

Liste des Acronymes

WSN	Wireless Sensor Network
RCSF	Réseau de Capteurs Sans Fil
MEMS	Micro-Electro-Mechanical Systems
GPS	Global Positioning System
LMEER	Layered Multi-hop Energy Efficient Clustering Routing Protocol for Wireless Sensor Networks
ADC	Analog to Digital Converter
7RF	Radio Frequency
IEEE	Institute of Electrical and Electronics Engineers
WECA	Wireless Ethernet Compatibility Alliance
PAN	Personal Area Network
Wi-Fi	WIreless-Fidelity
OS	Operating System
MAC	Media Access Control
MIPS	Millions D'Instructions Par Seconde
RTS	Request To Send
CTS	Clear To Send
DVS	Dynamic Voltage Scaling
FDMA	Frequency Division Multiple Access
TDMA	Time Devision Multiple Access
CSMA	Carrier Sense Multiple Access
CDMA	Code Division Multiple Access
SMACS	Self-organizing Medium Access Control for Sensor Networks
EAR	Eavesdrop And Register
B-MAC	Berkeley Media Access Control
S-MAC	Sensor Media Access Control
T-MAC	Timeout Media Access Control
DD	Directe Diffusion
PEBECS	Partition Energy Balanced and Efficient Clustering Scheme
GAF	Geographic Adaptative Fidelity

SAR	Sequential Assignment Routing
SPEED	A Stateless Protocol for Real-Time Communication in Sensor Networks
SPIN	Sensor Protocol for Information Negotiation
EAR	Energy Aware Routing for Low Energy Ad Hoc Sensor Networks
QoS	Quality Of Service
LEACH	Low-Energy Adaptive Clustering Hierarchy
LEACH-C	Low-Energy Adaptive Clustering Hierarchy Centralized
PEGASIS	Power-Efficient GAthering in Sensor Information Systems
TL-LEACH	Two Level hierarchy Low-Energy Adaptive Clustering Hierarchy
HEED	A Hybrid Energy-Efficient Distributed clustering approach for ad-hoc sensor networks
EECS	Energy Efficient Clustering Scheme in WSN
EEUC	An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks
FLOC	Fast Clustering Service
EEHC	Energy Efficient Hierarchical Clustering
RCC	Random Competition based Clustering
CLUBS	ALgorithm for Group Formation In An Amorphous Computer
LCA	Linked Cluster Algorithm
WCA	Weighted Clustering Algorithm
DCA	Distributed Clustering Algorithm
ACE	An Emergent Algorithm for Highly Uniform Cluster Formation
OTCL	Object Tool Command Language
NS-2	Network Simulator 2
GloMoSim	Global Mobile Simulator
TOSSIM	TinyOS SIMulator
OMNET ++	Objective Modular NETwork Test-bed in C++
J-Sim	Java-SIMultaor
ACA	Autonomous Component Architecture
Jacl	JAVA tCL

Tables des matières

Tables des matières

Liste des Figures	vii
Liste des tableaux	x
Introduction générale	1
Chapitre 1 Généralités sur les réseaux de capteurs sans fil	3
1.1 Introduction	3
1.2 Le capteur	3
1.2.1 Définition d'un capteur	4
1.2.2 Architecture d'un capteur	4
1.2.2.1 Matériel	4
1.2.2.2 Système d'exploitation	6
1.2.3 Caractéristiques d'un capteur	6
1.3 Les réseaux de capteurs	7
1.3.1 Définition d'un réseau de capteurs	7
1.3.2 Architecture du réseau	8
1.3.2.1 Topologies du réseau	9
1.3.3 Caractéristiques du réseau de capteurs	10
1.4 Domaines d'applications	12
1.4.1 Applications militaires	12
1.4.2 Applications environnementales	13
1.4.3 Applications à la sécurité	13
1.4.4 Applications médicales	13
1.4.5 La domotique	13
1.5 La communication dans les réseaux de capteurs	14
1.5.1 Pile protocolaire (modèle en couches)	14
1.5.2 Technologies et standards de communication	16
1.6 Facteurs et contraintes influant la conception du réseau	18
1.6.1 Tolérance aux fautes (pannes)	18
1.6.2 Passage à l'échelle (scalabilité)	18
1.6.3 Topologie dynamique	18
1.6.4 Coût de production	19
1.6.5 L'environnement	19
1.6.6 Qualité de service	19
1.6.7 Contraintes matérielles	19
1.6.8 Les médias de transmission	19
1.6.9 Consommation d'énergie	20
1.7 Différentes problématiques et axes de recherche dans les réseaux de capteurs	20
1.8 Conclusion	21

Chapitre 2 Routage à basse consommation d'énergie	22
2.1 Introduction	22
2.2 Energie	22
2.2.1 Consommation d'énergie dans les réseaux de capteurs	23
2.2.2 Facteurs intervenant dans la consommation d'énergie	23
2.2.3 Taxonomies d'approches de conservation d'énergie dans les réseaux de capteurs sans fil	25
2.3 Le routage dans les réseaux de capteurs	29
2.3.1 Contraintes à tenir en compte lors de la conception des protocoles de routage.....	29
2.3.2 Classification des protocoles de routage.....	30
2.3.2.1 Routage basé sur la structure du réseau	31
2.3.2.2 Routage basé sur le fonctionnement des protocoles	39
2.4 Conclusion	45
Chapitre 3 État de L'art sur Les Protocoles de Clustering Pour Les Réseaux De Capteurs Sans fil	46
3.1 Introduction	46
3.2 Notions de base et motivations	46
3.2.1 Définitions	47
3.2.2 Formation de clusters	48
3.2.3 Avantages e l'approche clustering	49
3.3 Contraintes et facteurs de conception lors du processus de clustering	49
3.4 Techniques de clustering	51
3.4.1 Architecture du réseau et le modèle de fonctionnement	51
3.4.1.1 Traitement de données dans le réseau.....	51
3.4.1.2 Mobilité du réseau.....	52
3.4.1.3 Mode de déploiement et capacité d'un nœud	52
3.4.2 Objectifs du clustering	53
3.4.2.1 Minimiser la consommation d'énergie	53
3.4.2.2 Équilibrage de la charge 'Laod balancing'	53
3.4.2.3 Tolérance aux pannes	53
3.4.3 Taxonomie d'attributs de clustering	54
3.4.3.1 Propriétés des clusters	54
3.4.3.2 Capacités d'un cluster-head	55
3.4.3.3 Le processus de clustering	55
3.4.4 Paramètres de décision	56
3.4.4.1 Décision basée sur le temps de convergence	56
3.4.4.2 Décision basée sur les heuristiques	56
3.4.4.3 Décision basée sur le poids	57
3.4.4.4 Décision basée sur l'émergence	57
3.5 Approches de clustering pour les réseaux de capteurs sans fil	57
3.5.1 Les protocoles hiérarchiques à temps de convergence constant	58
3.5.1.1 LEACH: Low-Energy Adaptive Clustering Hierarchy	58
3.5.1.2 HEED: A Hybrid Energy Efficient Distributed Clustering Approach For Ad-hoc Sensor Networ	61

3.5.1.3	EECS: Energy Efficient Clustering Scheme in WSN	63
3.5.1.4	EEUC: An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks	64
3.5.2	Les protocoles hiérarchique a temps de convergence variable	65
3.5.2.1	EEHC : Energy Efficient Hierarchical Clustering	65
3.5.2.2	RCC : Random Competition based Clustering	66
3.5.2.3	CLUBS : Algorithm For Group Formation In An Amorphous Computer	67
3.5.2.4	Hierarchical Control Clustering	68
3.5.3	Les protocoles heuristique	69
3.5.3.1	LCA2 : Linked Cluster Algorithm	69
3.5.3.2	Highest Connectivity Cluster Algorithm	70
3.5.3.3	Max-Min-D Clustering	70
3.5.4	Les protocoles pondérés	71
3.5.4.1	WCA : Weighted Clustering Algorithm	71
3.5.4.2	DCA : Distributed Clustering Algorithm	72
3.5.5	Les protocoles émergents	73
3.5.5.1	ACE: An Emergent Algorithm for Highly Uniform Cluster Formation.....	73
3.6	Tableaux comparatifs pour les protocoles de clustering.	74
3.7	Conclusion	80
Chapitre 4	LMEER: Un protocole de clustering à basse consommation d'énergie pour les réseaux de capteurs sans fil	81
4.1	Introduction	81
4.2	Motivations	81
4.3	Principe et fonctionnement	82
4.3.1	Configuration du réseau	83
4.3.2	Formation des clusters et communication des données	86
4.3.3	La maintenance des clusters.....	92
4.4	Simulations et résultats	93
4.4.1	Environnements de simulation	93
4.4.2	Choix de l'environnement de simulation	96
4.4.3	Les paramètres de simulation	97
4.4.4	Métriques d'évaluation de performances	98
4.4.5	Discussion des résultats	99
4.5	Conclusion	108
Conclusion et Perspectives		109
Bibliographie		111

Liste des figures

Fig.1.1: Processus De Capture	4
Fig.1.2: Architecture D'un Nœud Capteur	5
Fig.1.3: Exemples de noeuds capteurs (Berkeley Mote)	6
Fig.1.4: Exemple De Réseau De Capteurs	8
Fig.1.5: Architecture d'un réseau de capteurs	9
Fig.1.6: La topologie en étoile	9
Fig.1.7: La topologie maillée.....	9
Fig.1.8: La topologie hybride	10
Fig.1.9: La vision d'un capteur Dépend du rayon de réception	12
Fig.1.10: Modèle en couches (pile protocolaire) d'une architecture de capteurs [3]	14
Fig.2.1: Techniques de conservation d'énergie	26
Fig.2.2: Classification de protocoles de rgie	31
Fig.2.3: Exemple d'une grille virtuelle dans GAF.....	38
Fig.3.1: Hiérarchie d'un réseau de capteurs sans fil [42]	48
Fig.3.2: Taxonomie d'attributs de clustering dans un réseau de capteurs	54
Fig.3.3: Le mécanisme utilisé par EEUC	64
Fig.3.4: Hiérarchie de groupes.....	68
Fig.4.1: Configuration du réseau En se basant sur le nombre de sauts	84
Fig.4.2: Configuration du réseau En se basant sur la distance	85
Fig.4.3: Modèle de consommation d'énergie	97
Fig.4.4: Moyenne d'énergie consommée : LMEER-D vs LMEER-S vs LEACH	100
Fig.4.5: Taux moyen de livraison de paquets LMEER-D vs LMEER-S vs LEACH	101
Fig.4.6: Délai moyen de bout en bout de LMEER-D vs LMEER-S vs LEACH	102
Fig.4.7: Taux moyen de livraison de paquets LMEER-D vs LMEER-S vs LEACH	103

