même que celle de la diffusion dirigée. Un gradient dénote toujours l'état de connexion entre les nœuds. Cependant, il contient également des informations sur l'état solaire des voisins qui indique si un nœud utilise une batterie ou est alimenté par de l'énergie solaire. Comme dans la diffusion dirigée, ce protocole suppose que les nœuds adjacents à un nœud peuvent être distingués par des marques. Les étapes de ce protocole sont :

#### ***La propagation des intérêts***

Les intérêts sont produits par un puits et sont simplement inondés dans tout le réseau. Ces intérêts contiennent les paires attribut/valeur, les identificateurs de la source et de la destination, un intervalle qui énonce le taux auquel le puits veut recevoir des données aussi bien qu'un compteur de saut et l'état solaire de l'expéditeur. Les nœuds établissent des gradients vers tous les nœuds à partir desquels les intérêts ont été reçus. Un gradient vers un voisin est mis à jour quand un message d'intérêt pour le même intérêt est reçu avec un plus petit nombre de saut. Pour chaque intérêt, les nœuds retransmettent le premier message d'intérêt qu'ils reçoivent et les intérêts suivants avec un nombre de saut inférieur.

#### ***La propagation des données***

Quand la source reçoit l'intérêt, elle capte l'environnement à un taux indiqué et envoie un message de données vers le puits. Un message de données contient des paires attribut/valeur qui permettent aux nœuds intermédiaires d'identifier les gradients et le prochain saut vers le puits. Le choix du prochain saut est fait en évitant les boucles. La

propagation des données sur le plus court chemin est une méthode pour les éviter. Cependant, s'il y a exactement un plus court chemin, celui contenant des nœuds alimentés par de l'énergie solaire peut ne pas être choisi. Le protocole permet à la source et à exactement un nœud intermédiaire de choisir un prochain saut différent de celui plus proche au puits. Un drapeau est utilisé dans le message de données pour indiquer si un nœud a déjà routé le message via un nœud intermédiaire ne se trouvant pas sur le chemin le plus court. Si ce n'est pas le cas, un nœud intermédiaire peut choisir d'envoyer les données à un autre nœud, qui soit alimenté par de l'énergie solaire plutôt qu'un nœud sur le plus court chemin. Une fois que le drapeau est placé les nœuds intermédiaires doivent envoyer leurs messages aux voisins se trouvant sur les chemins plus courts. Si un de ses voisins se trouve sur le plus court chemin et est alimenté par de l'énergie solaire, les nœuds intermédiaires envoient leurs données vers un tel nœud.

### ***Les mises à jours d'états***

Quand les nœuds changent leur état solaire, ils devraient informer les nœuds au sujet de leur nouveau statut. Ceci permet aux autres nœuds de les choisir pour router leurs données en cas où ils sont alimentés par de l'énergie solaire dans leur nouvel état. Ou bien choisir d'autres nœuds en cas où ils sont devenus dépendants de batteries. Les nœuds peuvent annoncer leur état en le diffusant simplement ou attendent jusqu'à écouter une communication ou eux même reçoivent des données. L'avantage de la première méthode est qu'un chemin efficace en consommation d'énergie peut être immédiatement trouvé alors qu'avec la deuxième quelques messages de données pourraient encore être routés à travers des chemins moins efficaces. L'avantage de la deuxième méthode est qu'elle évite les émissions inutiles.

*Cette solution est différente des solutions déjà présentées parce qu'elle suppose d'autres hypothèses par rapport à celles proposées dans la diffusion dirigée. Cette hypothèse concerne la nature des nœuds capteurs et leur alimentation en ressource d'énergie. Les solutions déjà citées y compris la diffusion dirigée supposent tous des nœuds non rechargeables à la différence de celle-ci. Ainsi, elle est applicable dans des applications particulières qui utilisent des nœuds capteurs particuliers.*

*Les étapes visées dans cette solution sont les étapes de renforcement et de routage. Elle propose plusieurs critères dans le choix des nœuds et des routes.*

### **3.3.6 La diffusion dirigée avec routage multiple**

L'utilisation des chemins multiples dans les protocoles de routage vise à réaliser deux principaux objectifs. Le premier est l'équilibrage des charges (la consommation de l'énergie) en distribuant le trafic entre la source et la destination à travers de multiple chemins partiellement ou complètement disjoints. Le deuxième est l'augmentation de la capacité de livraison fiable de données. Dans cette approche, plusieurs copies de données sont envoyées le long des différents chemins. Ceci produit une résistance aux pannes des chemins et assure donc la livraison des données [92].

L'objectif pour lequel ce type de routage est utilisé dans les deux solutions suivantes est différent. Il découle des caractéristiques de la diffusion dirigée qui montrent deux points importants dans la conception du protocole. D'abord, pour des raisons d'efficacité en énergie, les chemins sont construits d'une manière réactive et pas pro-activement. En second lieu, pour des raisons de robustesse, un schéma d'inondation périodique avec de petit taux d'envoi est adopté pour informer le puits et d'autres nœuds des défaillances possibles et prévenir leur résolution.

Les deux solutions proposées dans cette catégorie visent à réaliser une efficacité en consommation d'énergie en diminuant le nombre des inondations (périodique ou non) dans la diffusion dirigée. Ceci est réalisé en étendant la phase de routage par le calcul et la maintenance de plusieurs chemins.

### 3.3.6.1 Solution1

Le renforcement, dans la diffusion dirigée, est effectué dans le but d'établir empiriquement le 'meilleur' chemin (consistance, meilleur latence, ...). Les autres chemins renforcés sont maintenus par des inondations périodiques en utilisant un petit taux d'envoi. En cas de défaillances du meilleur chemin, une inondation de tout le réseau est effectuée. Cette solution considère un tel chemin comme 'chemin primaire' et assure le recouvrement en cas de pannes en évitant les inondations du réseau (périodiques ou non). Ceci est fait en construisant et maintenant des chemins alternatifs calculés au moment de l'établissement des gradients [92].

Les auteurs ont prouvé la possibilité de construction des chemins en utilisant des mécanismes de calcul distribués dans le paradigme de la diffusion dirigée. Ces mécanismes s'appuient principalement sur l'information locale collectée au niveau des nœuds et sur la technique de renforcement (positif et négatif) proposée dans la diffusion dirigée. Deux types de chemins sont considérés à savoir les chemins disjoints (figures 3.4) et les chemins non-disjoints (braided multipaths) (figure 3.5). Les chemins non-disjoints sont des chemins partiellement disjoints (nœuds communs) proposés pour réaliser une efficacité en énergie par rapport au premier type de chemins. Dans ce type, les chemins alternatifs peuvent être plus longs que le chemin primaire menant à une consommation d'énergie plus grande.

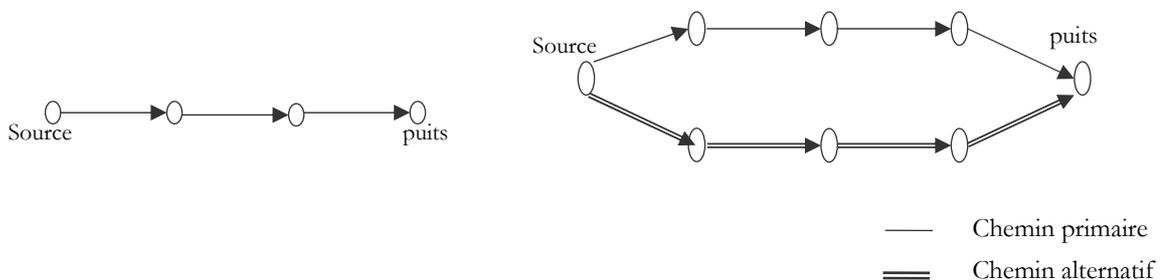


Figure 3.3 : Routage simple (à un seul chemin)

Figure 3.4 : Routage multiple « disjoint multipath »

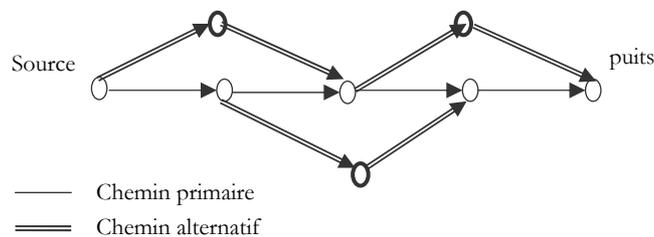


Figure 3.5: Routage multiple « braided multipath »

Pour construire les chemins disjoints, le puits renforce plusieurs chemins (y compris le primaire) vers ses voisins en direction de la source. En plus du chemin primaire, chaque nœud maintient un autre chemin en renforçant un de ses voisins (selon certains critères). Le renforcement négatif est aussi utilisé pour assurer la propriété de disjonction. Pour construire les chemins non disjoints, chaque nœud dans le réseau maintient une liste des chemins en renforçant un certain nombre de ses voisins.

Cette solution a été évaluée en considérant deux points : la résistance du schéma utilisé (disjoints ou non-disjoints) en cas de panne du chemin primaire et aussi le coût de maintenance (en énergie) requis. Ce dernier dépend du nombre de paquets utilisés dans la maintenance des chemins. Les deux types de chemins ont montré une résistance qui peut atteindre jusqu'à 50% de plus par rapport à la diffusion dirigée. Le deuxième type de chemins requiert un tiers du coût de maintenance par rapport au premier type. Aussi l'augmentation du nombre de chemins alternatifs augmente la résistance mais augmente aussi le coût de maintenance d'une manière proportionnelle.

*Cette solution laisse l'étape d'inondation d'intérêts telle que proposée dans la diffusion dirigée. Cependant, elle propose des améliorations dans l'étape de renforcement en proposant une construction des chemins plutôt qu'un renforcement par diffusion tout en essayant d'éviter l'inondation de tout le réseau. Aucun critère de sélection de route n'est utilisé, ainsi la solution vise à étendre la phase de routage sans lui appliquer des changements.*

### **3.3.6.2 Solution 2**

Cette solution vise à conserver plus de l'énergie par rapport à la première en proposant de réduire le coût de maintenance des chemins alternatifs tout en gardant le même niveau de résistance aux pannes [93]. Elle utilise les chemins non-disjoints (braided multipath) dans la phase de routage. L'idée est d'éliminer les paquets utilisés pour maintenir les chemins.

Chaque nœud dans la topologie peut avoir des renforcements vers plusieurs nœuds (braided multipath). Dans la phase de routage, le nœud envoie les paquets de données vers plusieurs voisins. Ainsi, si le nœud a renforcé deux chemins, il envoie le premier paquet de données vers le premier voisin et le deuxième paquet vers le deuxième voisin. Ceci mène à un maintien implicite des différents chemins sans l'utilisation de paquets particuliers. Le taux d'envoi des données est déterminant quant à la 'fraîcheur' des chemins maintenus. Cet algorithme est appelé l'algorithme BDR (Braided Dealer Routing). Selon le nombre des renforcements (chemins), les auteurs proposent deux autres algorithmes différents. Dans le premier (N-Braided Dealer Routing), chaque nœud renforce N chemins parmi ses voisins. Dans le deuxième (Log N-Braided Dealer Routing), chaque nœud renforce la moitié du nombre renforcé par son voisin précédent.

L'utilisation des deux premiers algorithmes a montré de meilleures performances en consommation d'énergie dans certains scénarios de la simulation. Cependant, leur résistance n'est pas toujours comparable à celle offerte par la première solution. L'algorithme Log-NBR a montré de meilleures performances en énergie et en résistance. La figure (figure 3.6) suivante montre l'alternance dans l'envoi des différents paquets de données telle que proposé dans cette solution.

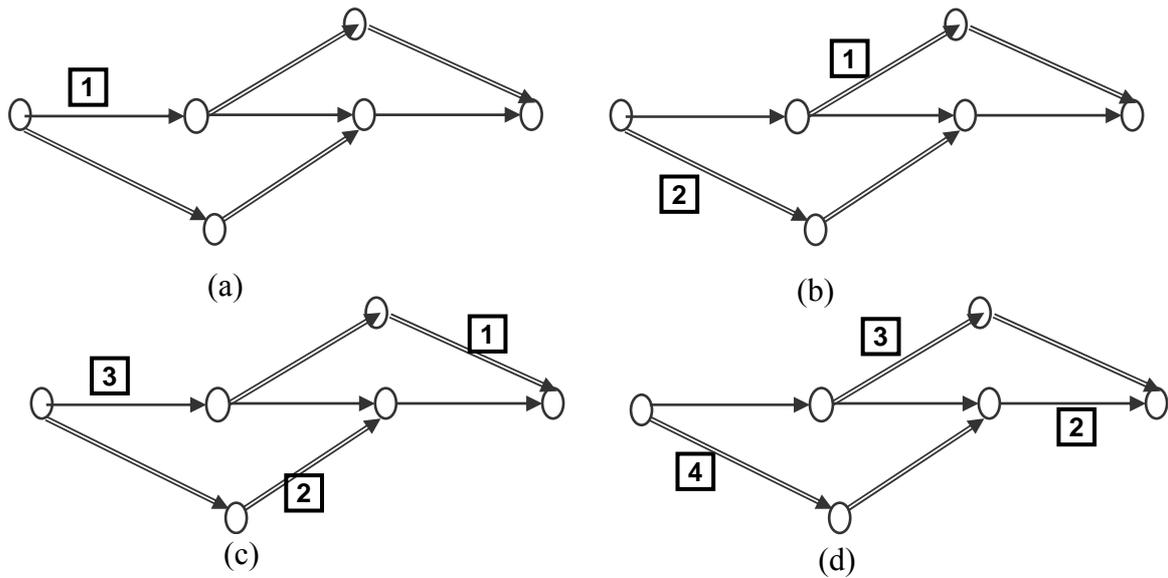


Figure 3.6 : Le dealer routing

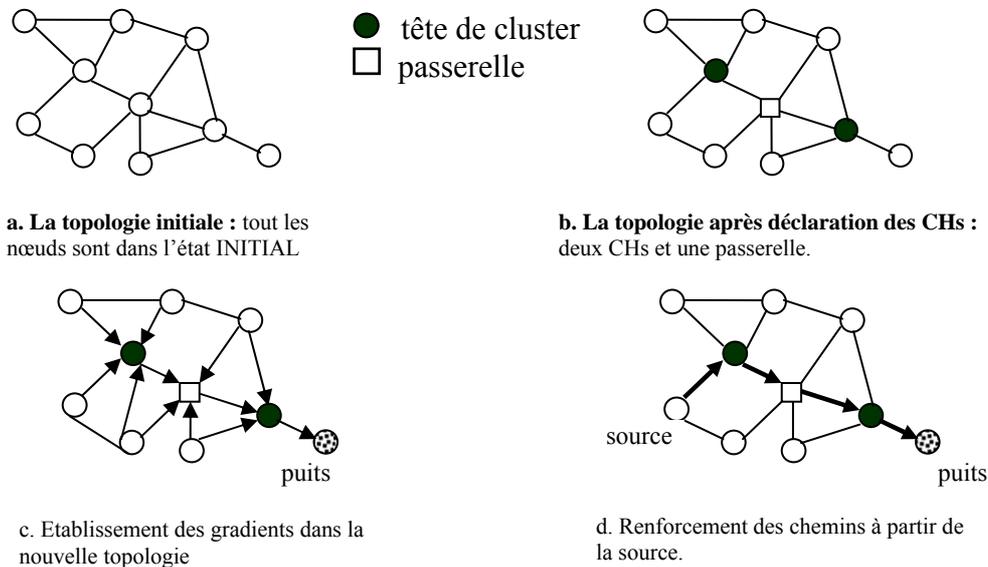
Cette solution vise à améliorer la consommation de l'énergie dans la solution 1 (paragraphe précédent) et donc dans la diffusion dirigée. Les améliorations portent sur les étapes de renforcement et de routage.

### 3.3.7 La diffusion dirigée avec le clustering passif

Le clustering peut être décrit comme le groupement des nœuds (d'un réseau) dans des entités logiques appelées clusters. Un représentant de chaque groupe (cluster) est élu pour le rôle de tête de cluster (clusterhead) en se basant sur différents critères. Le clustering passif est une technique de clustering proposée dans les réseaux Ad hoc dans le but de réaliser la diffusion efficace [81, 85]. La topologie résultante est constituée de quatre types de nœuds : les têtes de cluster, les passerelles (distribuées et complètes) et les nœuds ordinaires. Le clustering passif est aussi apparu pour pallier aux problèmes posés par les techniques de clustering traditionnelles tels que les coûts de formation et de maintenance. Dans cette solution, un nœud qui réclame en premier le rôle de tête de cluster se l'octroiera. Cette règle est appelée la règle du 'first declaration wins'. Une deuxième règle pour la déclaration des passerelles est appliquée. (L'algorithme du clustering passif sera détaillé dans le chapitre 'contribution')

La solution proposée dans [82] utilise le clustering passif dans le but de réaliser une efficacité en consommation d'énergie durant la phase d'inondation (messages d'intérêts et explorateurs de données). Au moment de l'inondation des premiers messages d'intérêt une nouvelle topologie est mise en place en appliquant l'algorithme du clustering passif. Les têtes de clusters et les passerelles résultantes sont les seules autorisées à diffuser les prochains messages d'intérêt et ceux explorateurs de données. Tout message diffusé de la part d'un nœud ordinaire est ignoré. La solution consiste à ajouter des informations supplémentaires dans les messages manipulés par la diffusion dirigée originale (comme proposé dans le clustering passif). Les informations ajoutées sont l'identité du nœud, son état dans le cluster, et deux adresses de têtes de cluster si le nœud est une passerelle. La combinaison proposée se base sur les hypothèses suivantes : (1) Les deux solutions utilisent des mécanismes réactifs (à la demande) qui prennent place au cas où il y a un trafic de données dans le réseau. (2) Seules les informations locales sont disponibles pour

leur fonctionnement. L'application du clustering passif avec la diffusion dirigée peut être schématisée dans la figure suivante :



**Figure 3.7 :** Application de la diffusion dirigée avec le clustering passif

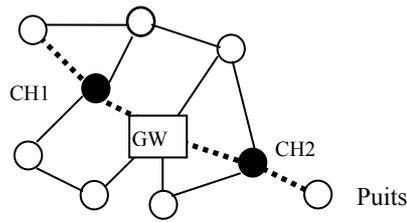
Les auteurs [82] ont montré au moyen de simulation que la diminution du niveau de connectivité introduit par la réduction de topologie n'affecte pas la distribution des données et les délais de réception. Aussi, la réduction du nombre de messages (topologie réduite) fournit un gain significatif dans l'énergie moyenne consommée dans le réseau.