Fig .4.8: Moyenne d'énergie consommée avec la présence d'un pourcentage de) 10% de nœuds défaillants (b) 30% de nœuds défaillants (c) 50% de taux de nœuds défaillants	104
Fig 4.9: Taux moyen de livraison de paquets avec la présence d'un pourcentage) 10% de nœuds défaillants (b) 30% de nœuds défaillants (c) 50% de taux de nœuds défaillants	105
Fig 4.10: Délai moyen de bout en bout avec la présence d'un pourcentage de : (a) 10% de nœuds défaillants (b) 30% de nœuds défaillants (c) 50% de taux de nœuds défaillants.....	106
Fig 4.11: La durée de vie du réseau avec la présence de :(a) 10% de taux de défaillance (b) 30% de taux de défaillance (c) 50% taux de défaillance.....	107

Liste des tableaux

Tab.1.1: Comparaison entre le réseau de capteurs et le réseau ad hoc [4,20].....	10
Tab.1.2: Comparaison entre les trois technologies IEEE802.11, Bluetooth et Zigbee.....	18
Tab.3.1: Quelques Algorithmes de Clustering pour les réseaux de capteurs.....	57
Tab.3.2: Classification des protocoles de clustering présentés selon les métriques de décision, des critères qui portent sur les objectifs à atteindre, l'architecture du réseau et modèle de fonctionnement.....	76
Tab.3.3 : Classification des protocoles de routage présentés selon les paramètres de décision et les attributs du clustering.....	78
Tab.4.1 : Paramètres de simulation	97

Introduction Générale

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

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

En effet, la petite taille des nœuds capteurs impose des limites sur leurs capacités de traitement, de stockage et surtout d'énergie ; car ils sont généralement alimentés par des batteries à capacités limitées. Recharger les batteries dans un réseau de capteurs est parfois impossible en raison de l'emplacement des noeuds, mais le plus souvent pour la simple raison que cette opération est pratiquement irréalisable. Il est donc couramment admis que la limitation énergétique est une question incontournable dans la conception des réseaux de capteurs en raison des contraintes strictes qu'elle impose sur son exploitation.

A l'heure actuelle, l'effort essentiel de la recherche porte sur les moyens de réduire au minimum la consommation d'énergie afin de maximiser la durée de vie du réseau. Ceci est à travers l'utilisation de techniques de conservation innovantes portants sur tous les niveaux de la hiérarchie du réseau : physique, MAC et routage. En effet, si nous voulons que le réseau fonctionne de manière satisfaisante aussi longtemps que possible, nous devons prendre en considération la contrainte énergétique imposée par ses capteurs. De façon générale, économiser l'énergie revient à trouver le meilleur compromis entre : la consommation énergétique des différentes tâches assurées par les capteurs, la capture, le traitement et la communication. C'est cette dernière tâche qui est la consommatrice d'énergie prééminente. De ce fait, l'objectif majeur pour ces réseaux, est de minimiser l'énergie consommée lors de la tâche de communication. L'une des solutions les plus communément employées, consiste à utiliser des protocoles de routage efficaces en termes de consommation énergétique.

Dans ce travail on s'intéresse particulièrement à la conception d'un nouveau protocole de routage répondant aux défis imposés par la ressource énergétique limitée au niveau des capteurs, et ce dans un but de maximiser leur durée de fonctionnement et par conséquent, la durée de vie du réseau en entier. Pour ce faire, nous avons étudié, dans un premier temps, les approches de routage sous contraintes énergétiques proposées dans la littérature pour les réseaux de capteurs. Ainsi, des critiques et des analyses sont apportées aux techniques utilisées dans les protocoles étudiés, et particulièrement les techniques adoptées dans les protocoles

proposés selon l'approche de clustering. Cela est afin de pouvoir, dans un second temps, concevoir un nouveau protocole de routage à basse consommation d'énergie selon l'approche de clustering. La raison en est que la structuration hiérarchique du réseau s'est présentée, dans ce contexte, comme une approche de routage qui contribue d'une manière efficace à l'économie d'énergie, mais aussi à la réduction des messages de contrôle, l'augmentation de la réutilisabilité de la bande passante et l'attribution de ressources, tout en organisant le réseau d'une manière favorable à la préservation de cette ressource épuisable et difficilement rechargeable.

Ce mémoire est organisé en quatre chapitres de la manière suivante :

Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil avec une description de leurs architectures et leurs caractéristiques principales ainsi que leurs domaines d'application. Nous spécifions par la suite le type de communication utilisé dans ces réseaux, pour discuter à la fin les facteurs et les contraintes influant leurs conceptions.

Dans le deuxième chapitre, nous présentons tout d'abord, le concept de l'énergie dans les réseaux de capteurs tout en discutant les facteurs intervenant dans la consommation de cette ressource énergétique, et les différentes approches adoptées pour sa conservation. La deuxième partie du chapitre, quant à elle, est consacrée à aborder le concept du routage sous contrainte énergétique dans les réseaux de capteurs, où nous exposons les classifications existantes pour les protocoles de routage à basse consommation d'énergie proposés pour ces réseaux et les facteurs influant leur conception, en citant quelques exemples de ces protocoles et en apportant une étude critique à chaque classe de routage.

Après l'introduction de la classe du routage hiérarchique dans le chapitre précédent, nous consacrons le troisième chapitre à l'étude détaillée de cette approche. Nous définissons en premier lieu, l'ensemble des concepts relatifs au mécanisme de clustering ainsi que les contraintes qui interviennent lors de la conception d'un protocole de routage selon cette approche. Puis, nous passons en revue les principales techniques de clustering utilisées dans la littérature en exposant un ensemble de protocoles dans chaque catégorie. Pour présenter à la fin de ce chapitre, une étude comparative entre les différentes caractéristiques des protocoles étudiés.

Le quatrième chapitre est consacré à la description détaillée du protocole de routage "LMEER" (*Layered Multi-hop Energy Efficient Routing protocol*) proposé comme solution à la problématique du routage à basse consommation d'énergie dans les réseaux de capteurs sans fil selon l'approche de clustering, tout en détaillant son principe de fonctionnement, puis analyser ses performances en le comparant avec le protocole LEACH.

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

Chapitre 1

Généralités sur les réseaux de capteurs sans fil

1.1 Introduction

Au cours de ces dernières décennies, la technologie des réseaux mobiles n'a cessé de se développer et a connu un attrait croissant pour un nouveau type de réseaux ad hoc : les réseaux de capteurs sans fil (*RCSFs*), ou Wireless Sensor Network (*WSNs*).

Ces réseaux est le résultat d'une fusion de deux pôles de l'informatique moderne : les systèmes embarqués et les communications sans fil. Ces derniers ont bénéficié des récentes avancées technologiques dans le domaine de la miniaturisation des systèmes micro-électro-mécaniques (MEMS) [3]. Ils utilisent dans leurs conceptions des capteurs à faible coût et consommation d'énergie, dont l'objectif principal est de recueillir des informations sur leur environnement immédiat, de les traiter pour les faire parvenir à un ou plusieurs points de collecte.

Ces réseaux à hautes autonomies et à infrastructures non prédefinies sont devenus des éléments incontournables dans les systèmes, où les informations issues de l'environnement extérieur sont nécessaires pour évaluer et agir. Ils couvrent une large étendue d'applications et apportent une performance irréprochable dans les domaines médicaux, militaires, environnementaux, sécurité, fabrication, télécommunications et robotique.

Dans ce chapitre sur les généralités des capteurs et les réseaux de capteurs sans fil seront présentées les définitions, leurs architectures et leurs caractéristiques. Nous illustrons par la suite leurs domaines d'applications, leurs architectures de communication ainsi que les différentes contraintes et facteurs influant leurs conceptions.

1.2 Le capteur

L'utilisation des capteurs sans fil est de plus en plus demandée dans la vie courante. Les industries proposent des capteurs sans fil qui peuvent renseigner l'utilisateur sur plusieurs données selon le domaine d'application. Ces capteurs ont une architecture et des caractéristiques bien déterminées qui seront détaillées par la suite.

1.2.1 Définition d'un capteur

Système ou organe qui sert à détecter un phénomène physique ou chimique, afin de prélever des informations comme la pression, la température et les vibrations qui seront élaborées à une autre grandeur de nature différente (très souvent électrique) (voir figure 1.1) utilisable à des fins de mesure ou de commande [1, 2].

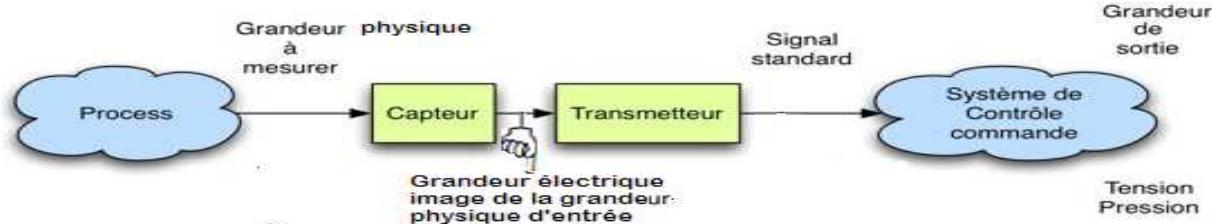


Fig.1.1 Processus De Capture.

Ces dernières années, le marché mondial a connu un intérêt primordial aux systèmes embarqués et technologies de miniaturisation ce qui a permis l'apparition d'un nouveau type de capteurs, nommés capteurs intelligents (smart sensors)¹ qui peuvent être définis selon deux points de vue :

- Fonctionnel: ce sont des organes capables de détecter, mesurer, convertir et traiter les données collectées en vue de les communiquer à d'autres dispositifs du système [4].
- Technologique: correspondent principalement à de véritables systèmes miniaturisés qui intègrent dans un volume parfois extrêmement réduit dans le corps du capteur, un organe de calcul interne (microprocesseur, microcontrôleur), un système de conditionnement du signal (programmable ou contrôlé) et une interface de communication bidirectionnelle avec l'extérieur² [19, 7].

Deux types de nœuds capteurs peuvent être distingués : capteurs (nœuds ordinaires) et noeuds puits [3]. Un nœud capteur détecte les phénomènes physiques et surveille son environnement immédiat pour transmettre les données nécessaires au nœud puits. Ce dernier les collecte, les stocke et les analyse pour communiquer les résultats à d'autre réseaux auquel il est relié par Internet ou Satellite [3, 4].

1.2.2 Architecture d'un capteur

L'architecture d'un capteur comprend deux parties: matériel et système d'exploitation embarqué.

1.2.2.1 Matériel

Un nœud capteur est composé de quatre unités [4] ou sous-systèmes principaux (voir figure 1.2). Il s'agit des sous-systèmes: de calcul ou traitement, communication, perception et de contrôle d'énergie [4, 5, 6, 8, 10]. Il peut contenir également, un ensemble supplémentaire de modules selon le domaine d'application, comme le système mobilisateur chargé de déplacer le

¹ En ligne : Rapport de veille Techno futur Industrie Mars 2008.

² En ligne : IUT CACHAN / Module2-Capteurs / S.POUJOULY.

micro-capteur en cas de nécessité, un générateur d'énergie (cellule solaire) [4, 5] ainsi qu'un système de localisation GPS, (Global Positioning System) [9].

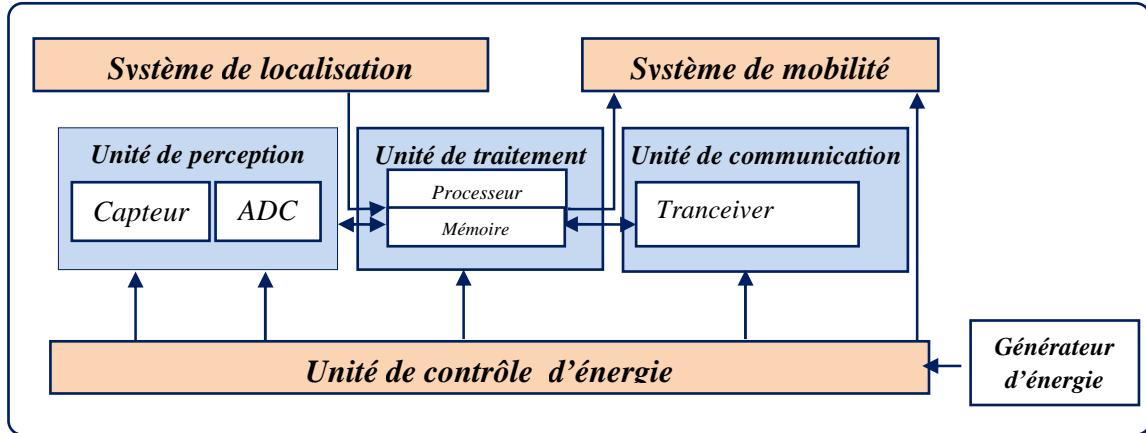


Fig.1.2 Architecture D'un Nœud Capteur.

- *L'unité de perception*: est composée de deux sous-unités: un capteur et un convertisseur Analogique/Numérique (ADC pour Analog to Digital Converter) [6, 8]. Le capteur permet de relier le noeud avec son environnement extérieur, il est responsable de fournir des signaux analogiques basés sur les phénomènes observés à l'ADC. Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement pour pouvoir l'analyser.
- *L'unité de traitement*: est à la charge du contrôle des capteurs et la gestion des procédures et protocoles de communication [4, 2]. De plus le microcontrôleur ou le microprocesseur embarqué est associe généralement à une unité de stockage nécessaire a l'implantation et a l'exécution d'un programme (logiciel) qui peut être un système d'exploitation spécialement conçu pour les capteurs (TinyOS par exemple voir § 1.2.2.2). Ce microprocesseur peut fonctionner selon des modes différents [11] en fonction de la consommation souhaitée ou de la quantité d'énergie restante dans la batterie. Mais faire la navette entre les divers modes de fonctionnement induira à une consommation supplémentaire d'énergie et diminuera donc la durée de vie du capteur.
- *L'unité de communication* : responsable d'effectuer toutes les émissions et réceptions de données sur un medium sans fil afin de permettre l'échange d'informations entre le noeud capteur et son environnement extérieur. Dépendamment du domaine d'application, trois médiums de communication sans fil peuvent être utilisés: optique (laser), infrarouge et radiofréquences (RF : radio ferequency) [12]. Le laser et l'infrarouge n'utilisent aucune antenne mais leurs capacités de diffusion sont limitées, contrairement à la RF qui nécessite une antenne est une portée radio en fonction de l'énergie consommée. Elle reste le moyen le plus répondu pour la communication des capteurs.
- *L'unité de contrôle d'énergie* : c'est la batterie qui, n'est généralement ni rechargeable ni remplaçable. Cette capacité énergétique limitée au niveau des capteurs représente une contrainte cruciale lors de la conception des différentes protocoles pour les réseaux de capteurs. Par ailleurs, les unités d'énergie rechargeables: supportées par des photopiles

(cellules solaire) [4] permettent de convertir l'énergie lumineuse en courant électrique et d'étendre la durée de vie de la batterie.

1.2.2.2 Système d'exploitation

Le système d'exploitation (*Operating System* ou *OS*) n'est qu'un ensemble de programmes responsables d'assurer l'exploitation des ressources matérielles d'un dispositif par les applications utilisatrices. Les OSs destinés aux réseaux de capteurs sont de petite taille, vue l'espace physiquement limité, mais ils doivent présenter plus de performances en termes de temps d'exécution, occupation de mémoire et en gestion d'énergie. Comme exemple de système d'exploitation le plus répandu pour les réseaux de capteurs : TinyOS.

Tiny OS

TinyOS est un système d'exploitation open-source conçu pour les réseaux de capteurs sans fil. Il a été développé à l'Université de Berkeley pour des applications qui fonctionnaient sur des dispositifs tels que les grains (motes) de Berkeley (voir exemples figure 1.3). Actuellement, il est utilisé par plus de 500 universités et centres de recherche dans le monde [13,14].

TinyOS respecte une architecture basée sur une association de composants, réduisant la taille du code nécessaire à sa mise en place (quelques kilo-octets). Cela s'inscrit avec l'utilisation du langage de programmation NesC (syntaxiquement proche du C) [15]. Sa bibliothèque de composant est particulièrement complète puisqu'elle inclut des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. Ces composants peuvent être utilisés tels qu'ils sont, comme ils peuvent être aussi adaptés à des applications précises³. En s'appuyant sur un fonctionnement événementiel, TinyOS propose une gestion très précise de la consommation d'énergie du capteur et permet une meilleure adaptation à la communication sans fil [14].

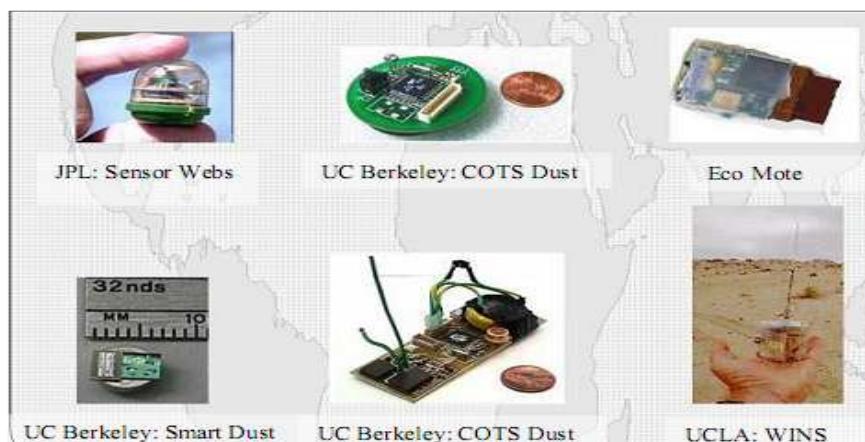


Fig.1.3 Exemples de noeuds capteurs (Berkeley Mote)^[4].

1.2.3 Caractéristiques d'un capteur

Actuellement, les capteurs sont considérés comme des éléments incontournables pour mesurer les paramètres physiques dans des environnements variés en vue de remplir une

³ En ligne : <http://www.tinyos.net/special/mission>

⁴ En ligne : <http://ceng.usc.edu/~raghu> (Cauligi Raghavendra Wireless Sensor Networks and Applications) .

application spécifique. De ce fait, le choix de capteur sera fortement conditionné par les contraintes de l'application et le type d'environnement. Différentes caractéristiques peuvent être tirées [19] dont les plus importantes sont :

- **Précision** : La précision d'un capteur s'exprime en fraction de la grandeur physique mesurée. Elle est généralement présentée comme étant la plus grande erreur prévue entre le réel perçu et les signaux électriques obtenus en sortie du capteur.
- **Sensibilité** : Est définie par le rapport entre le signal physique en entrée et le signal électrique de sortie. Caractérise ainsi l'aptitude du capteur à détecter la plus petite variation de la grandeur à mesurer. A terme d'exemple, une sensibilité élevée d'un capteur thermique se produira lors d'un petit changement de température qui le correspondra par conséquent, un grand changement de tension.
- **Etendue de mesure et bande passante** : L'étendue de mesure n'est qu'une caractéristique qui donne la plage de fonctionnement du capteur pour la grandeur à mesurer. Tandis que la bande passante, exprime la différence des fréquences extrêmes de fonctionnement.
- **La rapidité** : C'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant où l'information sera prise en compte par la partie commande.
- **Fidélité** : Pour une série de mesures de la même valeur de la grandeur d'entrée, le signal délivré par un capteur ne varie pas.
- **Bruit** : Tous les capteurs produisent un certain bruit de rendement, en plus du signal de sortie. Dans beaucoup de cas, le bruit généré limite les performances d'exécution du système embarqué sur le capteur.

1.3 Les réseaux de capteurs sans fil

Contrairement aux réseaux filaires traditionnels, les réseaux de capteurs sans fil offrent une grande flexibilité tout en s'affranchissant aux problèmes liés aux câblage et la mobilité des noeuds.

Dans le paragraphe suivant, nous étudierons ce nouveau type de réseaux sans fil, son architecture, les différences qui le distingue des réseaux ad hoc ainsi que ses principales caractéristiques.

1.3.1 Définition d'un réseau de capteurs

Les réseaux de capteurs sans fil (WSN pour Wireless Sensor Networks) sont considérés comme un type spécial de réseaux ad hoc. Cela sous-entend que ces réseaux sont dépourvus d'infrastructure préexistante et d'administration centralisée, où chaque nœud peut communiquer via des interfaces sans fil (généralement en utilisant le support radio) [12]. Les nœuds du réseau consistent en un grand nombre (centaines voir des milliers) de capteurs matériellement petits, construits à partir des composants pas chers pour maintenir un coût de réseau maniable, et placés généralement près des objets auxquels ils s'intéressent dans les

environnements où ils sont déployés. Ces capteurs sont capables de récolter, traiter et d'acheminer les données environnementales de la région surveillée d'une manière autonome, vers une ou plusieurs station de collecte appelées nœuds puits ou stations de base [17, 22] (voir figure 1.4).