*Cette solution vise à améliorer la quantité d'énergie moyenne consommée dans le réseau en minimisant le nombre des inondations. Ainsi, elle vise l'étape d'inondation et ne propose aucune modification dans les autres étapes.*

### 3.3.7.1 Discussion et critiques

L'application du clustering passif avec la diffusion dirigée a montré un gain significatif dans la quantité d'énergie moyenne consommée dans le réseau. Cependant, la solution proposée présente quelques limitations. Reprenons le schéma décrivant l'application du clustering passif dans un réseau de capteurs. D'après la règle du « First Declaration Wins », un nœud qui réclame le rôle de tête de cluster se l'octroiera selon sa capacité d'annoncer en premier son état dans le prochain message sortant.

Supposons que les nœuds CH1, CH2, GW sont les premiers à annoncer leurs états quand ils reçoivent des messages entrants (plus rapides). Ces nœuds vont alors jouer les rôles de têtes de clusters ou bien de passerelles. Par conséquent, ils seront responsables d'effectuer les inondations des messages d'intérêts et ceux explorateurs de données dans le réseau. Supposons encore que les mêmes nœuds vont garder leurs rôles pour une certaine durée de temps. Ceci va engendrer une concentration du trafic au niveau de ces nœuds et donc une variance dans leurs états de batteries par rapport aux autres nœuds du réseau (ordinaires). Après un certain nombre d'opérations, ces nœuds vont épuiser leur énergie. Plusieurs chemins vont alors devenir non valides, ce qui peut conduire au partitionnement du réseau.



**Figure 3.8 :** Résultat de l'application du passive clustering avec la diffusion dirigée

Pour cette raison et dans le but de minimiser la variance des états des batteries des nœuds, nous proposons de considérer le facteur énergie dans la formation et la maintenance des clusters. Notre principal objectif est de maximiser la durée de vie de chaque nœud et donc la durée de vie globale de tout le réseau. Les détails de notre contribution seront présentés dans le chapitre 4.

### **3.4 Conclusion**

Dans ce chapitre nous avons présenté les extensions de la diffusion dirigée visant à le rendre plus efficace en consommation d'énergie. Ces solutions emploient plusieurs techniques comme la définition des métriques sur les chemins utilisés, l'utilisation de l'information de localisation et l'emploi des protocoles de gestion de topologie.

Nous avons donné quelques critiques par rapport à chaque solution, particulièrement la dernière solution qui propose d'employer une technique de clustering afin de déduire une topologie plus réduite du réseau. Les solutions que nous proposons pour pallier aux limitations de cette solution feront l'objet de notre contribution qui sera détaillée dans le chapitre suivant.

## **Chapitre 4**

# **Présentation d'une amélioration d'un algorithme de clustering pour la conservation d'énergie dans la diffusion dirigée**

### **4.1 Introduction**

La diffusion dirigée représente l'une des propositions les plus prometteuses dans les réseaux de capteurs, grâce au nouveau paradigme de routage orienté données utilisé. Son efficacité en consommation d'énergie constitue l'étape de conception la plus importante.

Dans cette optique, nous décrivons dans ce chapitre l'amélioration proposée pour ce paradigme, en vue de le rendre plus efficace en consommation d'énergie. Notre contribution est à double volet : (1) Prendre en considération les états énergétiques (taux de décharge) dans la formation et la maintenance des groupes de nœuds dans une technique de clustering existante [81]. Ceci est fait dans le but d'équilibrer la consommation d'énergie dans le réseau en alternant les rôles des nœuds responsables des opérations de diffusion. (2) Appliquer la technique résultante avec la diffusion dirigée. Pour cela nous proposons deux points d'interactions : seulement les têtes de clusters et les passerelles effectueront les diffusions non optimistes et les têtes de clusters joueront le rôle des entités combinateurs de données.

Notre contribution a pour but de garder le réseau fonctionnel le plus longtemps possible en d'autres termes prolonger la durée de vie du réseau. De la sorte, prolonger le processus de captage que les nœuds du réseau mettent en œuvre.

Le reste de ce chapitre est organisé comme suit. Nous décrivons la technique de clustering passif utilisée. Ensuite, nous présentons nos propositions et ainsi la technique de clustering efficace en consommation d'énergie. Nous finirons le chapitre par la présentation de la combinaison entre la diffusion dirigée et notre algorithme de clustering.

### **4.2 Pourquoi le clustering ?**

Les deux principales classes de protocoles utilisés, à l'heure actuelle dans les réseaux sans fil, s'appuient sur la technique de routage multi-sauts ou de clustering (voire une hybridation des deux). La première se base sur le calcul de différents chemins menant de la source vers la destination. Ceci est fait selon différents critères tels que : le nombre de nœuds, la proximité à la destination, les nœuds ayant des quantités d'énergie maximales, la présence de nœuds particuliers dans le chemin, ...etc. Cependant, le premier inconvénient du routage multi-sauts dans les réseaux sans fil est la périodicité des messages envoyés pour maintenir les itinéraires valides, ce qui a pour effet d'encombrer le réseau et de consommer ainsi de l'énergie supplémentaire. Dans ce type de réseaux, le clustering donne en revanche de meilleurs résultats [84]. Dans les réseaux de capteurs, il est employé dans le but d'offrir plusieurs avantages tels que la scalabilité, la robustesse, et l'efficacité en consommation d'énergie [83].

L'application de la diffusion dirigée avec le clustering passif (présentée dans le chapitre précédent) a fourni, selon les résultats de simulation, un gain significatif dans la quantité d'énergie globale consommée. Cette solution a aussi amélioré les délais de transmission et le taux de livraison de données.

Tous ces facteurs ont motivé notre choix de porter notre contribution sur une technique de clustering en essayant de prendre de nouveaux aspects dans la formation et la maintenance des clusters. Nous jugeons que l'application des aspects proposés avec la diffusion dirigée nous permettra d'améliorer davantage l'efficacité en consommation d'énergie de ce paradigme.

### **4.3 Le clustering dans les réseaux de capteurs**

Les approches hiérarchiques ou basées-clusters, originellement proposées dans les réseaux filaires, sont des techniques connues qui ont des avantages spéciaux dans les réseaux de capteurs tels que la scalabilité et la communication efficace. Cette dernière peut se traduire par l'efficacité en consommation d'énergie, la fiabilité, l'absence de surcharge, la minimisation des délais de transmission, etc.

Ces approches guidées par les techniques de clustering (groupement ou classification) ont beaucoup été étudiées dans les réseaux sans fil et Ad hoc. Les études ont été faites sur ces approches en tant que protocoles autonomes pour les réseaux Ad hoc [84], ou dans le contexte des protocoles de routage [85].

Les algorithmes de clustering dans les réseaux de capteurs [23, 29, 30] diffèrent par les critères utilisés dans la formation des groupes ainsi que les objectifs visés par leur utilisation. Plusieurs techniques du calcul distribué peuvent être utilisées pour créer dynamiquement des clusters comme la règle de la plus petite identité (LID), le degré d'un nœud dans le réseau, les états énergétiques de nœuds, la proximité des nœuds à un but particulier, etc. Dans les réseaux de capteurs, ces critères peuvent être étroitement liés à la sémantique des applications.

Le clustering vise à mettre en place une topologie plus réduite du réseau comparée à celle initiale. Ceci réduit le coût des communications des messages en limitant leur retransmission. Cependant, cette approche souffre encore du coût de formation et de maintenance des différents clusters. L'efficacité d'une méthode de clustering se traduit par la différence entre le coût de formation/maintenance et le gain en ressources réalisé en réduisant le nombre de transmissions. Cette efficacité est difficile à atteindre en prenant en considération la nature proactive et intensive en ressources des algorithmes de clustering proposés.

#### **4.3.1 Le clustering passif**

Pour pallier au problème de coûts de formation et de maintenance, Kwon et Gerla ont proposé le Clustering passif [81]. Leur solution consiste en un algorithme de formation et de maintenance de clusters à la demande. Le fonctionnement de cet algorithme ne nécessite pas la manipulation de paquets explicites de contrôle (information de voisinage). Cet algorithme relie la fréquence d'envoi des messages de données avec la fraîcheur des entités clusters. La formation de ces dernières est initiée par le premier message de données à diffuser dans le réseau. La solution évite alors la longue période d'initialisation et réduit l'ensemble des nœuds impliqués dans les communications dans une courte durée de temps.

Contrairement aux algorithmes conventionnels, ce protocole peut fonctionner en parallèle avec les autres protocoles de la couche réseau du fait qu'il ne requiert pas une

phase d'initialisation. La formation des clusters se fait d'une manière implicite en exploitant l'information de voisinage contenue dans les messages de données. Elle se fait selon deux principales règles qui tentent d'éviter les problèmes engendrés par les algorithmes traditionnels tels que : la reconstruction des clusters en cas de changement dans le réseau et l'isolation des clusters quand le critère de formation (par exemple : plus petite identité) n'est pas applicable sur tous les nœuds du réseau [85].

#### **4.3.1.1 Les éléments du clustering passif**

##### **1. Les états manipulés**

Un nœud dans le réseau peut se trouver dans plusieurs états. Ces derniers sont de trois types : initial, interne ou externe.

##### **L'état initial :**

A l'initialisation du réseau tous les nœuds se trouvent dans l'état *Initial*. Cet état précède la réception ou l'émission du premier message de données. Un nœud qui n'appartient à aucun cluster est aussi dans l'état initial.

##### **Les états internes :**

Un état interne est un état non diffusé dans le réseau et est manipulé au niveau des nœuds. C'est un état de transition vers un état externe qui sera annoncé dans tout le réseau. Les états internes d'un nœud sont :

- **Prêt à être tête de cluster (CH-Ready)**: un nœud initial passe à cet état dès qu'il reçoit un message de la part d'un nœud qui n'est pas tête de cluster.
- **Prêt à être une passerelle (GW-Ready)** : un nœud membre d'un cluster ou se trouvant dans l'état initial qui connaît l'existence d'un ou de plusieurs têtes de clusters et pour qui aucune passerelle n'a été annoncée passe à cet état.

##### **Les états externes :**

Un état externe est un état final temporaire déclaré dans tout le réseau. Les états externes sont déclarés dans les messages sortants destinés au réseau. Les états externes d'un nœud sont :

- **Tête de cluster (CH)** : si un nœud dans l'état CH-Ready envoie un message vers le réseau, il se déclare comme CH. Cette déclaration est faite en ajoutant cette information dans l'entête du message (piggyback).
- **Passerelle (Full-GT)** : un nœud qui peut écouter de deux nœuds têtes de clusters jouera le rôle de passerelle entre elles. Cet état est précédé de l'état GW-Ready.
- **Passerelle distribuée (Distr-GW)** : quand un nœud membre d'un cluster peut écouter les messages d'un nœud membre d'un autre cluster, celui-ci jouera le rôle de passerelle distribuée. Cet état est aussi précédé de l'état GW-Ready.
- **Nœud ordinaire (Ordinary-Node)** : un nœud ordinaire est un nœud membre d'un cluster qui peut écouter suffisamment de passerelles dans son voisinage.

##### **2. La règle du 'first declaration wins'**

Elle consiste à donner le rôle de tête de cluster (CH) au premier nœud qui le réclame régnant ainsi sur l'ensemble des nœuds se trouvant dans sa zone de couverture. Ainsi, un nœud qui n'a pas encore reçu un message de la part d'un nœud CH se déclarera pour le rôle de CH dans son prochain message sortant (destiné au réseau). Aussi, si aucun nœud ne se déclare pour le rôle de passerelle pour une certaine durée de temps (timeout) le nœud

CH abandonne son rôle. En conséquence, une nouvelle construction est lancée dans laquelle participent tous les nœuds du réseau.

### **3. La règle de sélection de passerelle**

Une passerelle est un nœud qui appartient à deux clusters ou plus. L'idéal est d'avoir une passerelle pour chaque paire de CHs. Cette règle permet d'avoir exactement une passerelle pour chaque paire de CHs [81]. Cependant, il est possible d'avoir le cas où aucune passerelle n'existe pour deux clusters adjacents. Ce cas se présente quand les deux CHs sont voisins de trois sauts (au lieu de deux) dans ce cas les deux nœuds intermédiaires sont déclarés comme Distr-GW. Les règles peuvent se résumer en :

- **Passerelle** : un nœud qui appartient à deux clusters en même temps peut devenir un candidat de passerelle. Il devient passerelle pour deux clusters si aucune passerelle ne s'est déclarée pour ces deux clusters. En cas de conflit, des informations supplémentaires (identité des nœuds) sont utilisées pour élire la passerelle. Si le nœud appartient à seulement deux clusters, il peut leur servir immédiatement de passerelle.
- **Passerelle distribuée** : une passerelle distribuée assure la connectivité de trois sauts entre les clusters. Un nœud qui a reçu un message de la part d'un nœud CH peut passer à l'état ordinaire si au moins deux passerelles distribuées sont connues au niveau de ce nœud. Autrement, il garde l'état GW-Ready. Un nœud candidat de passerelle est considéré comme passerelle distribuée si aucun nœud appartenant au même cluster ne s'est déclaré pour ce rôle. Si un nœud ordinaire reçoit un message de la part d'une passerelle distribuée et qu'il n'a pas de passerelle dans son voisinage, il passe directement à l'état Distr-GW.

#### **4.3.1.2 Le principe de fonctionnement**

Le clustering passif ajoute (piggyback) aux paquets manipulés des champs supplémentaires (dans l'entête). Il inspecte les paquets reçus et enlève les champs ajoutés si le nœud est la destination finale du paquet. Ceci permet au clustering passif d'être appliqué avec les protocoles de la couche MAC et réseau sans qu'ils soient modifiés. Les deux règles citées auparavant sont utilisées dans la formation et la maintenance des clusters. Les informations ajoutées au paquet de données sont : l'identité du nœud, son état dans le cluster, et l'adresse de deux têtes de cluster si le nœud est une passerelle.

A l'initialisation tous les nœuds du réseau se trouvent dans l'état *Initial*. Un nœud peut transiter à l'état *CH-Ready* s'il reçoit un message de la part d'un autre nœud qui n'est pas un *CH*. Avec le prochain message sortant, ce nœud se déclare comme *CH* en mettant le champ état dans l'entête du message sortant à *CH*. Un nœud membre (qui appartient à un cluster) va servir de passerelle ou être un nœud ordinaire en se basant sur l'information collectée à partir des voisins. Il devient un nœud ordinaire si un certain nombre de nœuds *GW* ont déjà été déclarés. Autrement, il se déclare comme *GW-Ready* s'il connaît deux ou plusieurs têtes de clusters. De cet état, le nœud peut passer à l'état Full-GW ou Distr-GW avec le prochain message sortant. Si aucune passerelle ne s'est déclarée, il choisit deux têtes de clusters pour leur servir de passerelle (Full-GW). Sinon il passe à l'état Distr-GW dans le cas où il reçoit un message de la part d'un nœud ordinaire d'un autre CH.

Dans le clustering passif, chaque nœud manipule des listes qui contiennent les identités de ses voisins (nœuds à partir desquels il a reçu des messages). Chaque liste contient les nœuds se trouvant dans un état donné. Les listes sont gérées par un temps d'expiration (timeout) suggéré à 2 secondes [86]. Dans le cas où un nœud CH abandonne

son rôle, il utilise un message particulier dit 'message de give-up'. Les têtes de clusters sont les seules à abandonner leurs rôles d'une manière explicite.

*Le clustering passif a été proposé dans le contexte des réseaux Ad hoc comme nouveau mécanisme pour réaliser la diffusion efficace. Ce mécanisme a été appliqué avec plusieurs protocoles de routage comme : AODV, DSR, ODMRP [85] dans le but d'améliorer leurs performances en matière de consommation de ressources. Dans les réseaux de capteurs, plusieurs solutions utilisent ou étendent le clustering passif. Le but principal de ces solutions est de réaliser un gain en énergie. En plus de [82], présentée dans le chapitre précédent, [87] propose son utilisation avec un algorithme asynchrone de synchronisation d'horloge. Ce dernier est utilisé sur une topologie plus réduite dans le but de minimiser le nombre des opérations requises dans la convergence des horloges. Dans [88], une extension du clustering passif a été utilisée au niveau de la couche MAC.*

*Dans notre travail, nous proposons d'étendre le clustering passif pour prendre en considération d'autres facteurs dans la formation et la maintenance des clusters. Les détails de notre contribution feront l'objet des paragraphes suivants.*

#### **4.4 Le clustering passif efficace en consommation d'énergie: Energy aware passive clustering (EAPC)**

Les travaux proposés dans le domaine de la conservation d'énergie dans les protocoles de routage se focalisent généralement sur deux aspects : minimiser le coût par message (calcul des chemins optimaux en utilisant des métriques spécifiques) et aussi équilibrer la consommation d'énergie dans le réseau (exemple : au moyen d'exploration de plusieurs chemins) [53]. Le premier aspect vise à minimiser l'énergie globale consommée dans tout le réseau alors que le deuxième a pour objectif de garder le réseau fonctionnel le plus longtemps possible en évitant la surutilisation d'un ensemble de nœuds qui peut mener à un partitionnement rapide du réseau.

Dans le but de prendre en compte les deux aspects cités auparavant, la solution proposée vise à étendre celle proposée dans [81]. Elle consiste à effectuer un clustering passif en considérant les états des batteries des nœuds dans la formation et la maintenance de la topologie. Ceci a pour objectif de minimiser l'énergie globale consommée (avec l'application du clustering passif) et surtout de réaliser un équilibre dans la consommation de l'énergie des différents nœuds (en prenant en considération l'état des batteries des nœuds du réseau). Les principaux apports peuvent être résumés en :

- Alternier les rôles de nœuds responsables des opérations d'inondation (têtes de clusters et passerelles) entre les nœuds du réseau en fonction du taux d'énergie consommée. Aussi, tous les conflits qui peuvent avoir lieu entre les nœuds sont résolus en considérant leur taux d'énergie consommée. Ainsi, un nœud ayant un taux inférieur aura la priorité de garder ou de s'octroyer le rôle de têtes de cluster ou de passerelle. Dans le cas d'égalité l'identité des nœuds est utilisée pour l'affectation des rôles.
- Un nœud ne se déclare pas pour le rôle de tête de cluster ou de passerelle si son taux d'énergie consommé est supérieur à une valeur donnée. En cas où aucun de ses voisins ne se déclare comme tête de cluster dans un temps limité (timeout), il prend la tête de son groupe et ajuste son seuil (se déclare comme passerelle).
- Un nœud jouant le rôle de tête de cluster ou de passerelle l'abandonnera quand son niveau d'énergie tombe en dessous d'un seuil. Ce nœud change son état en fonction des informations locales se trouvant à son niveau et de la durée du timeout.