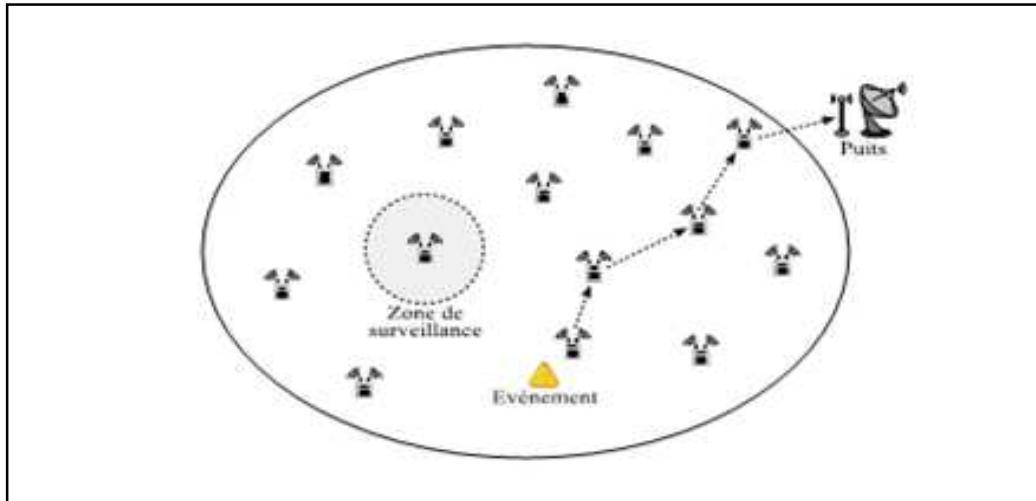


Fig.1.4 Exemple De Réseau De Capteurs.

En effet, ces réseaux se trouvent à la confluence d'une variété de domaines de recherche : systèmes distribués, réseaux mobiles ad hoc, robotiques et systèmes de sécurité. Et donc, leur capacité de communication et souplesse de déploiement, par simple dissémination aléatoire des nœuds capteurs (par exemple : largage depuis un avion ou un bateau) dans la zone de surveillance, à comporter à leur utilisation dans des secteurs différents et permettant ainsi leur intégration dans des environnements hostiles où l'intervention humaine après déploiement est difficile et parfois impossible. Comme par exemple : les incendies de forets, un site après tremblement de terre, champs de surveillances des ennemis et dans des régions glaciaires ou tropicales dans le but est de suivre de manière précise les effets du réchauffement de la planète, les changements climatiques et l'augmentation de la pollution [18, 24].

1.3.2 Architecture du réseau

Les noeuds capteurs décrits précédemment, sont très souvent dispersés à travers une zone géographique appelée champ de captage, qui définit la zone d'intérêt pour le phénomène observé. Les données sensées sont acheminées par une architecture d'infrastructure sans fil à multi-saut (routage multi saut) au point de collecte comme montré dans la figure 1.5. La station de base ou le nœud puits, peut être fixe ou mobile. Il n'est qu'une passerelle entre les noeuds du réseau et le nœud capteur chargé de la gestion des tâches ou l'utilisateur final.

Ainsi, l'usager peut surveiller et commander l'environnement à distance via Internet ou un satellite, en adressant des requêtes précisant le type de données requises à récolter aux autres nœuds du réseau via le nœud puits [1].

Les stations de base possèdent beaucoup plus de capacités que les capteurs tant au niveau de la mémoire que de la vitesse de traitement ou des réserves en énergie [4].

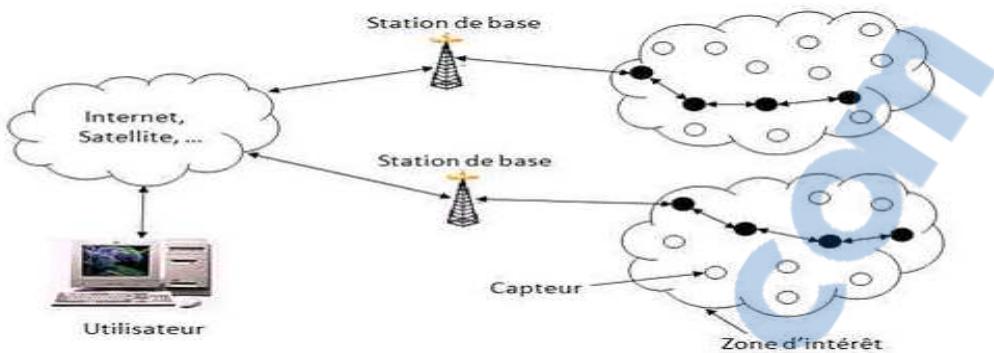


Fig.1.5 Architecture d'un réseau de capteurs.

1.3.2.1 Topologies du réseau

Pour prendre en charge le transfert de données depuis les nœuds capteurs jusqu'au noeud puits, le réseau peut être configuré selon différentes topologies (tiré de [7] [19]).

La topologie en étoile (Star Network)

C'est une topologie simple (représentée par la figure 1.6), où chaque nœud communique ses mesures directement à l'unique station de base sans avoir la possibilité d'échanger des messages avec les autres capteurs. La station de base est la seule à pouvoir recevoir ou envoyer des messages aux nœuds du réseau. Cette approche peut de manière significative simplifier la conception, car les soucis de gestion du réseau sont réduits au minimum. Néanmoins, elle présente des limites en terme : de scalabilité et de robustesse, due à sa dépendance sur un seul nœud pour contrôler et gérer le réseau.

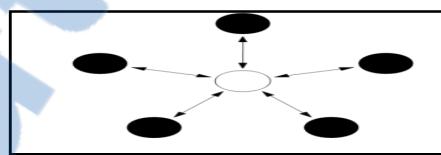


Fig.1.6 La topologie en étoile.

La topologie maillée (Mesh Network)

Dans ce type de topologie, une communication multi sauts est utilisée comme mode d'acheminement de messages, où n'importe quel nœud peut envoyer à n'importe quel autre nœud dans le réseau à condition qu'il soit dans sa portée de transmission, sinon un nœud intermédiaire intervient pour envoyer le message au nœud destinataire (voir Figure 1.7). L'avantage de cette topologie est la possibilité du passage à l'échelle et la tolérance aux pannes, par contre la consommation d'énergie dans la communication multi sauts et les latences créées sont les inconvénients majeurs de cette topologie.

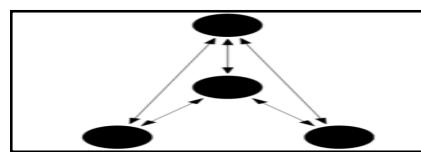


Fig.1.7 La topologie maillée.

La topologie hybride (Hybrid Star – Mesh Network)

Pour minimiser la consommation d'énergie dans les réseaux de capteurs, une topologie hybride entre celle en étoile et en mesh est conçue (voir figure 1.8). Elle fournit des communications réseau robustes et diverses. Il existe des nœuds qui ont la possibilité de router les messages des autres noeuds à faible puissance. Généralement, ces noeuds à capacité multi sauts ont une puissance plus élevée que les autres nœuds du réseau.

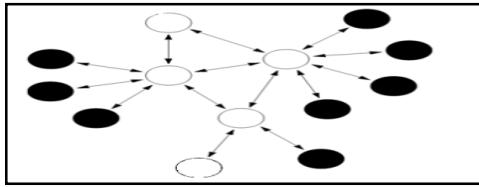


Fig.1.8 La topologie hybride.

1.3.3 Caractéristiques du réseau de capteurs

Comme les réseaux de capteurs sans fil sont apparentés aux réseaux ad hoc, ces deux types de réseaux ont de nombreux points communs comme l'absence d'infrastructure, l'autonomie et l'utilisation des ondes radio pour la communication [20]. Or, les contraintes et les besoins de ces réseaux diffèrent, le tableau ci-dessous montre ces différences.

	Réseau de capteurs	Réseau ad-hoc
Composition	Petits micro-capteurs	Portables, PDA.....
Flot de communication	plusieurs à un (many to one)	Plusieurs à plusieurs (many to many)
Communication	Diffusion /Aggregation	Point à point
Topologie	Mobilité faible	En constante évolution et mobilité forte
Relation entre les nœuds	Collaborant pour le même objectif	Chaque nœud à son propre objectif
Identification des noeuds	Très grand nombre de nœuds n'ayant pas tous une ID	Présence de la notion d'ID
Objectif du réseau	Objectif ciblé	Générique / communication
Contrainte clé	Ressource énergétiques	Débit /QOS

Tab.1.1 - Comparaison entre le réseau de capteurs et le réseau ad hoc [4, 20].

Dans le paragraphe suivant on va détailler d'autres caractéristiques spécifiques aux réseaux de capteurs sans fil.

Contrainte d'énergie

Dans les réseaux ad hoc, la consommation d'énergie a été considérée comme un facteur déterminant mais pas primordial car les ressources énergétiques peuvent être remplacées par l'utilisateur. Ces réseaux se focalisent plus sur la QoS que sur la minimisation de la consommation d'énergie. Par contre, dans les réseaux de capteurs, la consommation d'énergie est une contrainte très importante puisque généralement les capteurs sont déployés dans des zones inaccessibles. Ainsi, il est difficile voire impossible de remplacer les batteries après leur épuisement. De ce fait, la consommation d'énergie au niveau des capteurs a une grande influence sur la durée de vie du réseau en entier [4, 21].

Forte densité

Les réseaux de capteurs sont caractérisés par une grande densité des noeuds déployés dans la région à surveiller. Un réseau peut contenir des centaines voir des milliers de nœuds capteurs.

Ressources limitées

Les capteurs sont des objets communicants à taille très réduite ce qui limite leurs ressources en termes de : mémoire disponible, puissance de traitement, bande passante et particulièrement en quantité d'énergie embarquée.

Déploiement des nœuds

Il existe, en effet, plusieurs stratégies pour déployer les nœuds d'un réseau de capteurs sur une zone à surveiller, le choix d'une de ces stratégies dépend du type de l'application et de l'objectif du réseau. Le déploiement peut être déterministe ou aléatoire [1, 21].

Dans la première stratégie, les capteurs sont placés manuellement et les données sont conduites via des chemins pré-déterminés, mais cette solution devient impossible dès que l'on considère un très grand nombre de capteurs. Le déploiement aléatoire consiste à disperser les capteurs sur le champ de captage en masse (ex : lâchés d'un avion). D'autre part, un mélange de ces deux techniques peut être envisagé.

Les caractéristiques de déploiement, le fonctionnement autonome et la fréquence élevée de pannes (due à l'épuisement d'énergie ou la destruction) rendent la maintenance de la topologie d'un réseau de capteurs complexe. En effet, plusieurs centaines de capteurs sont déployés avec une densité pouvant être supérieure à 20 nœuds par m³, ce qui exige une bonne gestion de la maintenance de la topologie du réseau déployé. De plus, il est toujours préférable de prévoir une forte densité autour de la station de base car toutes les données sont acheminées vers cette station [21].

La mobilité

La mobilité devient une question clé pour certains types de réseaux de capteurs. Par exemple, dans des applications de détection où les capteurs sont embarqués sur des dispositifs mobiles tels que les véhicules, ou sur des animaux. Si la mobilité est trop fréquente, elle ne peut être considérée comme un problème secondaire dans ce type d'environnement.

Ainsi, la détection des voisins et la reconfiguration du réseau exigent habituellement un nombre important de messages de contrôle de topologie, ce qui induit à une dépense importante en énergie. En outre, un autre type de mobilité pourrait être pris en compte, qui est la mobilité de la station de base et dans d'autres cas la mobilité des deux dispositifs en même temps : noeuds et station de base [4, 21].

Auto organisation du réseau et la dynamicité de la topologie

Ceci peut être nécessaire, vu la grande densité des noeuds capteurs et leur déploiement dans des zones d'intérêt critiques où aucune intervention humaine ne peut être possible pour assurer leur organisation. Et donc, les noeuds peuvent échouer (par manque d'énergie ou destruction physique), comme de nouveaux nœuds peuvent joindre le réseau. Par conséquent, le réseau doit

être capable de s'auto organiser et de se modifier périodiquement de sorte qu'il puisse s'adapter aux changements de la topologie tout en assurant son fonctionnement [21].

Couverture limitée

Dans les réseaux de capteurs, chaque nœud obtient une certaine vue locale de l'environnement qui est limitée par sa portée (voir figure 1.9). La couverture d'une vaste surface est composée de l'union de nombreuses couvertures de petite taille assurée par ces noeuds.

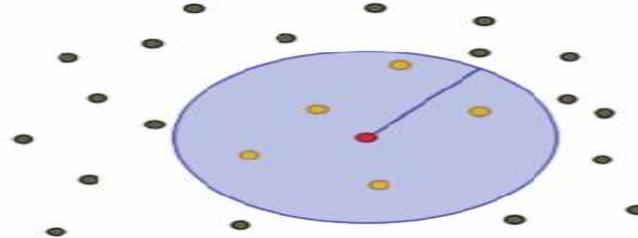


Fig.1.9 La vision d'un capteur Dépend du rayon de réception.

Connectivité

La densité très élevée des noeuds dans les réseaux de capteurs exclue complètement leur isolement et augmente ainsi leur connectivité. Un réseau de capteurs est dit connecté si et seulement si, il existe au moins une route entre chaque paire de nœuds [16]. La connectivité dépend essentiellement de l'existence des routes. Elle est affectée par les changements de la topologie du à la mobilité et la défaillance de ces nœuds.

Sécurité physique limitée

A l'instar des réseaux ad hoc, les réseaux de capteurs sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui minimisent le contrôle sur les données transférées [23].

1.4 Domaines d'applications

Le faible coût de production et l'usage de dispositifs miniaturisés et à faible consommation dans les réseaux de capteurs a révélé l'idée de leurs utilisations dans des domaines très variés. Les multiples applications envisagées dans ces réseaux sont souvent classés en cinq familles [4, 7, 25, 26].

1.4.1 Applications militaires

Comme pour beaucoup d'autres technologies, ce sont historiquement les militaires qui se sont intéressés les premiers aux réseaux de capteurs. Aujourd'hui, ces réseaux sont devenus un axe de recherche très important. Ils peuvent être rapidement déployés pour la surveillance des champs de bataille et utilisés pour fournir d'une manière intelligente l'endroit, le nombre, les mouvements et l'identité des troupes et des véhicules, ainsi que la détection des produits chimiques, biologiques et armes nucléaires.

1.4.2 Applications environnementales

Les réseaux de capteurs peuvent être employés pour détecter et surveiller les changements environnementaux des : forêts, océans, réserves naturels et champs agricoles. Ceci permet l'avertissement de tout risque survenu sur ces zones d'intérêts comme par exemple: le signale de début d'incendie, qui pourrait considérablement augmenter l'efficacité de la lutte contre les feux de forêts.

Sur les sites industriels, les centrales nucléaires ou pétrolières, des capteurs déployés peuvent détecter les fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et donc alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace.

1.4.3 Applications à la sécurité

Pour contrôler la sûreté, l'antisismique et la déformation, les bâtiments et les structures intègrent dans les murs des capteurs sans alimentation électrique ou autres connexions filaires. Les différentes applications des réseaux de capteurs dans la sécurité pourraient diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant de meilleurs résultats.

1.4.4 Applications médicales

L'utilisation des réseaux de capteurs dans les applications de santé est un potentiel très bénéfique en termes de qualité de vie et surveillance des patients et des médecins à l'intérieur d'un hôpital, pareillement, dans les sites éloignés, telle l'assistance à domicile des personnes âgées. Des capteurs peuvent être avalés ou implantés sous la peau (micro-caméras) offrant la possibilités de collecter les informations des fonctions vitales de l'être humain, réduisant ainsi le retard d'obtention des résultats de diagnostics de certains maladies et limitant le recours à la chirurgie en transmettant des images de l'intérieur du corps humain.

1.4.5 La domotique

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs..., donnant ainsi naissance à un autre type d'application dans lequel les réseaux de capteurs émergent : la domotique [13]. Dans ces applications, le réseau de capteurs est déployé dans l'habitation. Le principe est que le réseau forme un environnement, dit environnement pervasif. Son but est de fournir toutes les informations nécessaires aux applications de confort, de sécurité et de maintenance dans l'habitat. Les capteurs sont des capteurs de présence, de son, ils peuvent même être équipés de caméras. Un tel réseau doit permettre de créer une maison intelligente capable de comprendre des situations suivant le comportement des occupants et d'en déduire des actions.

Ces réseaux sont donc très hétérogènes (des éléments d'électroménager peuvent faire partie du réseau aussi bien que les ordinateurs personnels). Il se peut que certains éléments aient besoins d'être économies en énergie mais ce n'est pas le cas de tout le réseau (il ne s'agit pas pour ces applications de réseaux grande échelle).

En revanche, ces réseaux doivent être hautement reconfigurables d'une part la topologie du réseau peut changer d'un jour à l'autre avec l'aménagement, d'autre part on peut avoir besoin de changer le type d'application pendant la vie du réseau.

1.5 La communication dans les réseaux de capteurs

Dans les réseaux de capteurs, les nœuds doivent bien réagir avec l'environnement où ils sont placés. Ainsi, ils doivent permettre une communication multi sauts pour les données qui circulent dans la zone de capture. Pour cela, un modèle de communication est proposé par Ian et al [4] dont le rôle principal est la standardisation de la communication entre les participants afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce modèle (détailé dans 1.5.1) combine l'énergie et le routage, intègre les données avec les protocoles réseaux et promeut une communication efficace entre les différents nœuds à travers un medium sans fil.

1.5.1 Pile protocolaire (modèle en couches)

Par analogie au modèle OSI (Open Système Interconnexion) des réseaux filaires, le modèle de communication utilisé dans les réseaux de capteurs comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI.

Comme la communication n'est pas le seul souci dans les réseaux de capteurs, il y a d'autres critères très importants qu'il faut en tenir compte. De ce fait, d'autres couches supplémentaires sont ajoutées pour gérer l'énergie, la mobilité des nœuds et l'ordonnancement des tâches.

Le rôle et les caractéristiques des couches : application, transport, réseau, liaison de données, physique et les niveaux intégrés : plan de gestion d'énergie, plan de gestion de mobilité et le plan de gestion de tâche (voir la figure 1.10) sont résumés dans le paragraphe suivant.

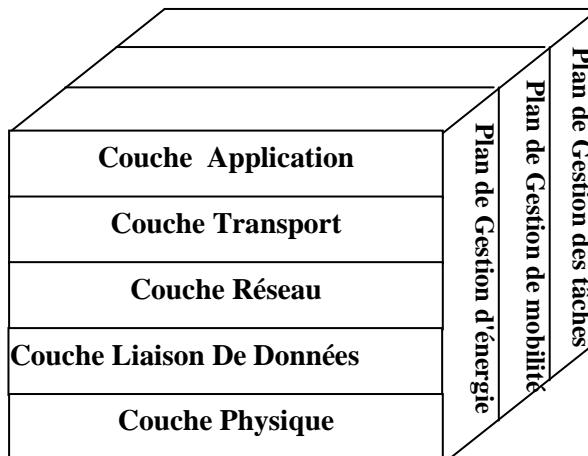


Fig. 1.10 Modèle en couches (pile protocolaire) d'une architecture de réseau de capteurs [3].

Couche application

Suivant la fonctionnalité des capteurs, différentes applications peuvent être utilisées et implémentées sur cette couche. Elle fournit des mécanismes pour permettre à l'utilisateur d'interagir avec le réseau en lui fournissant des interfaces pour la création et la diffusion des requêtes et un moyen d'interpréter les réponses reçues [3].

Couche transport

Cette couche constitue une interface entre la couche application et la couche réseau. Elle est responsable : du transport de données, leur découpage en paquets, contrôle de flux,

conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission [3].

Couche réseau

Dans un réseau de capteurs, les nœuds sont dispersés à forte densité pour observer un phénomène dans une région. Par conséquent, ils sont très proches les uns des autres. Pour assurer la propagation du signal sans effets de dégradation, les réseaux de capteurs utilisent une communication multi sauts. De plus, les nœuds doivent consommer moins d'énergie pour acheminer les données capturées dans le réseau au nœud puits, ce qui rend l'utilisation des protocoles de communication traditionnels des réseaux Ad hoc impraticables. Par ailleurs d'autres caractéristiques peuvent être citées [22] afin d'illustrer le besoin de nouveaux protocoles de routage pour les réseaux de capteurs:

- il n'est pas possible d'établir un système d'adressage global vu le grand nombre de nœuds capteurs.
- les applications des réseaux de capteurs exigent l'écoulement des données mesurées par de multiples sources à un nœud puits particulier.
- Les multiples capteurs peuvent produire les mêmes données à proximité d'un phénomène (redondance des données captées due à la forte densité de déploiement).
- les capteurs exigent une gestion soigneuse des ressources disponibles à savoir : la puissance de transmission, énergie de la batterie, ainsi que la puissance de traitement et stockage.