- Inclure l'information relative à l'énergie dans les messages manipulés en plus de l'identité, l'état du nœud et les adresses des têtes de clusters.

#### 4.4.1 Les éléments du protocole :

##### 4.4.1.1 Les états des nœuds

Les états des nœuds manipulés dans notre protocole sont les mêmes que dans la proposition originale du clustering passif [81]. De la sorte, nous manipulons cinq états visibles dans le réseau qui sont : *Initial*, *CH*, *Distr-GW*, *Full-GW* et *Ordinary*, et deux états internes manipulés au niveau des nœuds à savoir : *CH-Ready* et *GW-Ready*.

##### 4.4.1.2 L'entête des messages manipulés

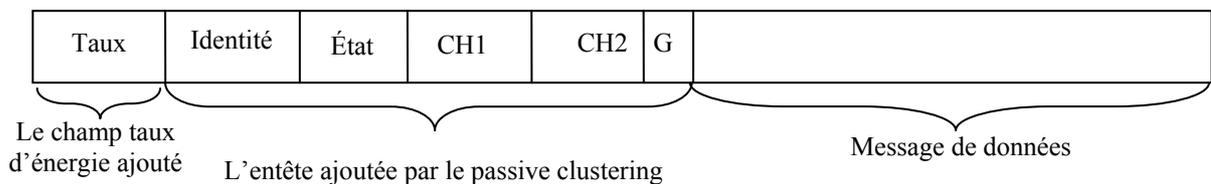
Dans le but de considérer le facteur 'énergie' dans la formation et la maintenance des clusters, nous ajoutons un champ taux (d'énergie) dans l'entête du message manipulé. Ainsi, l'entête du message va être composée de cinq champs supplémentaires (figure 4.1):

- **Le champ taux :** ce champ contient le taux d'énergie consommée au niveau du nœud.

$$\text{Taux} = \frac{E_c}{E_i}$$

Où  $E_c$  est la quantité d'énergie consommée et  $E_i$  est la quantité d'énergie initiale. Exemple : un nœud ayant consommé la moitié de son énergie, aura un taux d'énergie consommée égal à 0.5 (soit de 50%).

- **Le champ identité :** similairement à [81], ce champ contient l'identité du nœud originaire du message. Dans notre solution ce champ est utilisé en cas de conflit. Il sert à décider de l'attribution des rôles quand les taux d'énergie consommée des nœuds en conflit sont identiques.
- **Les champs CH1 et CH2 :** ces champs contiennent l'identité de deux têtes de clusters. Ces champs sont présents dans le cas où le nœud source est une passerelle. Ils vont contenir les identités de deux têtes de clusters pour lesquels le nœud source sert de passerelle.
- **Le champ G :** comme dans [82], ce champ est utilisé dans le cas où le CH abandonne son rôle. Ce champ est utilisé pour remplacer le 'message de give-up'. Dans notre cas nous utiliserons ce champ dans le cas où le nœud CH relâche son rôle quand son taux d'énergie consommée dépasse une valeur définie.



**Figure 4.1 :** Structure de l'entête d'un message manipulé par EAPC

#### 4.4.2 Les règles de fonctionnement

Dans notre solution nous proposons de modifier les règles de formation et de maintenance de topologie présentées dans [81]. Les deux règles utilisées dans la sélection

des têtes de clusters et de passerelles vont être modifiées de façon à prendre en compte les apports présentés précédemment :

#### ***La règle de sélection de CH***

Nous ajoutons une contrainte sur l'état énergétique des nœuds dans la règle de sélection des têtes de cluster. La règle du « first declaration wins » n'est appliquée que si le nœud a un taux d'énergie consommée inférieur à une valeur  $\lambda$ . Dans le cas contraire, le nœud passe de l'état CH-Ready à l'état Initial. Le nœud gardera cet état tant qu'il n'a pas reçu un message de la part d'un autre CH (dans ce cas il passera à l'état Ordinary) ou que la durée du timeout ne s'est écoulée. Un nœud jouant le rôle de tête de cluster peut garder son rôle tant que son taux est inférieur  $\lambda$ . Si son niveau d'énergie tombe en dessous d'un seuil, il relâche son rôle et passe à l'état Initial ou ordinaire selon qu'il connaît ou non d'autres têtes de cluster dans son voisinage (la liste des CHs n'est pas vide). Si la durée du timeout est écoulée et qu'un nœud dans l'état Initial ne reçoit aucun message de la part d'un CH. Il se déclare comme tête de cluster et ajuste son seuil.

#### ***La règle de sélection de passerelle***

Identiquement au cas de tête de cluster, un nœud ne passe de l'état GW-Ready à l'état Full-GW ou Distr-GW que si son taux d'énergie consommée est inférieur à un taux  $\lambda$ . Autrement, il se déclare comme nœud ordinaire. De même, le nœud gardera cet état mais finira par se déclarer pour le rôle de passerelle après une durée de temps (timeout), si aucun de ses voisins ne se déclare pour ce rôle. Aussi, un nœud qui sert de passerelle dans le réseau, abandonne son rôle quand son niveau d'énergie tombe en dessous d'un seuil et suit le même principe cité.

Dans le cas de tête de cluster, si le nœud ne satisfait pas la condition du taux d'énergie il peut basculer entre les deux états ordinaire ou Initial. Alors que dans le cas de passerelle, il passe obligatoirement à l'état ordinaire. Ceci est dû au fait qu'un nœud jouant le rôle de passerelle ou se trouvant dans l'état GW-Ready connaît certainement des têtes de clusters dans son voisinage.

### **4.4.3 Les étapes de l'algorithme**

Un nœud du réseau manipule les variables suivantes :

- CH-list : liste de têtes de clusters à partir desquelles le nœud a reçu un message.
- GW-list : liste des passerelles connues au niveau d'un nœud.
- ORD-list : liste des nœuds ordinaires connus au niveau d'un nœud.
- INIT-list : liste des nœuds initial connus au niveau d'un nœud.
- Mon-taux : taux d'énergie consommée.
- Give-up : variable manipulé pour indiquer si le nœud veut ou non abandonner son rôle.

L'algorithme de formation et de maintenance de cluster passe par trois principales étapes :

#### ***4.4.3.1 L'étape d'initialisation***

Au tout début, tous les nœuds du réseau sont dans l'état Initial, le nœud restera dans cet état jusqu'à ce qu'il envoie/reçoive un message vers/du réseau.

```
mon-état ← Initial
{
Attendre la réception ou l'émission d'un message
}
```

#### **4.4.3.2 L'étape de formation et de maintenance :**

Cette étape fonctionne en parallèle avec les autres protocoles de la couche protocolaire. Les nœuds du réseau forment et maintiennent les clusters en changeant leurs états internes et externes. A la réception d'un message, le nœud réévalue son état. Il le garde ou non selon les informations contenues dans le message reçu. Lors de l'émission du prochain message, le nœud annonce un état externe qui sera visible dans le réseau. A la réception et à l'émission d'un message le nœud exécute le pseudo-code suivant :

A la réception d'un message:

```
A la réception d'un message:
{
give-up ← faux
Si message.G == vrai alors
    {supprimer le nœud des listes et réévaluer l'état ;
    return ;}

// Le nœud expéditeur a abandonné son rôle, par
// conséquent une réévaluation des états doit être
// établie. Par exemple : si le nœud est membre
// d'un cluster que son CH a abandonné son rôle, le
// nœud passe à l'état Initial.

case message.état == CH
{
    Si mon-état == CH alors
        Si mon-taux > message.taux alors {give-up ← vrai;
        ajouter le CH à la CH-list ;
        }
        Sinon si mon-taux == message.taux alors
            {Comparer les identités des nœuds, si le nœud a une
            plus petite identité, il gardera son état}
            sinon { ajouter le CH à la CH-list ;
            return ;}
        fsi
    fsi
    sinon ajouter le CH à la CH-list et réévaluer l'état;
    fsi
}
```

```
case message.état == Full-GW ou Distr-GW
```

```
{ ajouter les passerelles dans la GW-list et réévaluer l'état.
```

```
    En cas de conflit (nœuds ayant le même état), le même principe cité dans le cas de CH, est appliqué. Ici le conflit peut avoir lieu entre deux passerelles si l'état est le même est que les deux CHs servis par les passerelles sont les mêmes. Une comparaison entre les taux d'énergie consommée est alors appliquée pour résoudre le conflit
```

```
}
```

```
case message.état == initial ou ordinaire
```

```
{ ajouter les nœuds dans les listes ORD-list et INIT-list et réévaluer l'état.
```

```
}
```

```
Si (mon-état ==initial ) & (message.état = CH) alors
```

```
    mon-état ← CH-Ready
```

```
fsi
```

```
}
```

A l'émission d'un message :

```
A l'émission d'un message:
```

```
{
```

```
    Si give-up == vrai alors {give-up← faux
```

```
        si la CH-list est non vide alors mon-état← Ordinary ;
```

```
            sinon mon-état← Initial ;
```

```
        fsi }
```

```
fsi
```

```
Case mon-état == CH-Ready
```

```
{
```

```
    Si mon-taux  $< \lambda$  alors mon-état←CH
```

```
        Sinon si (la durée du timeout est écoulée) alors {mon-état←CH ;
```

```
            réévaluer  $\lambda$  ;}
```

```
        sinon mon-état← Initial ;
```

```
    fsi
```

```
fsi
```

```
}
```

```
case mon-état == GW-Ready
{
Si mon-taux <  $\lambda$  alors mon-état ← Full-GW (Distr-GW : selon la règle de
sélection de passerelle)
    Sinon si (la durée du timeout est écoulée)
        alors {mon-état←Full-GW (Distr-GW) ;
                réévaluer  $\lambda$  ;}
        sinon mon-état← ordinaire ;
fsi
}
case mon-état == CH (Full-GW ou Distr-GW)
{
Si mon-taux >  $\lambda$  alors give-up ← vrai ;
    si CH-list n'est pas vide alors mon-état← ordinaire ;
    sinon mon-état← Initial ;
    fsi // dans le cas de passerelle mon-état← ordinaire
fsi
}
Ajouter les informations (taux, identité, les deux CHs en cas de passerelle, G)
dans le message ;
Envoyer le message ;
}
```

#### **4.4.4 La définition et la réévaluation des paramètres**

La difficulté de notre solution réside dans la définition et la réévaluation des valeurs de paramètres utilisés dans le protocole. Nous proposons d'utiliser la même valeur du timeout suggérée dans le draft IETF [86] de la solution du clustering passif à savoir : 2 secondes. La valeur du taux  $\lambda$  est comprise entre les deux valeurs 0 et 1. Trouver une valeur significative de  $\lambda$  n'est pas une tâche triviale, ainsi nous proposons une valeur de 0.5 et montrerons d'une manière empirique (au moyen de simulation) que c'est une valeur raisonnable pour le cas des réseaux de capteurs. L'ajustement de la valeur de  $\lambda$  est fait en ajoutant un taux de 5 % à chaque fois que la condition n'est pas satisfaite.

#### **4.5 L'application du EAPC avec la diffusion dirigée**

Les objectifs des travaux proposés dans le domaine de la conservation d'énergie dans les réseaux de capteurs peuvent se résumer en 'la prolongation de la durée de vie du réseau et donc du processus de captage que le réseau de capteurs est censé mettre en œuvre'. Ces objectifs sont atteints par la minimisation de la quantité d'énergie consommée dans le réseau et aussi la répartition de la consommation afin de garder les nœuds capteurs fonctionnels le plus longtemps possible [73].

Dans le but d'atteindre le deuxième objectif cité, nous proposons d'appliquer le protocole de clustering EAPC présenté dans la section précédente avec le paradigme de la

diffusion dirigée. Ceci est fait dans le but d'équilibrer la consommation d'énergie au niveau des nœuds. Ainsi, éviter le partitionnement rapide et donc prolonger la durée de vie du réseau.

#### **4.5.1 Les hypothèses de travail**

Considérons un ensemble de capteurs dispersés dans une zone géographique. Nous supposons les propriétés suivantes au sujet du réseau de capteurs:

- Les nœuds capteurs sont quasi-stationnaires. C'est typique pour les applications des réseaux de capteurs. Ainsi nous ne considérons pas la contrainte de mobilité sur les nœuds du réseau.
- Les liens sont symétriques, c.-à-d., deux nœuds N1 et N2 peuvent communiquer en utilisant la même puissance de transmission.
- La consommation d'énergie est non uniforme au niveau des nœuds.
- Aucun système de positionnement n'est supposé au niveau des nœuds.
- Tous les nœuds ont des capacités similaires, ceci motive le besoin d'étendre la durée de vie de chaque nœud.
- Les nœuds sont laissés sans surveillance après déploiement. Par conséquent, la recharge des batteries n'est pas possible.
- Le paradigme de la diffusion dirigée est utilisé dans la dissémination des données entre les nœuds du réseau. Nous considérons l'algorithme de diffusion de base de ce paradigme à savoir le 'Pull à deux phases'.
- Pour tenir compte des états énergétiques des batteries, nous supposons que chaque nœud est capable de déterminer à tout instant sa quantité d'énergie résiduelle et donc son taux d'énergie consommée. Pour ceci, il suffit juste d'implémenter un mécanisme de bas niveau qui permet de fournir le voltage de la batterie, et la courbe de décharge [73].

#### **4.5.2 Le principe**

Pendant les inondations des premiers messages dans le réseau, une nouvelle topologie est mise en place en exécutant le protocole EAPC en parallèle. Les nœuds vont alors s'organiser en plusieurs catégories (têtes de clusters, passerelles, nœud ordinaires). L'interaction entre les deux protocoles est résumée en deux points :

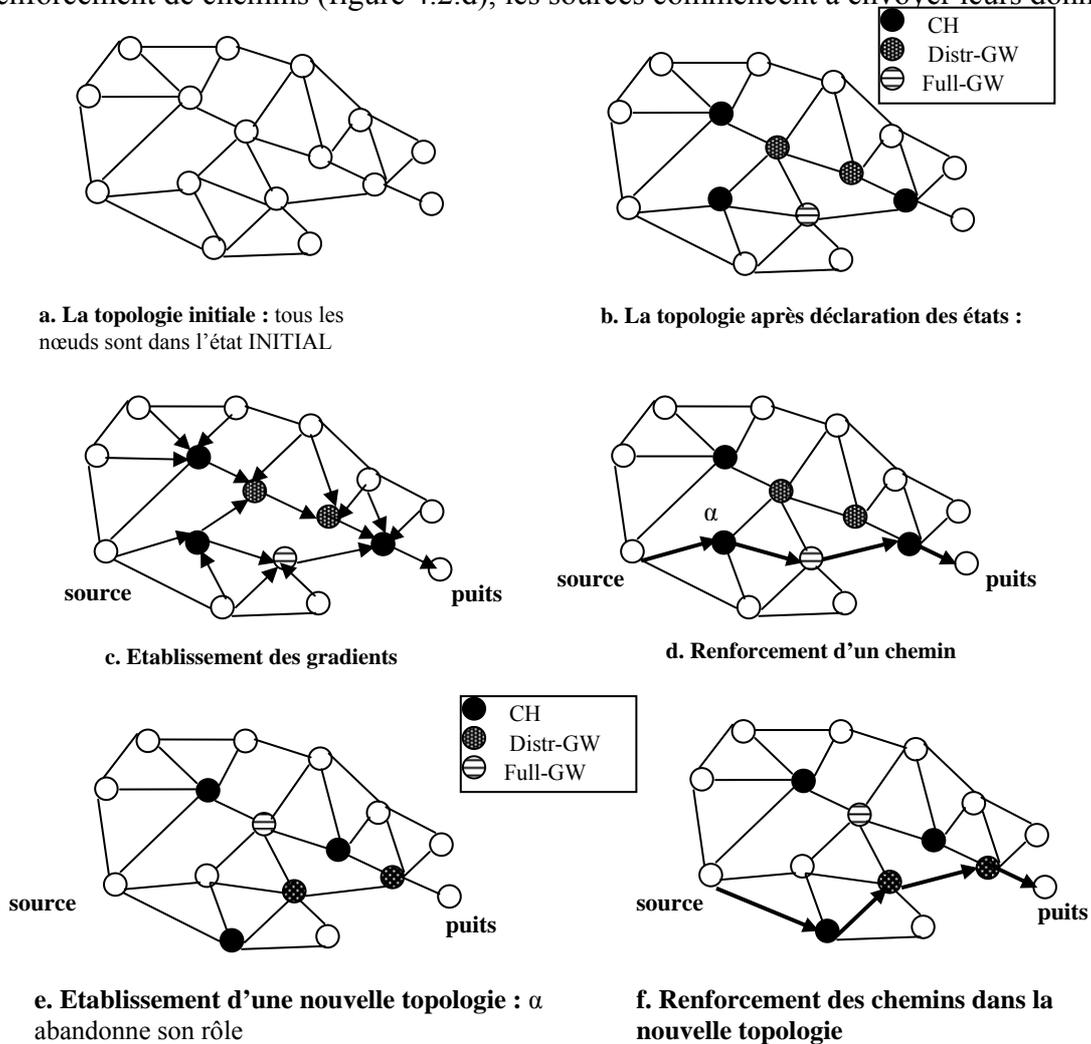
- Les nœuds ordinaires (non passerelles et non têtes de clusters) ne participent pas dans les diffusions non optimisées des messages d'intérêts et ceux explorateurs de données (inondations).

Nous proposons un deuxième point d'interaction qui est l'utilisation de la topologie résultante dans le mécanisme de routage :

- Les têtes de clusters vont jouer le rôle d'entités combinateurs de données. De ce fait, chaque nœud appartenant à un cluster enverra ses données ou les données qui lui sont transmises à son responsable qui par la suite les expédiera en direction du nœud collecteur (puits).

Pour expliquer les effets de cette interaction, nous démontrons dans la figure (4.2) suivante l'établissement des structures de routage de la diffusion dirigée quand elles sont utilisées en combinaison avec le protocole EAPC.