Couche liaison de données

La couche liaison [4, 22] est à la charge du multiplexage de données, détection des trames, la gestion de l'accès au support de communication et le contrôle d'erreurs. Elle assure la fiabilité de la communication point à point et multipoint. Cependant, les protocoles MAC (Media Access Control) conventionnels ne sont pas directement applicables dans les réseaux de capteurs. Cette couche, exige l'utilisation de protocoles qui doivent être capables : de réduire au minimum la collision avec l'émission des voisins et de minimiser les retransmissions.

Couche physique

Dans cette couche les besoins d'une modulation simple et robuste à motiver l'utilisation des techniques de transmission et de réception de données afin d'assurer la sélection des fréquences, la génération des porteuses et le cryptage des différentes données [4]. De manière générale, le choix d'un bon schéma de modulation est critique et dépend du système utilisé.

Les niveaux intégrés dans la pile protocolaire

Trois couches supplémentaires sont intégrées dans la pile protocolaire : plan de gestion d'énergie, plan de gestion de mobilité et le plan de gestion de tâche.

1. Plan de gestion d'énergie

Cette partie gère la manière dont les noeuds utilisent leurs énergies. Généralement un capteur ne nécessite qu'une source énergétique limitée (< 0.5 Ah, 1.2 V). Comme la vie du nœud a une dépendance forte à l'égard de la vie de la batterie, il doit par conséquent contrôler et minimiser sa consommation d'énergie. Par exemple, après la réception d'un

message, le capteur éteint son récepteur et se met en mode sommeil afin d'éviter la duplication des messages déjà reçus. En outre, si le niveau d'énergie atteint un seuil bas, le nœud diffuse à ses voisins une alerte pour les informer qu'il ne peut pas participer au routage. L'énergie restante est réservée à la capture [4].

2. Plan de gestion de mobilité

Il permet la détection et l'enregistrement des mouvements des nœuds capteurs afin de maintenir des informations sur leurs localisations et d'entretenir continuellement une route vers l'utilisateur final. Dans plusieurs cas les nœuds capteurs peuvent être mobiles suite à une configuration d'une nouvelle topologie ou bien par un changement d'emplacement désiré par l'utilisateur. Le système de gestion de mobilité doit être capable de commander les nœuds pour réaliser les mouvements nécessaires [4].

3. Plan de gestion de tâche

Ordonnance et balance les différentes tâches de capture de données dans une région spécifique. Il n'est pas nécessaire que tous les nœuds de cette région effectuent la tâche de capture en même temps ; certains nœuds exécutent cette tâche plus que d'autres selon leur niveau de batterie [4]

1.5.2 Technologies et standards de communication

Pour assurer la communication sans fil entre les nœuds d'un réseau de capteur, un certain nombre de modalités de communication peuvent être employées. Par exemple: la radio, la lumière diffusée, le laser...etc.

Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs on retrouve : WIFI (IEEE 802.1), Bluetooth (IEEE802.15.1 and .2), Zigbee (IEEE 802.15.4).

WI-FI (IEEE 802.11)

Ce standard [8] soutenu par l'alliance WECA (Wireless Ethernet Compatibility Alliance), est très connu actuellement, son nom vient de l'expression anglaise '*'Wireless Fidelity'*'. Un certain nombre de normes dérivées ont été créées. La technologie WiFi (802.11 et ses variantes), bien qu'elle soit conçue pour les LANs sans fil qui se composent habituellement d'ordinateurs portables et de PDAs, est également supposée utilisable aux réseaux de capteurs. Cependant, la consommation d'énergie élevée et le débit excessif rendent les protocoles 802.11 non appropriés aux réseaux de capteurs. Ce fait a motivé les chercheurs à concevoir des protocoles MAC efficaces en énergie et spécifiques aux réseaux de capteurs (voir § 2.2.2).

Bluetooth (IEEE 802.15.1 et .2)

La technologie Bluetooth, dont Ericsson a initié le projet en 1994 a pour but principal de remplacer les câbles sur de petites distances. Elle est utilisée dans les téléphones portables comme interface de connexion pour accéder à un PC. Malheureusement, le grand défaut de cette technologie est sa trop grande consommation d'énergie. Elle ne peut donc pas être utilisée par des capteurs qui sont alimentés par une batterie et qui, idéalement, devraient fonctionner

durant plusieurs années [13]. En voici quelques caractéristiques de la technologie Bluetooth tirées de [27] :

- Bluetooth est une technologie sans fil, déposé comme standard à l'IEEE, c'est le 802.15. Elle fonctionne dans le spectre de fréquences 2,4 GHz.
- Bluetooth est destiné à un usage personnel et se classe dans la catégorie PAN (Personal Area Network) ; réseau à zones restreintes de quelques dizaines de mètres.
- La technologie sans fil Bluetooth est orientée vers les applications données et voix .
- Elle peut fonctionner sur une distance de 10 ou de 100 mètres selon la classe de l'appareil Bluetooth. Le débit de données maximal avec des données rehaussées s'élève à 3 Mbits/s .
- Les ondes de la technologie sans fil Bluetooth peuvent traverser des objets massifs .
- Cette technologie est omnidirectionnelle et ne nécessite pas la visibilité directe des appareils connectés.

Lors du développement de la spécification Bluetooth, la sécurité a toujours été et continue d'être une priorité.

Zigbee (IEEE 802.15.4)

La technologie Zigbee, gérée par la Zigbee alliance et combiné avec IEEE 802.15.4, offre des caractéristiques qui répondent mieux aux besoins des réseaux de capteurs. Cette technologie qui est destiné aux environnements à faible consommation d'énergie et équipements électroniques portables permet un transfert stable de données, une installation facile, un coût réduit ainsi qu'une très basse consommation d'énergie nettement moins que Bluetooth (voir tableau 1.2)[13, 28]. Ces performances en termes de consommation d'énergie ont poussé plusieurs industriels de capteurs à l'intégrer dans leurs produits capteurs comme MicaZ, Telos, ...etc. Zigbee [28] fonctionne globalement sur la bande de fréquences des 2,4 GHz mais également à 916 MHz en Amérique du nord et à 866 MHz bande libre en Europe. Ses principales caractéristiques sont :

- Débits compris entre 20 et 250 Kbits/s.
- Supporte différentes types de topologies réseau (étoile, mesh).
- Signal robuste et résistant aux interférences (mode d'accès CSMA/CA).
- Pénétration à travers les murs et plafonds.
- Sécurité: exige un contrôle d'accès, cryptage de données (128-bit AES, cryptage + authentification).
- permet de connecter jusqu'à 255matériels par réseau sur une portée allant jusqu'à 100 mètres.

- Cycles d'émissions/réceptions et connexions au réseau très rapides ce qui permet une faible consommation d'énergie.

Ci-dessous un tableau comparatif entre les trois technologies de communications décrites précédemment.

Technologie	WIFI	Zigbee	Bluetooth
IEEE	802.11a/b/g	802.15.4	802.15.1
Besoins mémoire	1 Mb +	4 - 32 Kb	250 Kb +
Autonomie avec pile	Heures	Années	Jours
Vitesse de transfert	11-54 et + Mb/s	250 KB/s	1 Mb/s
Portée	300 m	100 m	10-100 m

Tab.1.2 - comparaison entre les trois technologies IEEE802.11, Bluetooth et Zigbee.

1.6 Facteurs et contraintes influant la conception du réseau

Plusieurs facteurs et contraintes guident la conception des réseaux de capteurs et agissent sur leur bon fonctionnement [4, 7, 21, 30, 33].

1.6.1 Tolérance aux pannes

Certains nœuds peuvent être défaillants ou inhibés à cause d'un épuisement d'énergie, ou subissent des dommages physiques (écrasés par des animaux ou lorsqu'ils sont jetés par un avion). La défaillance de ces capteurs ne doit pas affecter la globalité de la tâche du réseau. La tolérance aux pannes est la capacité de soutenir les fonctionnalités d'un réseau de capteurs sans causer des interruptions lorsqu'un capteur cesse de fonctionner (la panne ne devrait pas avoir une influence sur le fonctionnement du réseau) [49, 58]. De ce fait, des protocoles et des techniques peuvent être conçus pour évaluer le niveau de la tolérance aux pannes exigé dans les réseaux de capteurs. Si les capteurs sont déployés dans un habitat, la tolérance aux pannes exigée peut être basse, puisque ce type de réseaux n'est pas facilement endommagé, alors que si les capteurs sont déployés dans un champ de bataille pour la surveillance et la détection, la tolérance aux pannes devrait être élevée puisque les données surveillées sont critiques et les capteurs peuvent être détruits facilement par des actions hostiles.

1.6.2 Passage à l'échelle (scalabilité)

Le nombre de capteurs déployés peut être de l'ordre de la centaine ou du millier. Le réseau doit donc être capable de fonctionner efficacement avec une grande quantité de capteurs. De plus, les nœuds doivent traiter un grand nombre d'événements sans être saturés.

1.6.3 Topologie dynamique

La plupart des architectures réseau reposent sur des capteurs statiques. Pourtant, la mobilité des stations de base et/ou des capteurs est parfois nécessaire dans de nombreuses applications. Par exemple le cas des capteurs attachés à des objets mobiles qui se déplacent d'une façon libre

et arbitraire rendant ainsi, la topologie du réseau fréquemment changeante et en constante évolution. Egalement, l'état d'activité des capteurs (extinction, mise en veille et actif) intervient dans les changements de la topologie. La conception d'un protocole d'auto-organisation qui s'adapte continuellement et rapidement aux changements, s'avère nécessaire pour assurer le bon fonctionnement du réseau.

1.6.4 Coût de production

Le coût de production d'un seul capteur est très important pour l'évaluation du coût global du réseau car ces réseaux sont constitués d'un grand nombre de nœuds capteurs ce qui rend le prix d'un nœud très critique. Si ce dernier est coûteux, alors le coût du réseau sera très élevé et non maniable. Pour cela le coût d'un capteur ne doit pas être cher (souvent pas plus de 1\$) afin d'avoir un coût global du réseau raisonnable. Dès lors, la minimisation du coût de production du nœud capteur constitue un grand défi mené par les chercheurs, vu les fonctionnalités que doivent comporter ces nœuds et l'objectif désiré pour un coût inférieur à 1\$.