A l'initialisation, tous les nœuds du réseau sont dans l'état *Initial*. Les quantités d'énergie initiales au niveau des nœuds peuvent être égales ou non (cas réel). Les nœuds vont se servir des premiers messages d'intérêts pour mettre en place la nouvelle topologie suivant le pseudo code décrit dans le paragraphe (4.4.3). Une topologie possible est schématisée dans la figure (4.2.a). Après établissement des gradients (figure 4.2.c) et renforcement de chemins (figure 4.2.d), les sources commencent à envoyer leurs données.

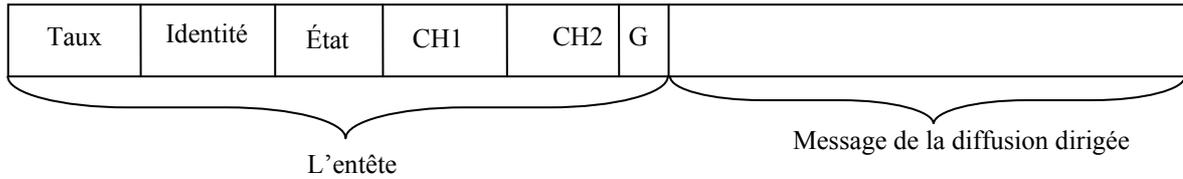


**Figure 4.2 :** Combinaison de la diffusion dirigée avec le EAPC

Quand le taux d'énergie consommée dépasse la valeur de  $\lambda$  au niveau du nœud  $\alpha$  (figure 4.2.d), il abandonne son rôle de tête de cluster. Ainsi, une nouvelle topologie est mise en place. Ceci est réalisé en utilisant les prochains messages circulant dans le réseau (messages de données, intérêts, explorateurs de données).

### 4.5.3 La structure des messages

La structure des messages manipulés est montrée dans la figure 4.3. Les cinq champs sont ajoutés dans tous les messages utilisés dans le paradigme de la diffusion dirigée. La partie message de la diffusion dirigée dans la figure 4.3 peut donc contenir un message d'intérêt, de données ou explorateurs de données.



**Figure 4.3 :** La structure du message manipulé

#### 4.5.4 L'implémentation du protocole

Le protocole EAPC s'exécute en parallèle avec les autres algorithmes de diffusion. Ainsi, il peut être implémenté de manière indépendante aux algorithmes et aux applications de la diffusion dirigée. L'implémentation du protocole peut être réalisée en utilisant les APIs fournies pour la gestion des filtres [35]. Ces APIs permettent la création et la gestion des filtres et des applications. Dans cette architecture, un message arrivant au niveau d'un nœud passe, dans un ordre de priorité, par l'ensemble des filtres définis à son niveau. Le nœud peut alors attacher des données au message, enlever des données, faire des traitements statistiques,...etc.

#### 4.6 Conclusion

Dans le but de conserver l'énergie dans la diffusion dirigée, nous avons présenté dans ce chapitre une amélioration de ce paradigme. Notre amélioration est à double volet : tout d'abord nous avons proposé la prise en compte du facteur 'énergie' dans la formation et la maintenance de la topologie dans un algorithme de clustering. Ensuite combiner l'algorithme proposé avec la diffusion dirigée.

En se basant sur la description de la solution, nous pouvons donner les observations suivantes :

- Par rapport à la solution proposée dans [82]. Nous proposons d'ajouter un champ supplémentaire représentant le taux d'énergie. Nous l'avons rajouté comme attribut dans le message manipulé. Ceci résulte en l'ajout de 8 octets supplémentaires (quatre octets sont réservés au taux et quatre autres pour la manipulation de l'attribut ajouté). Ceci donne une taille supplémentaire de 25 octets dans le message manipulé dans la diffusion dirigée (17 octets ajouté par le clustering passif).
- Une alternance des rôles selon le niveau d'énergie consommée des nœuds est effectuée, ceci est réalisé dans le but d'équilibrer la consommation d'énergie dans le réseau.
- Le degré de connectivité du réseau est réduit en comparaison avec la diffusion dirigée et aussi avec sa combinaison avec le clustering passif. Ceci est dû à la comparaison des états énergétiques entre les nœuds têtes de clusters et passerelles et aussi au relâchement des rôles par les nœuds ayant dépassés la valeur du taux.
- Les nœuds se trouvant dans l'état ordinaire ne participent pas aux diffusions des messages explorateurs ce qui réduit énormément le nombre de diffusions non optimistes.
- Aucun message de contrôle n'est ajouté pour maintenir ou former la topologie.
- Les têtes de clusters jouent le rôle d'entités combinateurs de données.

Dans le but de renforcer ces observations, nous avons mené une évaluation de performances au moyen de simulation. Cette étude va être détaillée dans le chapitre suivant.

## **Chapitre 5**

# **Évaluation de performances de la solution proposée**

### **5.1 Introduction**

Dans le chapitre précédent, nous avons d'abord présenté une solution visant à prendre en considération le facteur « énergie » dans l'algorithme du clustering passif. Ensuite, nous avons proposé sa combinaison avec le paradigme de la diffusion dirigée afin d'atteindre une efficacité en consommation d'énergie en visant surtout à prolonger la durée de vie du réseau. Pour évaluer les performances de la solution proposée nous avons procédé par simulation. Cette dernière est faite pour confirmer ou rejeter les observations déjà faites et accentuer les améliorations et les dégradations apportées par notre solution. Pour cela, nous comparons les trois protocoles suivant : la diffusion dirigée (DD), la diffusion dirigée en combinaison avec le clustering passif (PCDD) et la diffusion dirigée en combinaison avec le protocole de clustering proposé (EAPCDD).

Selon les observations faites dans la section précédente, l'efficacité de notre solution dépend des réponses qui peuvent être apportées aux questions suivantes :

- Est-ce que la solution proposée réalise un équilibre dans la consommation d'énergie ? donc est-ce que la durée de vie du réseau sera prolongée (partitionnement moins rapide du réseau)?
- Est-ce que la combinaison entre la réduction du nombre de messages inondés et l'augmentation de leur taille (ajout du taux) fournit un gain en énergie ?
- Est-ce que le degré de connectivité réduit conduit à une dégradation des performances par rapport au taux de livraison des données et aux délais de transmission?
- Quelle serait la réaction du protocole quant à la dégradation du niveau d'énergie des nœuds ?

Nous justifions tout au long de ce chapitre nos choix de conception et nos paramètres de simulation. Les graphes qui schématisent nos résultats et leurs interprétations seront exposés à la fin du chapitre.

### **5.2 Environnement de simulation**

Peu de projets de simulateur de réseau sansfil/mobile sont mûrs actuellement. Pratiquement, les environnements NS (Network Simulator) [55], Opnet [58] et GloMoSim [89] sont les plate-formes de simulation de réseau les plus avancées et les plus utilisées par la communauté des chercheurs. D'autres simulateurs proposés particulièrement pour les réseaux de capteurs sont apparus [95]. Cependant ils ont été conçus pour valider des

solutions particulières et sont encore en phase de développement. NS et GLoMoSim sont en libre accès aux chercheurs contrairement à Opnet. L'environnement GloMoSim ne contient pas une implémentation du paradigme de la diffusion dirigée. Ainsi, son utilisation exige une phase d'implémentation supplémentaire, difficile et longue vue la nature du paradigme et la diversité de ses algorithmes de diffusion. NS est considéré actuellement l'outil de simulation le plus populaire destiné aux réseaux de capteurs [96]. Il contient l'implémentation de plusieurs protocoles proposés pour ce type de réseaux. Ces protocoles sont situés à différents niveaux de l'architecture. Il comprend principalement une bonne implémentation du paradigme de la diffusion dirigée proche de l'implémentation réelle et qui permet son extension de manière efficace. Aussi, l'utilisation du même environnement de simulation proposé dans [20, 82] permet dans une première étape de valider notre implémentation du clustering passif et surtout de mener une bonne comparaison entre les solutions.

### **5.2.1 Le simulateur NS (Network Simulator)**

NS (Network Simulator) est un simulateur à événements discrets développé par le projet VINT "Virtual InterNetworked Testbed" de l'Université de Californie – Berkely<sup>8</sup>, il est destiné à simuler les protocoles de communication des réseaux filaires et sans fil (locaux et satellitaires) tel que TCP, le routage, multicasting, multimédia, Mobile-IP, ...

Le simulateur NS est en libre accès à tous les chercheurs afin d'unifier les efforts et construire un simulateur de réseau aussi pratique que riche et varié. Nous utilisons la version 2.27 du package NS dans notre environnement de simulation [55]

### **5.2.2 Architecture de NS**

NS est implémenté en C++ et OTcl (Object oriented Tool Command Language) ce qui allie rapidité d'exécution et facilité de paramétrage des simulations. Un code C++ est compilé et donc rapide en exécution cependant difficilement modifiable, il est approprié pour l'implémentation des détails d'un protocole. Le code OTCL est interprété, par conséquent, lent en exécution mais facilement modifiable, il est approprié pour la configuration et le contrôle des scénarios de simulations.

Le package NS inclut une hiérarchie de classe compilée d'objets C++ et une hiérarchie de classe interprétée d'objets OTcl. Ces deux hiérarchies sont étroitement liées ; quand l'utilisateur crée un nouvel objet par l'interpréteur OTcl, un objet correspondant est aussi créé dans la hiérarchie compilée. Bien entendu, les objets peuvent être accédés aussi bien en OTcl qu'en C++ grâce à la mise en place de procédures d'appel entre OTcl et C++. Les trois classes les plus importantes pour établir cette correspondance entre les objets compilés et les objets interprétés sont:

- La classe Tcl contient les méthodes que le code C++ utilise pour accéder et communiquer avec l'interpréteur. La classe Tcl encapsule l'instance en cours de l'interpréteur OTcl, appelle les procédures OTcl via l'interpréteur, recherche et retourne les résultats à l'interpréteur et surtout sauvegarde et interroge les objets « TclObjects ».
- La classe TclObject est la classe de base pour tous les objets (interprétés et compilés) du simulateur. Un objet de TclObject est créé et configuré par l'utilisateur via l'interpréteur OTcl. Cette classe permet aussi de lier cinq types de

---

<sup>8</sup> <http://www.isi.edu/nsnam/ns/>

variables (entier, réel, bande passante, temps, booléen) de la hiérarchie compilée avec ceux de la hiérarchie interprétée.

- La classe TclClass est une classe très importante lors de l'implémentation d'un nouveau protocole. Elle offre deux fonctionnalités : (1) construire la hiérarchie interprétée pour correspondre à la hiérarchie compilée ; (2) offrir des méthodes pour instancier un nouveau objet TclObject.

Bien que le simulateur NS soit le plus utilisé par la communauté des chercheurs vu la diversité des protocoles qu'il simule, il est complexe en terme de code. Le nombre de composants du simulateur NS étant énorme on ne peut le représenter de manière globale. Dans le reste du document, nous nous concentrons sur les composants de NS nécessaires pour notre travail à savoir : la diffusion dirigée et le modèle d'énergie adopté.

### 5.2.3 La diffusion dirigée dans NS

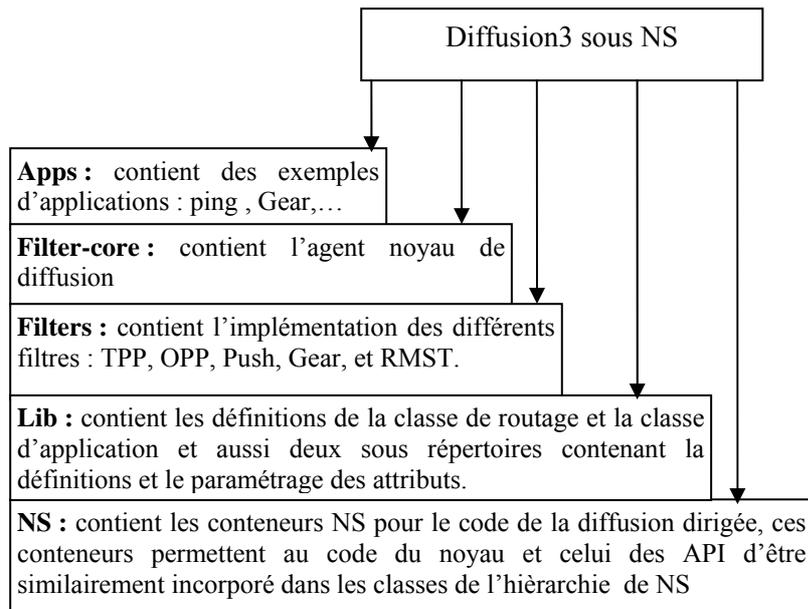
La proposition de la diffusion dirigée a été faite dans un projet de collaboration nommé SCADDS entre CENS (Center for Embedded Networked Sensing) et LECS (Laboratory for Embedded Collaborative Systems at UCLA). Il y a deux implémentations de la diffusion dirigée fonctionnant chacune dans un environnement différent :

- **La diffusion complète (Full diffusion)** : cette implémentation est destinée à son tour à deux environnements. Le premier est constitué des nœuds capteurs (32 bits) et de PCs en utilisant des interfaces radios appropriées. Le second est le simulateur de réseau NS version 2.1b9. Deux implémentations de la diffusion dirigée dans NS ont été établies ; une version simple et une autre version plus élaborée que nous utilisons dans nos simulations (nous discuterons notre choix de la version dans la partie qui présente les éléments de la simulation)
- **La diffusion optimisée (Tiny diffusion)** : cette implémentation est maintenue au niveau du centre CENS, elle est destinée aux environnements Motes UCB (capteurs réels).

Le modèle de la diffusion dirigée dans NS consiste en :

1. **Le noyau de la diffusion** : c'est la partie responsable de l'envoi/réception des messages au/du réseau.
2. **la bibliothèque de la diffusion** : elle fournit les APIs (Application Programming Interfaces) pour la création et la gestion des filtres et des applications de diffusion.
3. **la couche application** : elle contient les différents filtres (Pull à deux phases, Pull à une phase, Push, Gear) et applications (ping, applications correspondantes aux différents filtres) relatives à la diffusion dirigée.

L'implémentation de la diffusion dirigée dans NS est présente sous le répertoire diffusion3 qui a la structure suivante :



**Figure 5.1** : Structure du répertoire diffusion3 dans NS

Le mécanisme de publication/souscription est utilisé par les applications pour l'envoi des messages d'intérêts et les messages de données au réseau. L'agent noyau et l'agent application sont attachés à des numéros de ports précis dans NS. Les applications de diffusion attache au noeud capteur les agents lui permettant de publier/souscrire/envoyer les données.

#### 5.2.4 Le modèle d'énergie dans NS

Le modèle d'énergie dans NS est un attribut de la classe *nœud*. Il représente le niveau d'énergie d'un nœud mobile. Il est caractérisé par une valeur initiale (*initialEnergy\_*), une valeur décrémentée lors de chaque transmission (*txPower\_*) et une valeur décrémentée suite à la réception d'un paquet (*rxPower\_*). Les valeurs d'énergie requises pour l'émission et la réception dépendent de l'interface radio utilisée dans la simulation [55]. Un nœud mobile peut aussi être dans l'état *idle*. La quantité d'énergie consommée dans ce mode est dépendante de la couche MAC (Medium Access Control) utilisée.

Ce modèle a été étendu pour prendre en considération l'énergie dissipée dans l'état *sleep*. Ceci a été fait pour supporter les simulations d'un protocole de la couche MAC (Sensor MAC S-MAC) destiné aux réseaux de capteurs [64] et d'autres protocoles proposés dans le domaine de la conservation d'énergie.

#### 5.2.5 Implémentation de la solution dans NS

Le protocole peut être implémenté sous forme de deux filtres englobant le filtre original de la diffusion dirigée (figure 5.2). Un filtre de prétraitement appelé pré-filtre et un filtre de post-traitement appelé post-filtre. Le filtre original que nous utilisons est celui relatif à l'algorithme de diffusion du Pull à deux phases appelé aussi *le filtre gradient*. Une grande priorité est attribuée au pré-filtre pour lui permettre d'intercepter en premier les messages entrants (2). Ce filtre enlève l'entête ajouté par le EAPC et change l'état interne des nœuds selon les informations contenues dans les messages reçus. Le message est alors passé du pré-filtre au gradient (3). Celui-ci reçoit les messages venant de la part du pré-filtre (4), les traite et les envoie vers le noyau de la diffusion (5). Le post-filtre ayant une priorité inférieure à celle du pré-filtre et aussi à celle du gradient, effectue les derniers

traitements sur les messages avant d'être expédiés vers le réseau (6, 7, 8). Il est responsable des changements des états des nœuds vers des états externes visibles dans le réseau. Il est aussi chargé d'ignorer les messages à inonder (intérêts et explorateurs de données). La création et la gestion des filtres sont faites de la même manière que celles des applications de diffusion. Les deux dérivent de la même classe (DiffApp). Nous avons tout d'abord procédé à l'implémentation du clustering passif ensuite, ajouté notre proposition selon le pseudo-code présenté précédemment.

Dans notre simulation nous avons utilisé un seul filtre (gradient) pourtant cette implémentation est valide même en cas d'utilisation d'autres filtres à savoir : Pull à une phase, Push et Gear.

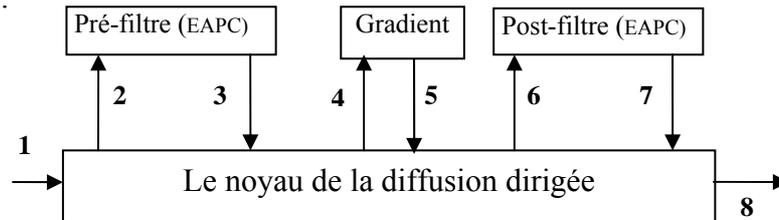


Figure 5.2 : schéma d'implémentation

### 5.3 Les éléments de la simulation

Notre simulation comporte trois éléments : la version du protocole de la diffusion dirigée, l'application utilisée qui définit la charge du réseau et le modèle d'énergie :

#### 5.3.1 La version de la diffusion dirigée

Deux implémentations de la diffusion dirigée existent dans NS. Une version simplifiée implémentée par Intanagonwiwat [20] pour l'évaluation du protocole et une autre version plus complète (diffusion 3). L'implémentation est décrite dans [18] et fournit une évaluation plus réaliste du protocole. Les différences majeures entre les deux implémentations sont :

- *Taille des messages* : dans la version simplifiée un message d'intérêt a une taille égale à 32 octets, et un message de données de taille de 64 octets. Dans celle complète, la taille des messages est beaucoup plus réaliste. Elle dépend de l'application utilisée. Pour l'application 'ping', la taille d'un message d'intérêts est de 88 octets et celle d'un message de données est de 116 octets.
- *La manipulation de l'horloge* : la gestion de l'horloge définit la charge du réseau dans la simulation. Elle définit les intervalles utilisés pour l'envoi des messages (intérêts, explorateurs de données, données).
- *Les extensions* : la version complète donne une liberté d'implémentation au développeur dans la création et la gestion de leurs applications et filtres. Les APIs utilisées sont les mêmes utilisées dans l'implémentation réelle du protocole.

Pour ceci et dans le but de mener une comparaison avec la solution proposée dans [82], nous choisissons la version complète se trouvant dans le répertoire diffusion3 de NS.

#### 5.3.2 La charge du réseau :

Dans la version complète de la diffusion dirigée, l'envoi des messages d'intérêts est fait chaque 25 secondes. Les messages explorateurs de données sont envoyés chaque 60 secondes et les messages de données chaque 5 secondes. Pour mener une bonne

comparaison, nous avons changé l'application 'ping' pour envoyer les intérêts chaque 20 secondes et les données chaque 2 secondes. Ce qui résulte en l'envoi de 10 messages par inondation d'intérêt.

### 5.3.3 Le modèle d'énergie

Chaque message envoyé, reçu ou même écouté (overheard) requiert une quantité d'énergie qui sera réduite de la quantité d'énergie du nœud. Aussi, la quantité d'énergie consommée dans le mode *idle* doit être pris en charge. Celle-ci dépend de la couche MAC utilisée dans la simulation. Dans notre cas, l'implémentation NS de la 802.11 tel que proposé dans l'évaluation du protocole de la diffusion dirigée. Nos choix sont faits dans le but de comparabilité. Ainsi, nous utiliserons le même modèle adopté dans [82, 20] à savoir le modèle de la carte WLAN PCMCIA. Dans ce modèle, l'énergie consommée dans l'envoi est de 0.666 W et celle de réception est de 0.395 W. L'énergie consommée dans l'état *idle* est supposée nulle. Ceci n'est pas le cas dans l'implémentation NS de la 802.11 parce que l'énergie prévue dans le cas *idle* est voisine, dans cette implémentation, à celle de réception. Ceci est dû au fait qu'un nœud utilisant la 802.11 (CSMA MAC) [90] doit écouter le canal de transmission de manière permanente. Cette technique peut alors masquer les effets de la couche réseau (routage) dans les applications avec de longues périodes d'inactivité. Nous choisissons une quantité d'énergie nulle dans le cas *idle* dans le but de focaliser l'étude sur la quantité d'énergie consommée dans les différents protocoles de routage.

## 5.4 Les scénarios

Nous proposons deux types de scénarios dans le but de considérer le comportement des trois solutions (DD, PCDD, EAPCDD). Tous les nœuds du réseau sont éparpillés de manière aléatoire dans une zone de taille fixe ou variable. La portée de communication est considérée la même pour chaque nœud dans tous les scénarios.

- **Cinq sources et cinq puits :** Ce type de scénarios sert à examiner l'effet de la topologie du réseau sur les différentes solutions. Ainsi, cinq sources et cinq puits (un trafic régulier des messages) sont utilisés avec un nombre de nœuds dans une zone de taille fixe ou variable. Le premier cas vise à garder la taille de la zone de simulation constante alors que le deuxième vise à garder fixe la densité moyenne.
- **Un nombre variable de sources/puits :** Ce type de scénarios sert à examiner l'effet du trafic en changeant le nombre de sources et de puits. En conséquence, un nombre fixe de nœuds (100 noeuds) est utilisé dans une zone de taille fixe en utilisant un seul puits et un nombre variable de sources ou bien une seule source avec un nombre variable de puits.

**1<sup>er</sup> type :**

### Configuration 1

Cinq sources et cinq puits, zone fixe	
Rang de transmission	40 m
Taille de la zone topographique	160*160m <sup>2</sup>
Nombre de noeuds	25, 50, 75, 100, 125, 150

**Configuration 2**

<b>Cinq sources et cinq puits, zone variable (densité moyenne fixe)</b>						
Rang de transmission	40m					
Taille de la zone topographique	50*50m <sup>2</sup>	100*100m <sup>2</sup>	150*150m <sup>2</sup>	200*200m <sup>2</sup>	250*250m <sup>2</sup>	300*300m <sup>2</sup>
Nombre de noeuds	25	50	75	100	125	150

**2<sup>ème</sup> type :**

**Configuration 1**

<b>1 puits, nombre de sources variable</b>	
Rang de transmission	40m
Taille de la zone topographique	160*160m <sup>2</sup>
Nombre de noeuds	100
Nombre de sources	1, 2, 4, 8, 10

**Configuration 2**

<b>1 source, nombre de puits variable</b>	
Rang de transmission	40m
Taille de la zone topographique	160*160m <sup>2</sup>
Nombre de noeuds	100
Nombre de puits	1, 2, 4, 8, 10

**5.4.1 Génération des scénarios**

Les scénarios ont été générés à l'aide de l'outil TopoGen [91]. Ce dernier est un outil de création de topologies aléatoires pour des expériences de simulation du paradigme de la diffusion dirigée. Il peut être configuré pour produire des fichiers de code TCL destinés à l'utilisation dans le simulateur NS. Pour chaque configuration, nous avons généré de manière aléatoire 5 scénarios différents. La valeur correspondante à un point sur les graphes résultats est la moyenne de cinq valeurs obtenues par exécution des cinq scénarios.

**5.4.2 Les paramètres de la simulation**

Les paramètres utilisés dans notre simulation sont :

Le temps de simulation	1000 secondes		
Modèle de propagation	TwoRayGround		
La couche MAC	La 802.11		
Antennes	Omnidirectionnelles		
Energie initiale	1000 joules	4 ou 30 joules	variable
Energie de transmission	0.666 W		
Energie de réception	0.395 W		
Energie dans l'état <i>idle</i>	0 W		
Taille d'un message d'intérêt	88 (+25) octets		
Taille d'un message de données	116 (+25) octets		
Taille d'un message explorateur de données	116 (+25) octets		

Nous avons utilisé plusieurs valeurs pour la quantité d'énergie initiale au niveau des nœuds. Ceci est fait dans le but de tester le comportement des trois solutions dans plusieurs cas. Le premier est celui où les nœuds du réseau ont des quantités d'énergie suffisantes pour terminer le processus de simulation (1000 secondes). Dans le deuxième cas, les nœuds ont des quantités d'énergie réduites (30 joules pour le premier type de scénarios et 4 joules pour le deuxième). Ceci nous permet d'examiner la durée de vie des différents nœuds dans le réseau en présence de petites quantités d'énergie et donc de considérer le partitionnement du réseau. Nous avons aussi simulé des scénarios avec des quantités d'énergie variables.

## 5.5 Les métriques

Nous avons choisi un ensemble de métriques pour évaluer les trois solutions et exposer leurs différentes caractéristiques. Chacune des métriques est évaluée en fonction du nombre de nœuds déployés, de la charge de données, et de la densité des nœuds dans le réseau. Selon la valeur attribuée à la quantité d'énergie initiale au niveau des nœuds, nous proposons l'évaluation de deux ensembles de métriques :

### 5.5.1 1<sup>er</sup> cas : une quantité d'énergie suffisante

Les métriques utilisées sont souvent proposées dans l'évaluation des protocoles de routage dans les réseaux Ad hoc et capteurs [20]. Elles servent à capter le comportement des protocoles et leur capacité de réaliser une communication efficace des données. Les métriques utilisées dans ce cas sont :

### 1. Le délai moyen

L'information transportée dans les réseaux de capteurs est souvent de type '*event driven*' (surveillance de phénomène). Cette information est extrêmement critique, et le délai d'acheminement de l'information des noeuds capteurs qui ont détecté l'évènement vers le puits doit être minime. Dans une application de type surveillance de gaz chimique, les concepteurs aimeraient bien détecter la présence du gaz le plus tôt possible. De ce fait, nous avons décidé de nous intéresser aux délais d'acheminement de l'information des noeuds capteurs vers le puits en terme de délais moyens observés [37]. Cette métrique calcule la latence moyenne observée entre la transmission d'un évènement et sa réception au niveau de chaque puits. Elle indique l'exactitude temporelle que fournit le réseau de capteurs en utilisant un protocole de routage

### 2. Le taux de livraison (*goodput*)

Cette métrique représente le rapport entre le nombre de paquets de données envoyés par les sources (producteurs) et le nombre de paquets de données reçus par les puits (consommateurs) [18]. Dans la diffusion dirigée (réseau de capteurs), le mot paquet de données est remplacé par *évènement*. Dans notre simulation on s'intéresse au nombre d'évènements distincts. Ainsi, la métrique calcule le rapport entre le nombre d'évènements envoyés par les sources et le nombre d'évènements *distincts* reçus par les puits.

Soient :  $V_s$  = le nombre d'évènements envoyés par les sources.

$V_c$  = le nombre d'évènements distincts reçus par les puits.

$$\text{Alors, le rapport } r = \frac{V_c}{V_s}$$

Cette métrique indique le rapport de perte d'un protocole de routage et l'effort requis pour recevoir des données. Dans le scénario idéal le rapport devrait être égal à 1.

Rq : cette métrique est calculée dans les deux cas (énergie suffisante ou insuffisante)

### 3. La quantité d'énergie moyenne par évènement distinct

Cette métrique mesure le rapport entre la quantité d'énergie moyenne consommée par les noeuds du réseau et le nombre d'évènements distincts reçus par les puits. Ainsi, elle indique la quantité d'énergie nécessaire à la réception d'un évènement. Soient :

$E_{Ni}$  la quantité d'énergie initiale d'un noeud ;  $E_{Nf}$  la quantité d'énergie d'un noeud à la fin de la simulation et  $N$  le nombre de noeuds dans la simulation.

La quantité d'énergie moyenne consommée par tous les noeuds du réseau est donnée par :

$$E_m = \frac{\sum_{k=1}^{k=N} (E_{Nik} - E_{Nfk})}{N}$$

La métrique est calculée par :

$$E_{mv} = \frac{E_m}{V_c}$$

Où  $V_c$  est le nombre d'évènements distincts reçus par le puits.

### **5.5.2 2<sup>ème</sup> cas : une quantité d'énergie insuffisante**

Afin de mettre le point sur les apports de notre proposition par rapport aux autres solutions, nous avons utilisé deux métriques différentes qui nous renseignent mieux sur la durée de vie et le partitionnement du réseau ainsi que la qualité de service qu'offrent les différentes solutions en présence d'une petite quantité d'énergie.

#### **1. Le temps avant le partitionnement du réseau**

Cette métrique peut être interprétée de différentes manières. Elle peut être définie par la présence de chemins non valides dans les tables de routage au niveau des différents nœuds et donc le temps où le premier nœud est épuisé en énergie. Une autre définition de cette métrique a été donnée dans [25], indiquant le temps où la première source est épuisée en énergie. Dans notre cas nous utilisons la première définition. Cette métrique indique la durée de vie du réseau. Elle peut aussi indiquer si le protocole répartit ou non les charges au niveau des nœuds du réseau.

#### **2. Le nombre d'évènements livrés avant le partitionnement du réseau**

Cette métrique calcule le nombre d'évènements envoyés et bien reçus avant le partitionnement du réseau. Dans un réseau de capteurs, le nombre d'évènements reçus peut indiquer la qualité de service qu'offre un protocole de routage. Ce nombre est d'une grande importance dans la plupart des applications des réseaux de capteurs (surveillance, reportage des données, ...). Cette métrique indique la capacité de livraison du protocole de routage dans le cas où le niveau d'énergie des nœuds est petit.

*Ces métriques sont calculées à partir des fichiers traces générés par NS, vue leur taille énorme (souvent supérieure à 300 Mo), nous avons écrit des programmes AWK pour automatiser le traitement des fichiers traces. La section qui suit présente les résultats obtenus ainsi que leurs interprétations.*

## **5.6 Résultats et interprétation des résultats**

Nous allons présenter les résultats obtenus durant nos expériences de simulation. Une première partie présentera le comportement des trois solutions quand les nœuds ont des quantités d'énergie suffisante. Leur comportement en présence d'une quantité d'énergie insuffisante sera étudié dans la deuxième partie. Finalement, une comparaison entre la combinaison de la diffusion dirigée avec le clustering passif (PCDD) et celle avec le protocole EAPC en supposant des quantités d'énergie variables sera présentée dans une troisième partie.

### **5.6.1 Une quantité d'énergie suffisante**

Dans cette partie, les nœuds du réseau ont des quantités d'énergie suffisantes. Ceci est fait dans le but d'examiner les performances de notre solution en considérant les effets de nos changements sur la qualité des communications et la quantité d'énergie moyenne consommée.

#### **5.6.1.1 Effet de la topologie du réseau**

Pour examiner l'effet de la topologie du réseau sur les différentes solutions nous considérons deux cas. Dans le premier, un nombre de nœuds variable (entre 50 et 150 nœuds ; en ajoutant 25 nœuds dans chaque scénario) sont placés dans une zone de taille fixe (160 \*160 m<sup>2</sup>). Dans le deuxième cas, un nombre de nœuds variable sont aussi placés dans une zone de taille variable (de 100\*100 m<sup>2</sup> à 300\*300m<sup>2</sup> ; 50\*50m<sup>2</sup> pour chaque 25

nœuds) dans le but d'avoir une densité moyenne fixe des nœuds dans le réseau. Le trafic dans les deux cas est régulier, il est généré par cinq sources et cinq puits.

Nous remarquons sur les deux figures (5.3.a et 5.3.b) que le taux de livraison offert par notre solution est proche de celui offert par l'application du clustering passif avec la diffusion dirigée à partir de 75 nœuds. Les deux solutions offrent de meilleurs résultats par rapport à la diffusion dirigée quand le nombre de nœud augmente. Avec l'augmentation du nombre des nœuds, le clustering (PC ou EAPC) aide à réduire la topologie communicante. Nous avons remarqué que ceci aide à réduire le nombre de collisions et donc à avoir de meilleurs taux de livraison. Pourtant, les trois taux de livraisons obtenus par les trois solutions restent élevés ( $> 0.96$ ) [82]. Ce taux a atteint la valeur de 1 (donc tous les événements envoyés sont reçus au niveau des puits) dans l'application du PC et de EAPC même quand le nombre de nœuds augmente.

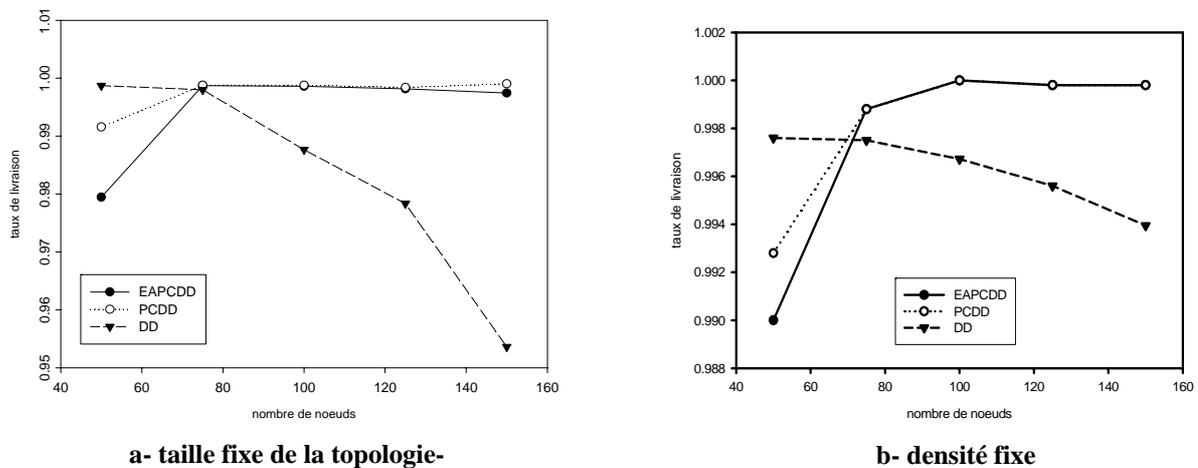


Figure 5.3 : le taux de livraison (goodput)

La figure 5.4 présente les résultats relatifs aux délais moyens des trois solutions. Aussi dans ce cas, les délais offerts par l'application du clustering passif et celui de notre solution sont proches et toujours inférieurs à ceux de la diffusion dirigée. Les valeurs obtenues sont toujours comprises entre 0.1s et 0.4 s dans les deux cas (taille et densité fixe). Pour la diffusion dirigée, le délai augmente avec l'augmentation du nombre des nœuds. Dans le cas d'une taille fixe de topologie (figure 5.4.a), le délai augmente d'une manière significative et atteint la valeur de 6 secondes.

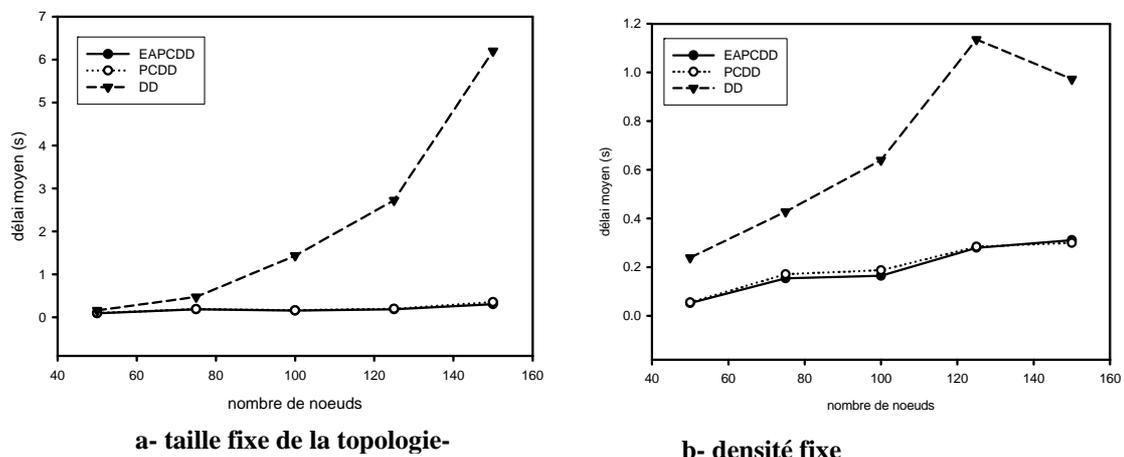


Figure 5.4 : Le délai moyen

La quantité d'énergie moyenne consommée par évènement est schématisée dans la figure 5.5. La quantité d'énergie consommée dans notre solution est légèrement inférieure à celle obtenue dans l'application du clustering passif. Les deux solutions offrent en revanche de meilleurs résultats par rapport à ceux de la diffusion dirigée. L'application du clustering avec la diffusion dirigée (PC ou EAPC) diminue le nombre de transmissions inutiles et donc conserve de l'énergie dans le paradigme. Nous pouvons expliquer cette petite amélioration dans notre solution par rapport au clustering passif par la diminution du nombre de nœuds responsables des opérations d'inondation. Quand les nœuds têtes de clusters et passerelles relâchent leurs rôles (en cas de comparaison ou quand les valeurs des taux dépassent la valeur de  $\lambda$ ), ils passent à l'état initial et ordinaire et donc n'effectuent pas les inondations des messages. La différence en terme de quantité d'énergie entre notre solution et celle du clustering passif est égale en moyenne à 5 joules, ce qui est non négligeable dans le cas des réseaux de capteurs.