1.6.5 L'environnement

Les capteurs sont souvent déployés en masse dans des endroits tels que les champs de bataille au-delà des lignes ennemis, à l'intérieur de grandes machines, au fond d'un océan et dans des champs biologiquement ou chimiquement souillés. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans ces régions relativement éloignées ou hostiles.

1.6.6 Qualité de service

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

1.6.7 Contraintes matérielles

Un nœud capteur est constitué d'un ensemble de composants (voir §1.1.2.1) nécessitant d'être regroupés dans un module convenable qui peut être de la taille d'une boîte d'allumettes [30]. La taille exigée peut être plus petite que même un centimètre cube [34]. Cette dimension assez réduite permet un déploiement flexible et simple du réseau, mais par contre elle limite la puissance de calcul, capacité de stockage et l'autonomie d'énergie. La contrainte énergétique dans ce type de réseaux étant très forte, puisque l'on considère généralement qu'il est difficile voire impossible de recharger les batteries des capteurs (capteurs déployés dans des zones hostiles et inaccessibles). L'ordonnancement d'activité et l'auto-organisation sont considérés comme des solutions de premier choix, dans le but d'éteindre l'équipement radio de certains capteurs et permettre seulement aux capteurs disposant de plus d'énergie de transmettre les données capturées à la station de base.

1.6.8 Les médias de transmission

Les nœuds communicants sont liés par un médium sans fil qui peut être : la radio (WIFI, Zigbee), le Bluetooth. Le médium doit être compatible avec l'environnement de l'application.

Les capteurs possèdent le matériel nécessaire pour effectuer des communications par ondes radio. Toutefois, la diffusion de l'information via ces liens est peu aisée à cause de l'instabilité et du manque de fiabilité qu'ils présentent. De plus, l'utilisation d'un médium de communication partagé pour faire face aux interférences radio, réduit considérablement la capacité d'exploitation du canal.

1.6.9 Consommation d'énergie

Les capteurs utilisent leur réserve d'énergie à des fins de calcul et de transmission de données. La durée de vie d'un capteur dépend essentiellement de celle de sa batterie. Dans un réseau de capteurs, chaque nœud peut jouer le rôle d'émetteur et de routeur. Par conséquent la défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et impose une réorganisation coûteuse. De ce fait, la majorité des recherches actuelles se concentrent principalement sur les moyens de réduire cette consommation énergétique.

1.7 Différentes problématiques et axes de recherche dans les réseaux de capteurs

Les travaux de recherche menés dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques. Tous les challenges posés dans ces réseaux sont dus principalement à la taille des capteurs. Autrement dit, on recherche une miniaturisation maximale et des performances optimales quant à la transmission, le débit et la consommation d'énergie qui est un facteur primordial dans ce type de réseaux. Parmi ces problématiques, nous citons :

- *Routage* : concevoir un protocole de routage performant en termes de minimisation de la consommation d'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets, ainsi qu'au passage du réseau à l'échelle sans que ses performances se dégradent, constitue la plus grande part des travaux réalisés dans le domaine des réseaux de capteurs. La majorité des recherches menées actuellement sur les protocoles de routage se focalisent sur le routage hiérarchique. Le routage hiérarchique consiste à structurer le réseau en un ensemble de groupes nommés clusters selon un processus de clustering. Un cluster est constitué d'un chef 'cluster-head' et de ses membres. Chaque nœud membre envoie les données captées à son cluster-head. Pour garantir cela, deux schémas sont utilisées: dans le premier, un nœud membre ne communique qu'avec son 'cluster-head'. Par contre, le deuxième schéma tolère l'utilisation d'autres nœuds cluster-heads comme relais. Le chapitre trois présente en détails cette approche de clustering.
- *Couche MAC* : la spécificité des réseaux de capteurs sans fil nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux. Ceci dans le but de minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets. Le chapitre suivant va détailler quelques travaux réalisés selon cet axe.
- *Qualité de service* : des protocoles au niveau de la couche MAC devraient être capables d'établir des priorités entre les flux, limiter les pertes de paquets pour la gestion du réseau, ou du moins en restreindre l'impact. En outre, dans les réseaux de capteurs orientés événements où l'information devra être remontée au centre de contrôle dans un meilleur délai, il est nécessaire d'optimiser conjointement les critères latence et

conservation d'énergie, car la faible latence peut avoir un impact négatif sur le nombre de communications et par la suite sur la consommation d'énergie dans ces réseaux. Pour cela, les protocoles de routage garantissant la qualité de service doivent également respecter la contrainte énergétique. Cet axe de recherche motive d'avantage les chercheurs et attire de plus en plus d'intention.

- *Cross-layer* : dans les modèles classiques, les concepteurs essaient d'optimiser les performances au niveau d'une couche indépendamment des autres couches. Par exemple, le routage et les fonctions de la couche MAC sont optimisés indépendamment les uns des autres. Cependant, une telle indépendance est communément considérée comme trop onéreuse pour les réseaux de capteurs. Par conséquent, une coopération permettant un compromis entre performances, dépendance et flexibilité doit être proposée pour optimiser les capacités du réseau en général et sur tous les niveaux.
- *Diffusion de l'information* : les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées. Et donc, pour la conception d'un protocole efficace pour ce type de réseaux on doit assurer une couverture maximale des capteurs composant le réseau, minimiser le nombre des réémetteurs et des réceptions redondantes et tolérer la consommation d'énergie.
- *Sécurité* : pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité et d'intégrité doivent être mis en place au sein de leur communauté. Les algorithmes de cryptographie conçus pour les réseaux de capteurs doivent tenir compte des ressources limitées que présentent ces réseaux.

1.8 Conclusion

L'intérêt pour les réseaux de capteurs s'accroît considérablement, ils sont devenus actuellement des éléments indispensables dans plusieurs domaines tout en exploitant de multiples technologies de communication. Les différentes utilisations possibles de ces réseaux démontrent la diversité de leurs applications. En effet, les applications des réseaux de capteurs sont fortement dépendantes du domaine et des objectifs envisagés. A chaque application correspond une architecture du réseau et une implémentation. Ainsi, le déploiement de telle infrastructure ou topologie, à la fois dynamique et reconfigurable pose de grands défis auxquels il faut répondre. Par exemple la durée de vie limitée de la batterie du capteur, où chaque nœud est alimenté par une source d'énergie généralement irremplaçable et limitée, pose un grand problème sur la durée de vie totale du réseau en entier. Pour cela, beaucoup de chercheurs essayent d'intégrer des mécanismes qui permettent de prolonger la durée de vie du réseau. Ils consacrent une grande importance à la conservation d'énergie de la communication qui est la plus gourmande dans ce type de réseaux, et alors ils développent par la suite des protocoles de routage appropriés.

Le chapitre suivant est consacré à l'étude détaillée du concept d'énergie dans les réseaux de capteurs sans fil ainsi que les techniques utilisées pour sa conservation, et aussi pour présenter des protocoles de routages à contrainte énergétique déployés pour ces réseaux.

Chapitre 2

Routage à basse consommation d'énergie

2.1 Introduction

Les réseaux de capteurs sans fil prennent de l'ampleur dans notre quotidien, et la recherche internationale, aussi bien académique qu'industrielle y accorde de plus en plus d'efforts. En revanche ces réseaux posent un certain nombre de défis scientifiques. La majorité des travaux de recherche menés actuellement se concentrent principalement sur les moyens de réduire au minimum la consommation d'énergie de sorte à maximiser la durée de vie du réseau. Cette contrainte énergétique est induite par l'utilisation de capteurs alimentés par des batteries à capacités limitées et très souvent irremplaçables. Ces capteurs assurent l'acquisition des données, leur traitement et leur communication à la station de base. C'est cette dernière tâche qui est la plus consommatrice d'énergie. Un bon schéma de gestion d'énergie doit en priorité, prendre en compte les facteurs intervenants dans sa consommation.

La plupart des protocoles de communications utilisés dans les réseaux Ad hoc ne s'adaptent pas aux particularités des réseaux de capteurs. D'où, la nécessité de les améliorer ou de développer de nouveaux protocoles assurant la conservation d'énergie afin d'augmenter la durée de vie du réseau.

Dans ce chapitre, on va aborder en premier lieu le concept d'énergie, puis décrire les différents facteurs intervenant dans sa consommation ainsi que les approches utilisées pour sa conservation.

Dans la seconde partie de ce chapitre, après avoir introduit le concept de routage dans les réseaux de capteurs, un panorama de différentes classifications proposées dans la littérature pour les protocoles utilisés dans ces réseaux, ainsi que les facteurs influant leur conception, sont présentés. Pour chaque classe, des exemples sont fournis. La conclusion de ce chapitre est donnée dans la section 2.4.

2.2 Energie

Comme les nœuds capteurs sont des composants microélectroniques, ils ne peuvent être équipés que par des sources limitées d'énergie. De ce fait, l'énergie est un phénomène prépondérant pour le bon fonctionnement du réseau et d'une considération importante dont les protocoles de routage doivent en tenir compte. Dans le paragraphe suivant seront détaillés : la consommation d'énergie, les facteurs intervenant dans cette consommation ainsi que les techniques utilisées pour sa conservation.

2.2.1 Consommation d'énergie dans les réseaux de capteurs

Le but des réseaux de capteurs est de surveiller une zone: prendre des mesures et de les faire remonter vers un certains nœuds qui sont en mesure de relayer l'information à l'utilisateur final. Pour cela, les capteurs déployés utilisent leur énergie pour réaliser trois actions principales : la capture, le traitement et la communication [2, 26, 29, 33].

- ***Energie de perception*** : L'énergie consommée pour effectuer l'acquisition n'est pas très importante. Elle se résume dans l'échantillonnage et le traitement des signaux, la conversion des signaux physiques en signaux électriques et la conversion analogique numérique. Néanmoins, cette énergie varie en fonction du phénomène et du type de surveillance effectué. Par exemple une capture à intervalle régulier consomme moins d'énergie qu'une surveillance continue.
- ***Energie de traitement*** : L'énergie consommée par les opérations de calcul est beaucoup plus faible que l'énergie consommée lors de la communication. Par exemple : l'énergie nécessaire pour transmettre 1 KB sur une distance de 100 m est approximativement équivalente à l'énergie nécessaire pour exécuter trois millions d'instructions avec une vitesse de 100 millions d'instructions par seconde (MIPS) [34].
- ***Energie de communication*** : La communication couvre les communications en émission et en réception et consomme beaucoup plus d'énergie que les autres tâches à cause de l'amplificateur utilisé dans les transmissions. La puissance du support radio : média de transmission utilisé pour assurer la communication est déterminée par plusieurs facteurs comme le type du système de modulation, quantité de données à communiquer et la distance de transmission.