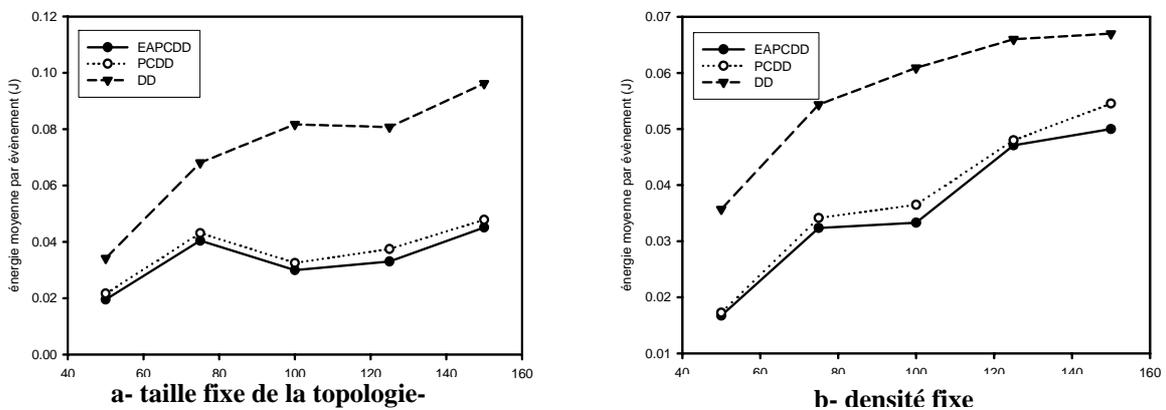


Figure 5.5 : La quantité d'énergie moyenne consommée par évènement

A partir de cette expérience nous pouvons dire que même si la quantité d'énergie moyenne est réduite ceci n'influe pas sur le taux de livraison ou le délai offert par notre solution. Au contraire, les transmissions inutiles sont omises et cela aide à donner de meilleurs résultats. Nous remarquons sur les trois figures que dans le cas d'une taille de topologie fixe la valeur du taux dans la diffusion dirigée diminue à une valeur de 0.95. Le délai atteint la valeur de 6 secondes et la quantité d'énergie augmente et atteint une valeur de 0.1 (joules/nœuds/évènements). Dans ce cas, le nombre des nœuds est augmenté dans une même taille de la zone et donc le nombre de voisins de chaque nœud augmente. En conséquence, un grand nombre de diffusions non optimisées sont effectuées. Ceci influe sur les performances de la diffusion dirigée. Cependant, l'application du clustering stabilise l'augmentation.

### 5.6.1.2 Effet du trafic

Pour examiner l'effet du trafic, nous avons placé 100 nœuds dans une zone de taille fixe (160\*160 m<sup>2</sup>) et varié le nombre de sources et de puits. Ces derniers sont responsables de générer le trafic dans le réseau (messages).

La figure 5.6 montre l'effet du trafic sur le taux de livraison. Nous remarquons qu'à partir de 2 sources (figure 5.6.a), les résultats obtenus pour le clustering passif et notre solution sont meilleurs de ceux de la diffusion dirigée. Dans certains cas (4, 6, et 8 sources) notre solution donne de meilleurs résultats par rapport aux autres solutions. Avec un nombre de sources égale à 1, la diffusion dirigée donne de meilleurs résultats. Le trafic généré par un seul puits et une seule source n'est pas grand et la réduction de la topologie

(en appliquant le clustering) pénalise la livraison de message. Dans le deuxième cas (nombre de puits variables figure 5.6.b), le taux de livraison de donné par notre solution et celle du clustering passif est toujours égal à 1 (valeur maximale) et donc meilleur de celui donné par la diffusion dirigée (les deux graphes sont confondus).

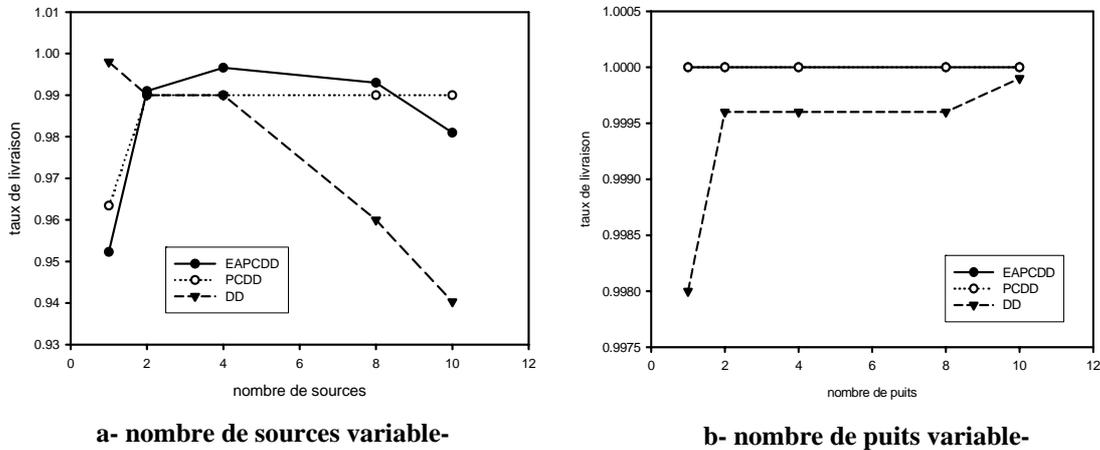


Figure 5.6 : Le taux de livraison (goodput)

La figure 5.7 montre l'effet du trafic sur le délai moyen des messages. Dans les deux cas (a et b), notre solution donne des résultats égaux ou bien meilleurs par rapport au clustering passif, et meilleurs par rapport à la diffusion dirigée. Dans le premier cas, les valeurs du délai obtenues par notre solution ne dépassent pas 0.23 secondes. Dans le deuxième cas, elles sont comprises entre 0.0078 secondes et 0.0604 secondes. Alors que pour la diffusion dirigée ces valeurs atteignent 2.2 secondes dans le premier cas et 0.11 secondes dans le deuxième cas. Cette différence entre les valeurs des délais dans la diffusion dirigée est expliquée par l'augmentation du trafic dans le premier cas dû à l'augmentation du nombre de sources. Cette augmentation est plus significative par rapport au deuxième cas parce que le taux d'envoi des intérêts est beaucoup plus inférieur au taux d'envoi des données (0.05 pour les messages d'intérêts et 0.5 pour les messages de données).

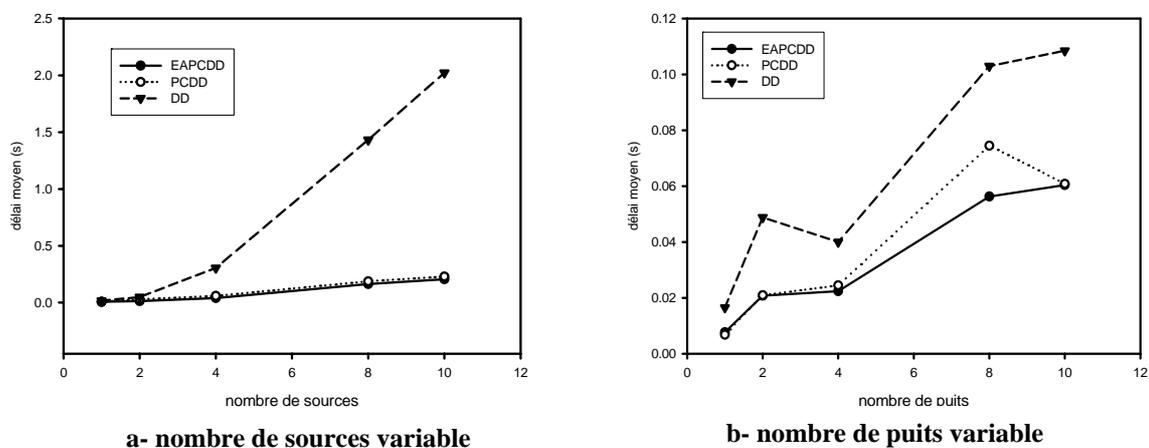


Figure 5.7 : Le délai moyen

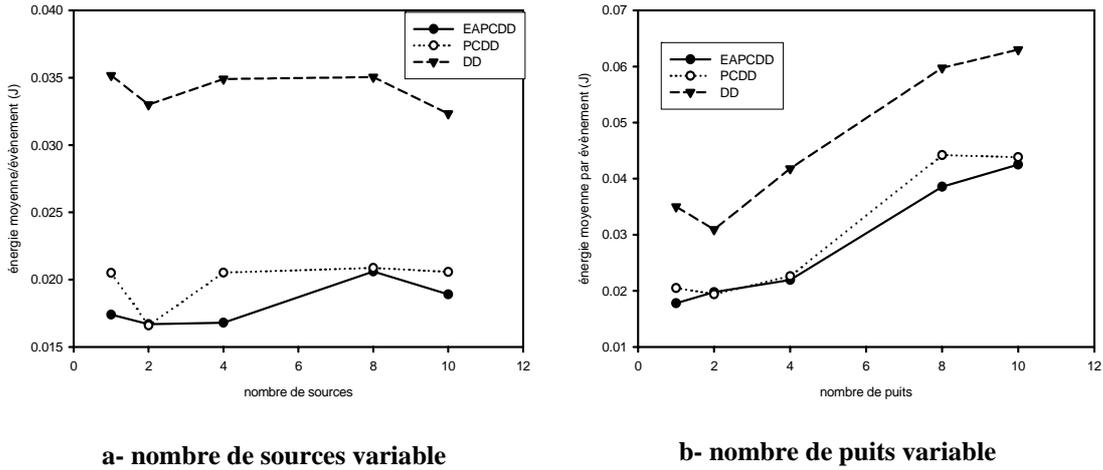


Figure 5.8 : La quantité d'énergie moyenne consommée par évènement

La figure (5.8) présente la quantité moyenne d'énergie consommée par évènement. Selon les résultats obtenus, nous pouvons dire que notre application donne en moyenne de meilleurs résultats par rapport au clustering passif et à la diffusion dirigée. Les valeurs obtenues par les trois solutions dans le premier cas sont inférieures à celles obtenues dans le deuxième cas. Ceci est dû au fait que le nombre d'évènements dans le premier cas est grand (généré par plusieurs sources en augmentant le nombre de sources) donc la quantité d'énergie diminuera en la divisant par un grand nombre.

### 5.6.2 Une quantité d'énergie insuffisante

Dans cette partie, nous allons présenter les résultats obtenus par l'exécution des scénarios précédents dans le cas où les nœuds ont des petites quantités d'énergie. Ces expériences permettent de considérer les apports de notre proposition par rapport aux autres solutions.

#### 5.6.2.1 Effet de la topologie du réseau

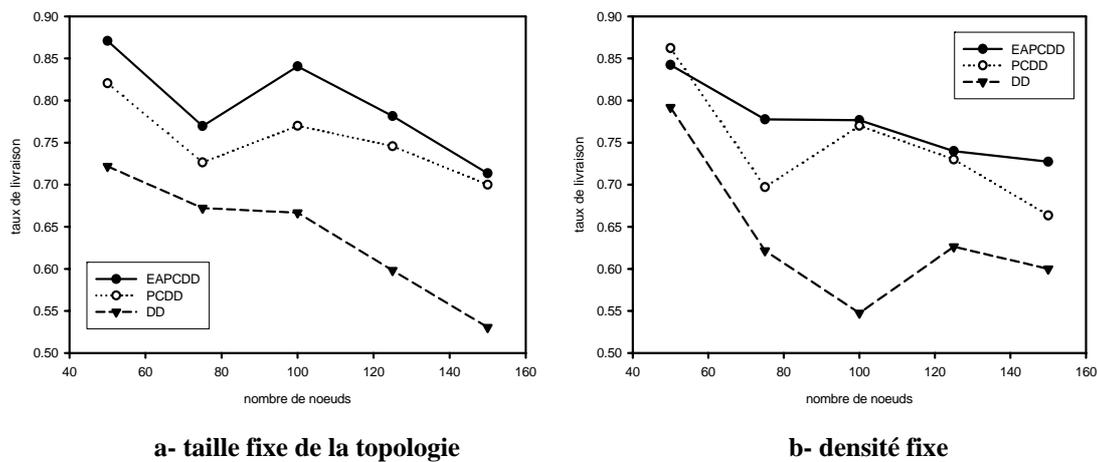


Figure 5.9 : Le taux de livraison

La figure 5.9 montre l'effet de la topologie sur le taux de livraison quand la quantité d'énergie initiale n'est pas suffisante (30 joules) pour terminer le processus de simulation

(1000 secondes). Nous remarquons que notre solution offre de meilleurs résultats par rapport aux deux autres solutions. Ce taux diminue en augmentant le nombre de nœuds mais reste meilleurs que les taux obtenus dans la diffusion dirigée et dans l'application du clustering passif.

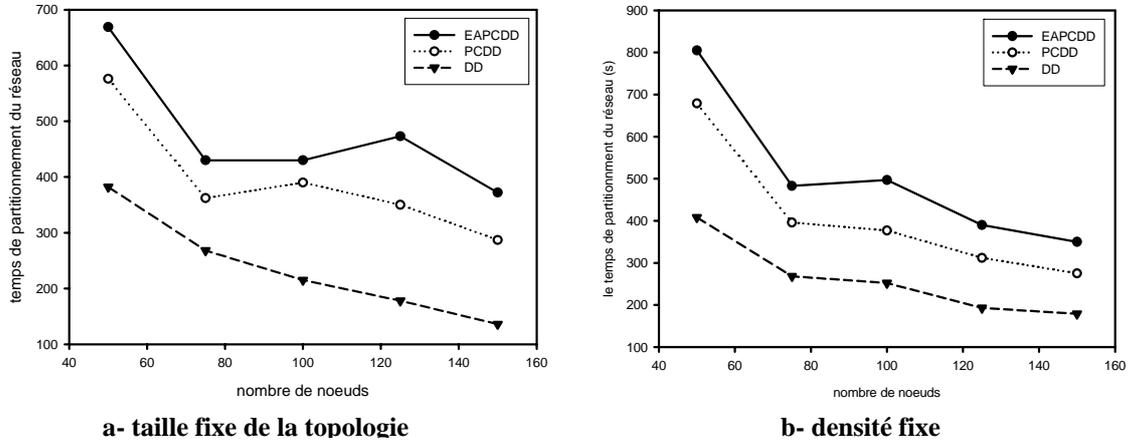


Figure 5.10 : Le temps de partitionnement du réseau

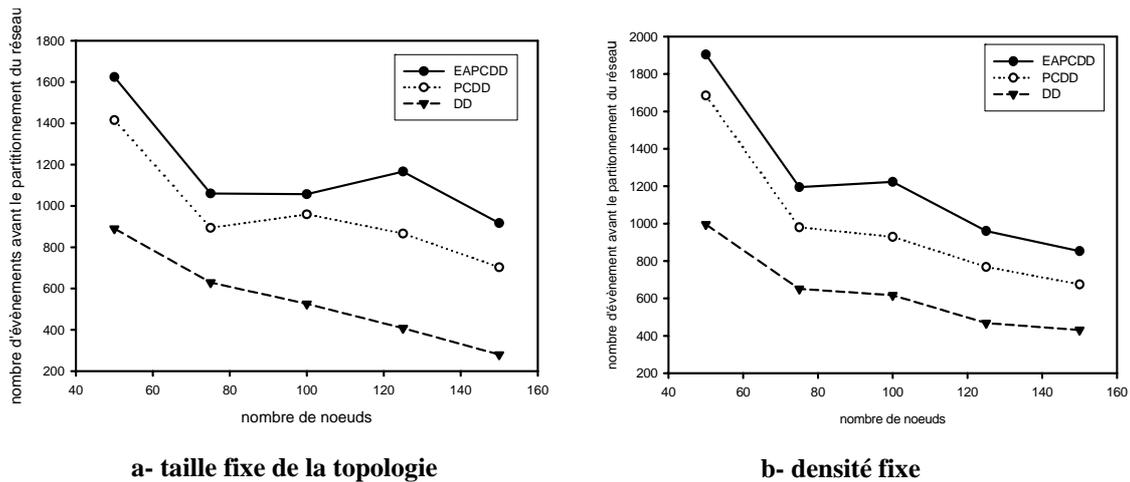


Figure 5.11 : Le nombre d'évènements avant le partitionnement du réseau

Le temps de partitionnement du réseau dans les trois solutions est présenté dans la figure 5.10. Cette dernière montre une amélioration nette dans le temps de partitionnement obtenu dans notre solution par rapport aux deux autres. Dans le cas d'une taille de topologie fixe cette amélioration est comprise entre 4,5% et 12,3% par rapport à l'application du clustering passif et entre 21,5% et 29,5% par rapport à la diffusion dirigée. Dans le cas de densité fixe, elle est comprise entre 7,5% et 13 % par rapport au clustering passif et entre 17,1% et 39,7% par rapport à la diffusion dirigée. Ceci explique ce qui est présenté dans la figure 5.11. Ainsi, le nombre d'évènements avant le partitionnement du réseau obtenu dans notre solution est dans tous les scénarios meilleur que ceux obtenus par les deux autres solutions.

### 5.6.2.2 Effet du trafic

Les trois figures suivantes montrent l'effet du trafic sur les trois solutions en présence de petites quantités d'énergie. A partir de deux sources ou deux puits (augmentation du

trafic), notre solution donne de bons résultats par rapport aux deux autres en terme de taux de livraison, de temps de partitionnement et de nombre d'évènements avant ce dernier.

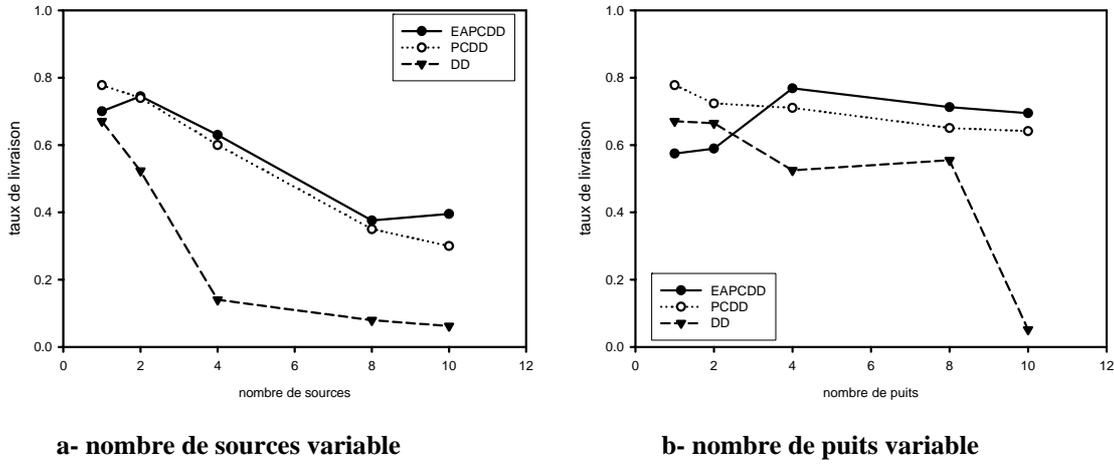


Figure 5.12 : Le taux de livraison

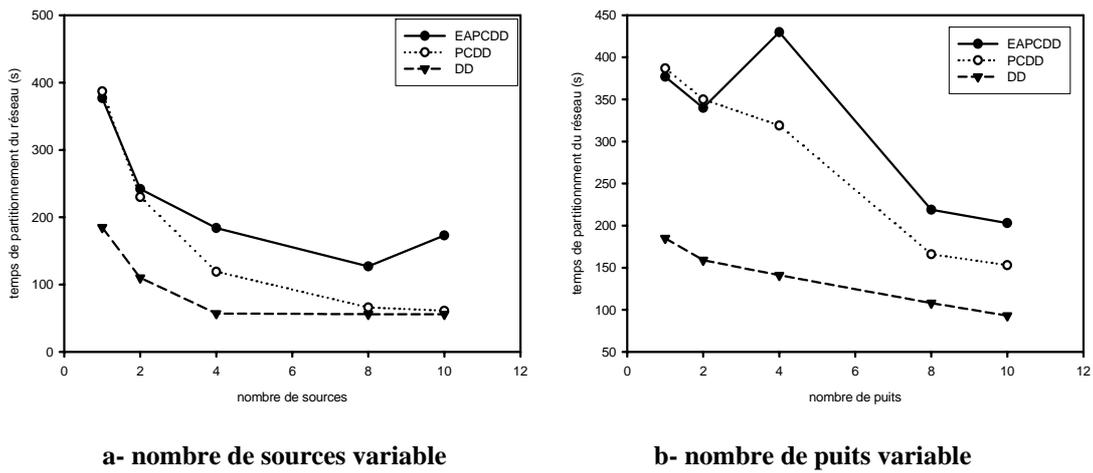


Figure 5.13 : Le temps de partitionnement du réseau

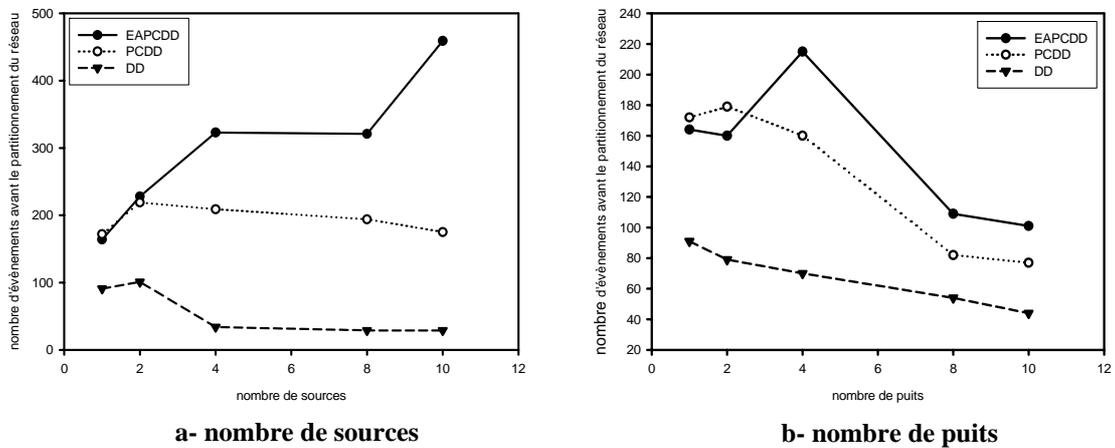


Figure 5.14 : Le nombre d'évènements avant le partitionnement du réseau

Dans la figure 5.13, le temps de partitionnement du réseau dans notre solution est meilleur avec l'augmentation de la charge du réseau (nombre de puits et de sources). Dans le cas d'un nombre de sources variable l'amélioration a atteint 11,2% par rapport au clustering passif et 19.2% par rapport à la diffusion dirigée. Dans le cas d'un nombre de puits variable, cette amélioration atteint 11,1% par rapport au clustering passif et 28,9% par rapport à la diffusion dirigée. Dans la figure 5.14, le nombre d'évènements augmente avec l'augmentation du nombre de sources. Ceci est dû au fait que le nombre d'évènements générés par les sources augmente aussi dans le réseau. Donc, dans une durée de temps inférieur on a un nombre d'évènements plus grand.

Nous pouvons dire en conclusion, que notre solution prolonge la durée de vie du réseau par rapport aux deux autres. Cette prolongation permet au puits de recevoir plus de données. Dans les applications des réseaux de capteurs les données sont d'une grande importance. Par exemple, dans une application de détection d'objets ou de matériel, la prolongation de la durée de vie du réseau aide à augmenter la probabilité de détection et donc améliore la qualité de service. L'application du clustering passif efficace en consommation d'énergie tel que nous l'avons proposé réalise un équilibre dans la consommation d'énergie. Cet équilibre est traduit par la prolongation de la durée de vie des nœuds et en conséquence de tout le réseau.

### 5.6.3 Une quantité d'énergie variable

Nous avons remarqué que dans toutes les expériences faites précédemment, l'application du clustering (PC ou AEPC) donne de meilleurs résultats par rapport à la diffusion dirigée. Dans ce paragraphe nous proposons de comparer le clustering passif et notre solution quand elle est appliquée avec la diffusion dirigée. Ceci est fait dans le but de confirmer les résultats déjà obtenus dans la partie précédente.

Nous avons choisi un scénario dans lequel la quantité d'énergie consommée par évènement est proche (presque égale) dans les deux solutions. Ce scénario est le cas de topologie fixe avec un nombre de nœuds égal à 100. La quantité d'énergie consommée par évènement dans notre solution est 0.0320 (joule/nœuds/évènement), celle obtenue avec le clustering passif est égale à 0.0327 (joule/nœuds/évènement). Nous avons attribué des valeurs d'énergie différentes (de 5 à 30 joules ; en ajoutant à chaque fois 5 joules). Les métriques évaluées sont le taux de livraison, le temps de partitionnement et le nombre d'évènements avant le partitionnement du réseau

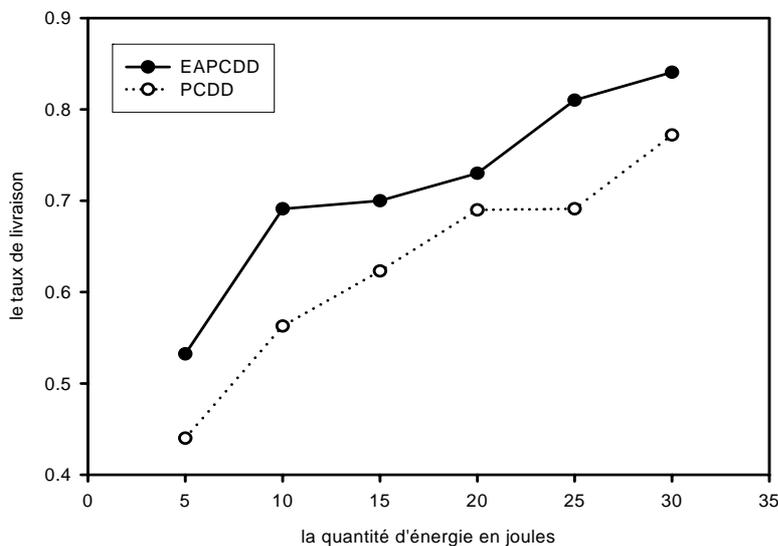


Figure 5.15 : Le taux de livraison avec une énergie variable

A partir des trois figures (5.15, 5.16, 5.17), nous pouvons dire que malgré que la quantité d'énergie moyenne consommée par événement est proche (pour ce scénario) pour les deux solutions, avec sa diminution notre solution donne de meilleurs résultats en terme de taux de livraison, de temps de partitionnement et de nombre d'évènements. Ceci confirme que l'application de notre solution réalise un équilibre dans la consommation d'énergie des différents nœuds menant à une prolongation de la durée de vie du réseau. Assurant, ainsi la livraison d'un plus grand nombre d'évènements.

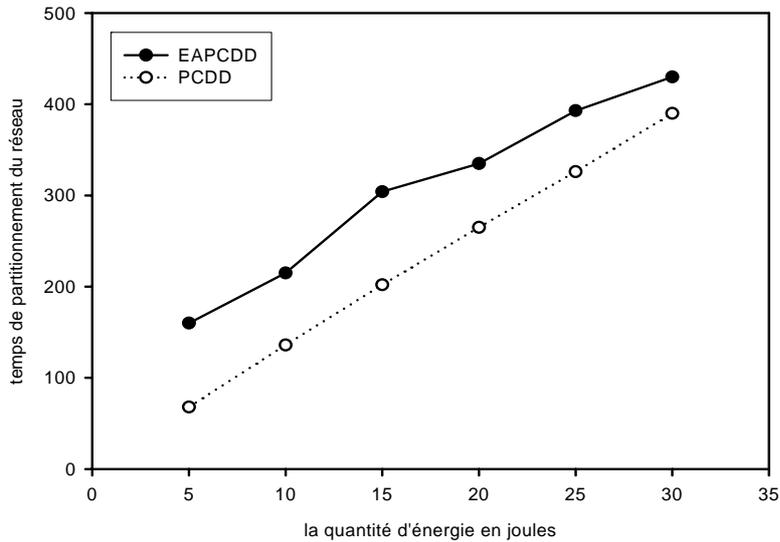


Figure 5.16 : Le temps de partitionnement du réseau avec une énergie variable

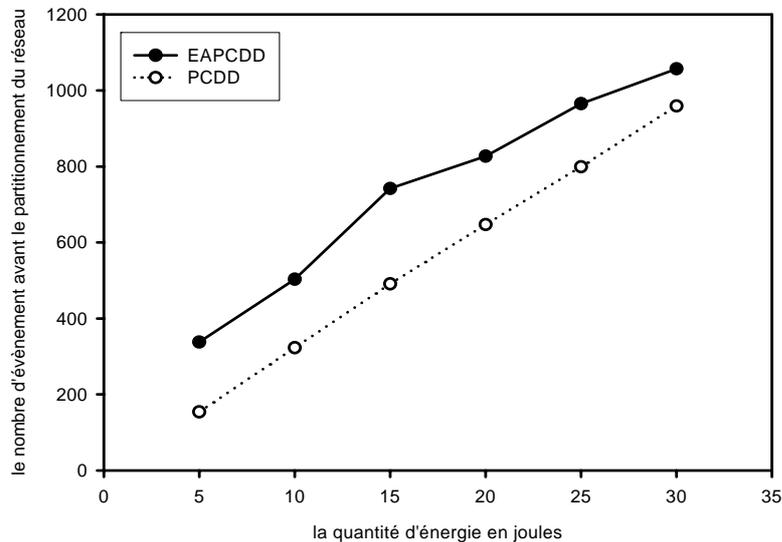


Figure 5.17 : Le nombre d'évènements avant le partitionnement du réseau

## 5.7 Conclusion

Dans ce chapitre nous avons mené une évaluation de performances de notre solution au moyen de simulation, en utilisant le simulateur NS2 version 2.27. L'application du

clustering avec la diffusion dirigée permet de mettre en œuvre une topologie plus réduite du réseau minimisant ainsi le nombre de transmissions inutiles. Ceci mène à réduire la quantité d'énergie moyenne consommée dans le réseau. Aussi, la prise en compte des états énergétiques des différents nœuds a permis, selon les résultats de simulation présentés, de répartir la consommation d'énergie au niveau des nœuds dans le réseau.

Les résultats présentés dans ce chapitre montrent l'efficacité de notre solution en termes de durée de vie du réseau et du nombre d'évènements reçus. Ces améliorations ne sont pas faites au prix des performances du paradigme. Notamment, les délais de transmissions et les taux de livraison.

# Conclusion

Le problème de la conservation d'énergie est un problème largement étudié dans les réseaux mobiles et sans fil. Cependant, avec l'arrivée des réseaux de capteurs, le problème s'est accentué par les contraintes matérielles sévères qui caractérisent les nœuds capteurs et la nature de leur déploiement.

Dans cette thèse, nous nous sommes intéressés au problème de la conservation d'énergie dans les réseaux de capteurs. Particulièrement, dans un paradigme de routage appelé la diffusion dirigée.

Dans la première partie de notre travail, nous nous sommes focalisés sur l'étude du paradigme de la diffusion dirigée, les caractéristiques de dissipation d'énergie dans les réseaux de capteurs et les niveaux de conservation d'énergie envisagés. Nous pouvons dire que le paradigme de la diffusion dirigée réalise une efficacité en consommation d'énergie dans ses étapes de fonctionnement. Cependant, son recours aux mécanismes d'inondation, en l'absence des systèmes d'adressage global dans ce type de réseau, pénalise cette efficacité. Aussi, nous avons constaté que les caractéristiques de dissipation d'énergie diffèrent d'un nœud à un autre selon le type et les fonctionnalités prises en charge. Au niveau d'un seul nœud, la quantité d'énergie consommée dépend de la tâche en cours (captage, traitement ou communication) et du mode de fonctionnement des différents composants. En conséquence, les travaux visant à réaliser une conservation d'énergie se placent à différents niveaux de l'architecture. Ceci peut être fait au niveau des nœuds individuels ou bien au niveau de tout le réseau. Dans ce dernier cas, les solutions proposées visent en particulier la phase de communication.

La diffusion dirigée étant un paradigme de dissémination de données qui assure la communication des données entre les nœuds du réseau, nous nous sommes orientés, après ce constat, vers l'étude de la conservation d'énergie dans le paradigme lui-même. Nous pouvons conclure que les travaux menés varient selon les étapes du paradigme visées et donc les objectifs recherchés. Les solutions diffèrent aussi par le type d'applications à réaliser.

Dans ce cadre, nous avons proposé une amélioration pour le paradigme de la diffusion dirigée en vue de le rendre plus efficace en consommation d'énergie. Notre proposition consiste en l'amélioration d'un algorithme de clustering afin de prendre en compte les états énergétiques des nœuds lors de la formation et la maintenance de la topologie. Ensuite appliquer cet algorithme avec la diffusion dirigée dans le but de prolonger la durée de vie des nœuds et ainsi du réseau entier. L'utilisation d'un algorithme de clustering diminue la topologie communicante et réduit, de la sorte, la quantité d'énergie globale consommée dans le réseau. Aussi, la prise en compte des états des batteries des nœuds communicants permet de répartir la consommation d'énergie dans le réseau et donc prolonge sa durée de vie. Une évaluation de performances de la solution proposée, au moyen de simulation, a aussi été donnée dans le but de confirmer ces suggestions.

Les principales caractéristiques de notre contribution sont :

- Notre solution préserve le paradigme de la diffusion dirigée tel que proposé par ses auteurs, ainsi, il reste applicable dans toutes les applications pour lesquelles il a été conçu.

- En plus du problème de la conservation d'énergie, notre contribution répond aussi au problème de scalabilité du fait qu'elle utilise une technique de clustering.
- La solution proposée réalise une prolongation de la durée de vie du réseau allant de 7% à 39,5% par rapport à la diffusion dirigée.
- L'efficacité en consommation d'énergie réalisée n'est pas faite au prix des performances du paradigme telles que le délai de transmission et le taux de livraison. Au contraire, dans la plupart des cas, les valeurs obtenues pour ces métriques avec notre solution sont meilleures.
- La solution proposée est assez ouverte et peut être appliquée avec d'autres protocoles de routage qui utilisent, dans leur fonctionnement, le mécanisme d'inondation des messages.

Ce travail nous a permis de proposer une solution efficace en consommation d'énergie dans les réseaux de capteurs. Cependant, nous pensons qu'il nous a surtout ouvert d'autres voix de recherche :

Nos résultats ont été établis sur un seul algorithme de diffusion à savoir le 'Pull à deux phases'. Nous envisageons dans le futur proche d'utiliser les autres algorithmes définis dans la diffusion dirigée et tester ainsi différents types d'applications.

La définition des paramètres de notre algorithme de clustering a été faite d'une manière empirique et leur validation a été réalisée au moyen de simulation. Cependant, une étude analytique formelle serait nécessaire pour définir ces valeurs.

Une continuité de ce travail serait de tester les résultats obtenus avec des protocoles de routage dans les réseaux Ad hoc. Ceci ne constitue pas une perspective dans le but d'améliorer la solution mais aide à la valider dans un autre environnement.

# Bibliographie

- [1] : Scott Corson, Joseph Macker. RFC 2501, Mobile Ad hoc Networking (MANET). IETF (1999)
- [2] : T. Lemlouma, N. Badache, "le routage dans les réseaux Ad hoc", rapport de mini projet, USTHB. Septembre 2000.
- [3] : J-M. Percher, B Jouga, ' Détection d'intrusions dans les réseaux ad hoc' apparu en proceeding du Symposium sur la Sécurité des Technologies de l'Information et de la Communication. 2003.
- [4] : I. F. Akyildiz et al., 'Wireless sensor networks: a survey', Computer Networks, Vol. 38, pp. 393-422, march 2002
- [5]: Wireless Integrated Network Sensors, University of California, Los Angeles. Available: <http://www.janet.ucla.edu/WINS>
- [6]: J.M. Kahn, R.H. Katz, and K.S.J. Pister, 'Next century challenges: Mobile networking for smart dust', in Proc. Mobicom, 1999, pp. 483-492
- [7]: D. Culler, E. Brewer, D. Wagner, 'A Platform for WEbS (wireless embedded sensor actuator) systems', Technical Report, University of California, Berkeley, 2001
- [8]: Jan Rabaey, et.al., 'PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking', IEEE Computer, pp. 42-48, July 2000
- [9]: EYES Project Website. <http://www.eyes.eu.org>
- [10] : V, Handziski, A, Kopke, Holger Karl, A, Wolisz. 'A common wireless sensor network architecture' appeared in proc 1. GI/ITG Fachgesprach "Sensornetze" (Teknical Report TKN-03-012 of the Telecommunications Networks Group, Technische Universitat Brln), pp. 10-17, Berlin, July 2003.
- [11] : E. Welsh, W. Fish, P. Frantz, 'GNOMES: A Testbed for Low-Power Heterogeneous Wireless Sensor Networks', IEEE International Symposium on Circuits and Systems (ISCAS), Bangkok, Thailand, 2003. <http://citeseer.ist.psu.edu/welsh03gnomes.html>
- [12] A. Chandrakasan, Etal, 'low power wireless sensor network' in VLSI design; 2001

- [13]: Vijay Raghunathan, Curt Schurgers, Sung Park, and Mani B. Srivastava; 'Energy-Aware Wireless Microsensor Networks'. IEEE Signal Processing Magazine 1053-5888/02/, March 2002
- [14]: S. Park, A. Savvides, and M. Srivastava, 'Battery capacity measurement and analysis using lithium coin cell battery', in Proc. ISLPED, 2001, pp. 382-387.
- [15]: G. Amato, A. Caruso, S. Chessa, V. Masi and A. Urpi, 'State of the Art and future Directions in Wireless Sensor Network's Data Management', Project: Management of Data in Wireless Sensor Networks (MaD-WiSe), *Deliverable 1*
- [16]: D. Culler, 'Tiny OS – a component-based OS for the networked sensor regime', URL: <http://webs.cs.berkeley.edu/tos/>, 2003.
- [17]: C. Y Chong, S. P Kumar, 'Sensor Networks : Evolution, Opportunities and Challenges' in proceeding of the IEEE, Vol. 91, N° 8, August 2003.
- [18]: Fabio Silva, John Heidemann, Ramesh Govindan, Deborah Estrin, 'An Overview of Directed Diffusion' (2004)
- [19]: W. Heinzelman, J. Kulik, and H. Balakrishnan, 'Adaptive Protocols for Information Dissemination in Wireless Sensor Networks', Proc. 5th ACM/IEEE Mobicom Conference (MobiCom '99), Seattle, WA, August, 1999. pp. 174-85.
- [20]: C. Intanagonwiwat, R. Govindan, and D. Estrin, 'Directed diffusion: a scalable and Robust communication paradigm for sensor networks', Proceedings of ACM MobiCom '00, Boston, MA, 2000, pp. 56-67.
- [21]: C. Schurgers and M.B. Srivastava, 'Energy efficient routing in wireless sensor networks", in the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.
- [22]: D. Braginsky and D. Estrin, 'Rumor Routing Algorithm for Sensor Networks', in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, October 2002.
- [23] A. Manjeshwar and D. P. Agarwal, 'TEEN: a routing protocol for enhanced efficiency in wireless sensor networks', In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001.
- [24]: Y. Yao and J. Gehrke, 'The cougar approach to in-network query processing in sensor networks', in SIGMOD Record, September 2002.
- [25]: Y. Yu, D. Estrin, and R. Govindan, 'Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks', UCLA Computer Science Department Technical Report, UCLA-

- [26]: Y. Xu, J. Heidemann, D. Estrin, 'Geography-informed Energy Conservation for Ad-hoc Routing', In Proceedings of the Seventh Annual ACM/IEEE International Conference on Mobile Computing and Networking 2001, pp. 70-84
- [27]: V. Rodoplu and T.H. Ming, 'Minimum energy mobile wireless networks', IEEE Journal of Selected Areas in Communications, Vol. 17, No. 8, pp. 1333-1344, 1999.
- [28]: W. Heinzelman, A. Chandrakasan et H. Balakrishnan, 'Energy-efficient Communication Protocol for Wireless Microsensor Networks', Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS '00), January 2000.
- [29]: S. Lindsey, C. Raghavendra, 'PEGASIS: Power-Efficient Gathering in Sensor Information Systems', IEEE Aerospace Conference Proceedings, 2002, Vol. 3, 9-16 pp. 1125-1130.
- [30]: L. Subramanian and R. H. Katz, 'An Architecture for Building Self Configurable Systems', in the Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, Boston, MA, August 2000.
- [31]: K. Akkaya and M. Younis, 'An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks', in the Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN2003), Providence, Rhode Island, May 2003.
- [32]: T. He et al., 'SPEED: A stateless protocol for real-time communication in sensor networks', in the Proceedings of International Conference on Distributed Computing Systems, Providence, RI, May 2003.
- [33]: D. Estrin, R. Govindan, J. Heidemann, S. Kumar, 'Next Century Challenges: Scalable Coordination in Sensor Networks', Mobicom '99 Seattle Washington USA Copyright ACM 1999.
- [34]: R. Di Pietro, L V. Mancini, Y Law, S Etalle, et P Havinga, 'LKHW: A Directed Diffusion-Based Secure Multicast Scheme for Wireless Sensor Networks', in proceedings of 2003 International Conference on Parallel Processing Workshops, 2003. October 2003; pp 397-406
- [35]: D. Coffin, D. V. Hook, R Govindan, John Heidemann, Fabio Silva, 'Network Routing Application Programmer's Interface (API) and Walk Through 8.0' Technical Report 01-741, USC/ISI, March 2001.
- [36]: J, Heidemann, F. Silva, D Estrin, 'Matching Data Dissemination Algorithms to Application Requirements', Technical Report ISI-TR-571, USC/Information Sciences Institute, April 2003.

- [37]: L. Khelladi, N. Badache 'le service de routage dans les réseaux de capteurs', Thèse de Magister USTHB, Novembre 2005
- [38]: V. Tiwari, S. Malik, A. Wolfe, and M.T.C. Lee, 'Instruction level power analysis and optimization of software', J. VLSI Signal Processing, vol. 13, no. 2, pp. 1-18, 1996.
- [39]: A. Sinha and A.P. Chandrakasan, 'Jouletrack: A web based tool for software energy profiling', in Proc. Design Automation Conf., 2001, pp. 220-225.
- [40]: Y. Xu, J. Heidemann, and D. Estrin, 'Geography-informed energy conservation for ad hoc routing', in Proc. Mobicom, 2001, pp. 70-84.
- [41]: A. Wang, S-H. Cho, C.G. Sodini, and A.P. Chandrakasan, 'Energy-efficient modulation and MAC for asymmetric microsensor systems', in Proc. ISLPED, 2001, pp 106-111.
- [42]: WINS project, Rockwell Science Center. Available: <http://wins.rsc.rockwell.com>
- [43]: C.F. Chiasserini and R.R. Rao, 'Pulsed battery discharge in communication Devices', in Proc. Mobicom, 1999, pp. 88-95.
- [44]: S. Park, A. Savvides, and M. Srivastava, 'Battery capacity measurement and analysis using lithium coin cell battery', in Proc. ISLPED, 2001, pp. 382-387.
- [45]: L. Benini and G. DeMicheli, 'Dynamic Power Management: Design Techniques and CAD Tools', Norwell, MA: Kluwer, 1997
- [46]: T.A. Pering, T.D. Burd, and R.W. Brodersen, 'The simulation and evaluation of dynamic voltage scaling algorithms', in Proc. ISLPED, 1998, pp 76-81.
- [47]: V. Raghunathan, P. Spanos, and M. Srivastava, 'Adaptive power-fidelity in energy aware wireless embedded systems', presented at IEEE Real Time Systems Symposium, 2001.
- [48]: F. Yao, A. Demers, and S. Shenker, 'A scheduling model for reduced CPU energy', in Proc. Annu. Symp. Foundations of Computer Science, 1995, pp. 374-382.
- [49]: V. Tsiatsis, S. Zimbeck, and M. Srivastava, 'Architectural strategies for energy efficient packet forwarding in wireless sensor networks', in Proc. ISLPED, 2001, pp. 92-95.
- [50]: C. Schurgers, O. Aberthorne, and M. Srivastava, 'Modulation scaling for energy aware communication systems', in Proc. ISLPED, 2001, pp. 96-99

- [51]: K. Sohrabi, J. Gao, V. Ailawadhi, and G. Pottie, 'Protocols for self-organization of a wireless sensor network', in IEEE Pers. Commun. Mag., vol. 7, pp. 16-27, Oct. 2000.
- [52]: B.Noble, M. Satyanarayanan, 'Experience with adaptive mobile applications in Odyssey', Mobile Networks and Applications Vol. 4, 1999
- [53] John Heidemann and Wei Ye, 'Energy Conservation in Sensor Networks at the Link and Network Layers', in USC/ISI TECHNICAL REPORT ISI-TR-599, November 2004
- [54]: Ram Ramanathan and Regina Rosales-Hain, 'Topology control of multihop wireless networks using transmit power adjustment', In Proceedings of the IEEE Infocom, Tel Aviv, Israel, Mar. 2000, p. xxx, IEEE.
- [55]: The Network Simulator - ns-2, 'The *ns* Manual', The VINT Project, collaboratoin between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, Kevin Fall and Kannan Varadhan Editors, 2003, <http://www.isi.edu/nsnam/ns>.
- [56]: Jeffrey Monks, Vaduvur Bharghavan, and Wen-Mei Hwu, 'A power controlled multiple access protocol for wireless packet networks', In Proceedings of the IEEE Infocom, Anchorage, Alaska, USA, Apr. 2001, p. to appear, IEEE.
- [57]: Piyush Gupta and P. R. Kumar, 'The capacity of wireless networks', IEEE Transactions on Information Theory, vol. IT-46, no. 2, pp. 388. 404, Mar. 2000.
- [58]: OPNET Technologies, 'OPNET Modeler accelating networks R&D', 2004, [www.opnet.com](http://www.opnet.com)
- [59]: Gregory J. Pottie and William J. Kaiser, 'Wireless integrated network sensors', Communications of the ACM, vol. 43, no. 5, pp. 51.58, May 2000.
- [60]: Dongjin Son, Bhaskar Krishnamachari, and John Heidemann, 'Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks', in Proceedings of the First IEEE Conference on Sensor and Adhoc Communication and Networks, Santa Clara, California, USA, Oct. 2004, IEEE.
- [61]: 'LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification', IEEE, New York, NY, USA, IEEE Std 802.11-1999 edition, 1999.
- [62]: Yu-Chee Tseng, Chih-Shun Hsu, and Ten-Yueng Hsieh, 'Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks', In Proceedings of the IEEE Infocom, New York, NY, June 2002, pp. 200. 209.

- [63]: S. Singh and C.S. Raghavendra, 'PAMAS: Power aware multi-access protocol with signalling for ad hoc networks', *ACM Computer Communication Review*, vol. 28, no. 3, pp. 5.26, July 1998.
- [64]: Wei Ye, John Heidemann, and Deborah Estrin, 'An energy-efficient MAC protocol for wireless sensor networks', in *Proceedings of the IEEE Infocom*, New York, NY, USA, June 2002, USC/Information Sciences Institute, pp. 1567.1576, IEEE.
- [65]: Wei Ye, John Heidemann, and Deborah Estrin, 'Medium access control with coordinated, adaptive sleeping for wireless sensor networks', *IEEE/ACM Transactions on Networking*, vol. 12, no. 3, pp. 493.506, June 2004, A preprint of this paper was available as ISI-TR-2003-567.
- [66]: Tijs van Dam and Koen Langendoen, 'An adaptive energy-efficient mac protocol for wireless sensor networks', in *Proceedings of the First ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, Los Angeles, California, USA, Nov. 2003, pp. 171. 180, ACM.
- [67]: Curt Schurgers, Vlasios Tsiatsis, Saurabh Ganeriwal, and Mani Srivastava, 'Optimizing sensor networks in the energy-latency-density space', *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70.80, 2002.
- [68]: Jason L. Hill and David E. Culler, 'Mica: A wireless platform for deeply embedded networks', *IEEE Micro*, vol. 22, no. 6, pp. 12.24, Nov/Dec 2002.
- [69]: Ya Xu, John Heidemann, and Deborah Estrin, 'Geography-informed energy conservation for ad hoc routing', in *Proceedings of the ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, pp. 70.84, ACM.
- [70]: Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, 'Span: an efficient coordination algorithm for topology maintenance in ad hoc wireless networks', in *Proceedings of the ACM International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001, p. xxx, ACM.
- [71]: Ya Xu, Solomon Bien, Yutaka Mori, John Heidemann, and Deborah Estrin, 'Topology control protocols to conserve energy in wireless ad hoc networks', Tech. Rep. 6, University of California, Los Angeles, Center for Embedded Networked Sensing, Jan. 2003.
- [72]: D. Djenouri; N. Badache, 'Optimisation de la consommation d'énergie pour le routage dans les réseaux mobiles ad hoc', Thèse de Magister, USTHB, Juillet 2003
- [73]: Suresh Singh, Mike Woo, and C. S. Raghavendra, 'Power-aware routing in mobile ad hoc networks', in *Proceedings of the ACM International Conference on Mobile Computing and Networking*, Dallas, Texas, USA, Oct. 1998, pp. 181.190, ACM.

- [74]: Jae-Hwan Chang and Leandros Tassiulas, 'Energy conserving routing in wireless ad-hoc networks', in Proceedings of the IEEE Infocom, Tel-Aviv, Israel, Mar. 2000, IEEE.
- [75]: Charles E. Perkins and Pravin Bhagwat, 'Highly dynamic destination sequenced distance-vector routing (DSDV) for mobile computers', In Proceedings of the ACM SIGCOMM Conference, London, UK, Aug.
- [76]: David B. Johnson and David A. Maltz, 'Dynamic source routing in ad hoc wireless networks', in Mobile Computing, Tomasz Imielinski and Hank Korth, Eds. Kluwer Academic Publishers, 1996.
- [77]: Suman Banerjee and Archan Misra, 'Minimum energy paths for reliable communication in multi-hop wireless networks', in Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing, Lausanne, Switzerland, June 2002, pp. 146.156, ACM.
- [78]: Alec Woo, Terence Tong, and David Culler, 'Taming the underlying challenges of reliable multihop routing in sensor networks', in Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, USA, Nov. 2003, pp. 14.27, ACM.
- [79]: B. Karp and H. T. Kung, 'GPSR: Greedy perimeter stateless routing for wireless sensor networks', in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '00), Boston, MA, August 2000.
- [80]: R. Shah and J. Rabaey, 'Energy Aware Routing for Low Energy Ad Hoc Sensor Networks', in the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.
- [81]: Kwon, T.J., Gerla, M, 'Efficient flooding with passive clustering (PC) in ad hoc networks', ACM SIGCOMM Computer Communication Review 32 (2002) 44–56
- [82]: V. Handziski, A. Koepke, H Karl, C Frank, W Drytkiewicz, 'Improving the Energy Efficiency of Directed Diffusion Using Passive Clustering', in First European Workshop, EWSN 2004 Berlin, Germany, January 19-21, 2004 Proceedings.
- [83]: L. Tzevelekas, A. Ziviani, MD De Amorim, P Todorova, I Stavrakakis, 'Towards potential-based clustering for wireless sensor networks', in Proceedings of the 2005 ACM conference on Emerging network experiment and technology.
- [84]: S. Banerjee and S. Khuller, 'A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks,' in Proceedings of IEEE INFOCOM, April 2001.

- [85]: M. Gerla, T. J. Kwon, and G. Pei, 'On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering,' in Proceeding of WCNC, 2000.
- [86]: Yunjung Yi, Taek Jin Kwon, Mario Gerla, 'Passive Clustering in Ad Hoc Networks (PC)', IETF Internet Draft; 2001
- [87]: Md. Mamun-Or-Rashid, Choong Seon Hong, Chi-Hyung In, 'Passive Cluster Based Clock Synchronization in Sensor Network,' (AICT/SAPIR/ELETE'05), pp. 340-345; 2005
- [88]: Guangyu Pei; Chien, C, 'Low power TDMA in large wireless sensor networks', In Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations. IEEE; Volume 1, Issue , 2001 Page(s): 347 - 351 vol.1
- [89]: GLOMOSIM ( Global Mobile Information Systems Simulation Library), UCLA Paralle Computing Laboratory, <http://pcl.cs.ucla.edu/projects/glomosim/>, 2001.
- [90]: P.P.Pham, 'Comprehensive Analysis of IEEE 802.11', Mobile Networks and Applications, Vol 10, page 691-703, 2005.
- [91]: I-LENSE Topology Generator (topo\_gen), available at: [http://www.isi.edu/ilense/software/topo\\_gen/topo\\_gen.html](http://www.isi.edu/ilense/software/topo_gen/topo_gen.html)
- [92]: D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. 'Highly- Resilient, Energy-Efficient Multi-path Routing in Wireless Sensor Networks', Mobile Computing and Communications Review, Vol. 4, No. 5, October 2001.
- [93]: Lawrence A. Bush, Christopher D. Carothers and Boleslaw K. Szymanski, 'Algorithm for optimizing energy use and path resilience in sensor networks', in: Wireless Sensor Networks, 2005. Proceedings of the Second European Workshop; p.p: 391- 396; 2005
- [94]: Jochen Schiller, Achim Liers, Hartmut Ritter, Rolf Winter, Thiemo Voigt, 'ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing', hicss, p. 286c, Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9, 2005.
- [95]: Jonathan Pollet, et al., 'ATEMU: A Fine-Grained Sensor Network Simulator', in Proceedings of SECON'04, First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004.
- [96]: David Curren, 'A survey of Simulation in Sensor Networks', available at [www.cs.binghamton.edu/kang/cs580s/david.pdf](http://www.cs.binghamton.edu/kang/cs580s/david.pdf)