2.2.2 Facteurs intervenant dans la consommation d'énergie

La consommation d'énergie dans les réseaux de capteurs dépend de plusieurs facteurs. Dans ce qui suit, nous présentons les principales sources de perte d'énergie.

1. L'état du module radio

Pour assurer la communication entre les éléments du réseau, les capteurs utilisent leurs modules radio. Ce module est le composant qui consomme le plus d'énergie.

Pour ce module radio, quatre états de fonctionnement peuvent être distingués: actif, réception, transmission et sommeil:

- Etat actif : La radio est allumée, mais elle n'est pas employée. En d'autres termes, le nœud capteur n'est ni en train de recevoir ni de transmettre. Cet état provoque une perte d'énergie suite à l'écoute inutile du canal de transmission. Pour éviter cette perte d'énergie, un capteur ne doit s'activer qu'en cas de nécessité, et le reste du temps il doit se mettre dans l'état sommeil.
- Etat sommeil : la radio est mise hors tension.
- Etat de transmission : la radio transmet un paquet.
- Etat de réception : la radio reçoit un paquet.

Dans la plupart des cas, la consommation d'énergie est relativement élevée dans le mode actif, puisque ce dernier nécessite que le module radio soit mis sous tension et décode

continuellement les signaux radio afin d'apercevoir l'arrivée des paquets. Comme première solution, mettre le module radio en état de sommeil. Le passage fréquent de l'état actif à l'état sommeil peut avoir comme conséquence : une consommation d'énergie plus importante que de laisser le module radio en mode actif. Cela est dû à la puissance nécessaire pour la mise sous tension du module radio. Cette énergie est appelée l'énergie de transition. Il est ainsi souhaitable d'arrêter complètement la radio, plutôt que de transiter dans le mode sommeil. Le changement d'état du module radio doit être géré par un protocole de la couche MAC [31].

2. L'accès au médium de transmission

Le bon fonctionnement du réseau dépend, dans une large mesure, sur l'efficacité du contrôle d'accès au médium de transmission : protocole MAC utilisé. Ce dernier assure l'accès au support de transmission, le contrôle de flux, la fiabilité de la communication, la détection des erreurs et la retransmission des paquets en cas de collisions. En effet, la couche MAC essaye de faire en sorte que les transmissions de deux nœuds ne seront pas interférées les unes avec les autres et par conséquent coordonner les transmissions des différents capteurs pour minimiser les collisions et réduire la perte d'énergie [26].

Les principales causes de perte d'énergie au niveau de la couche MAC sont [32, 26] :

- ***La retransmission due à la collision ou la congestion*** : la communication dans les réseaux de capteurs est généralement par diffusion radio. Ce mode de communication est caractérisé par sa non fiabilité : si plusieurs capteurs émettent de façon simultanée il y a un risque de collision et aucun paquet ne pourra être reçu correctement (les paquets seront corrompus). Afin d'assurer une transmission robuste, un nœud capteur ayant détecté une collision doit retransmettre son paquet de données. En raison de la capacité limitée du canal sans fil, le réseau peut courir un risque de congestion si la quantité de messages échangés entre les noeuds est très élevée. Dans ce cas, une retransmission des données perdues est inévitable.
- ***L'écoute active*** : un nœud ne sait pas quand il sera le destinataire d'un message envoyé par l'un de ses voisins, pour cela il doit tenir sa radio en mode de réception tout le temps. Ceci est connu sous le phénomène de l'écoute active (idle listening) du canal. Pour cette raison, une énergie supplémentaire est perdue. Cette énergie consacrée à l'écoute d'un canal vide (dans les réseaux de capteurs le canal est libre la plupart du temps) est considérable par rapport à l'énergie consommée dans la réception ou l'émission des données.
- ***La sur-écoute (overhearing)*** : les données transmises par un nœud capteur peuvent atteindre d'autres nœuds qui se trouvent dans sa portée de transmission. Ceci est due au partage du médium de communication sans fil (tous les noeuds communiquent via un médium commun). Un nœud peut alors recevoir des paquets qui ne lui sont pas destinés.
- ***La surémission (overmitting)*** : le phénomène de surémission se produit quand un capteur envoie les données à un destinataire qui n'est pas prêt à les recevoir. En effet, ces messages envoyés sont considérés inutiles et consomment une énergie additionnelle.
- ***La surcharge (overhead)*** : plusieurs protocoles de la couche MAC fonctionnent par échange de messages de contrôle, dans le but est d'assurer différentes fonctionnalités tels que: la signalisation, la connectivité et l'établissement de plan d'accès pour éviter les collisions. Tous ces messages échangés nécessitent une énergie additionnelle. Par

exemple, les trames de contrôle RTS/CTS (Request To Send/ Clear To Send) ne véhiculent aucune informations alors que leur transmission consomme de l'énergie.

- ***La taille des paquets*** : La taille des messages échangés dans le réseau a un effet sur la consommation d'énergie des nœuds émetteurs et récepteurs. Ainsi, la taille des paquets ne doit être ni trop grande ni trop petite. Si la taille est petite, le nombre de paquets de contrôle (acquittements) généré augmente l'overhead. Dans le cas contraire, une grande puissance de transmission est nécessaire pour transmettre ces paquets de grande taille.

3. La classe d'application

Les applications dans les réseaux de capteurs sont classées en quatre catégories : orientées temps (time driven), orientées événements (event driven), orientées requêtes (query driven) et hybride [22]. Le type d'application a un impact direct sur la fonction de capture, le taux des données transmises et la topologie du réseau, et par conséquent, sur la durée de vie totale du réseau. Par exemple, dans un réseau time driven approprié à des applications nécessitant un prélèvement périodique de données, si les durées inter-transmission sont élevées, dans ce cas, la consommation d'énergie est minimale. Ce prélèvement périodique de données est inadapté pour les applications event driven qui sont généralement des applications temps réel, où les capteurs envoient leurs données seulement si un événement spécifique se produit. Pour cela, les capteurs doivent réagir et donner des réponses rapides à l'occurrence de cet événement. Ce type de scénarios exige que les noeuds contrôlent leurs radios continuellement, ce qui consomme de l'énergie. De plus, les alarmes qui sont remontées à la station de base doivent respecter certains délais de transmission.

Alors que dans un scénario orienté requête, un capteur n'envoie l'information que suite à une demande explicite de la station de base. Dans ce cas, la consommation d'énergie est liée aux exigences de l'utilisateur (délais de réponse, taux de transfert et nombre de requêtes).

4. Routage de données

Les réseaux de capteurs sont basés sur un routage multi-sauts. L'acheminement des paquets d'une source donnée à une destination se fait à travers plusieurs nœuds intermédiaires. Ainsi, un nœud consomme l'énergie soit pour transmettre ses données ou pour relayer les données des autres nœuds. Une mauvaise politique de routage peut avoir des conséquences graves sur la durée de vie du réseau. C'est pour cela, l'énergie est considérée comme ressource très précieuse dans les réseaux de capteurs dont les protocoles de routage doivent en tenir compte. Plus de détail sur le routage sera donné dans la section 2.3, où on va présenter les contraintes qui interviennent dans la conception des protocoles de routage ainsi que les différentes techniques utilisées pour assurer un routage à basse consommation d'énergie afin de tirer profit de leurs avantages et voire leurs limitations.

2.2.3 Taxonomies d'approches de conservation d'énergie dans les réseaux de capteurs sans fi

Dans les réseaux de capteurs, la consommation d'énergie est une contrainte très importante puisque généralement les capteurs sont déployés dans des zones inaccessibles. Ainsi, il est difficile voire impossible de remplacer les batteries après leur épuisement.

Plusieurs approches dans la littérature ont été proposées pour diminuer la consommation d'énergie et augmenter la durée de vie du réseau. Ces techniques d'économie d'énergie sont

prises en compte durant toutes les étapes de la conception et de la mise en œuvre d'un réseau de capteurs.

Dans le paragraphe suivant, on va citer quelques mécanismes (voir figure 2.1) visant à économiser la consommation d'énergie d'un réseau de capteurs au niveau: capture, traitement et communication.

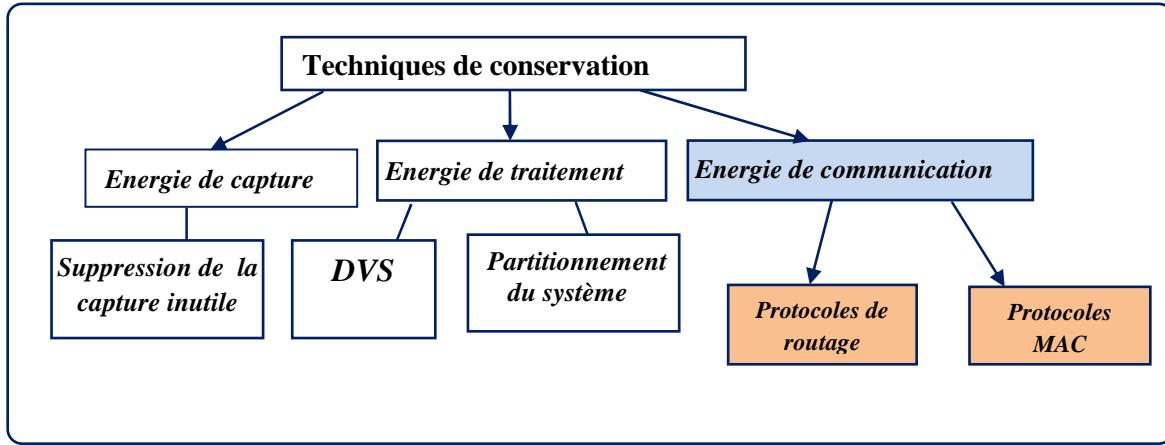


Fig.2.1 Techniques de conservation d'énergie.

- **Energie de capture** : la tâche de capture est effectuée par les composants d'acquisition qui traduisent les phénomènes physiques en signal électrique. L'énergie consommée lors de la capture peut être réduite en utilisant des composants à faible consommation mais en réduisant ainsi leurs performances. Alors, la solution la plus envisageable consiste à réduire les durées de capture et donc supprimer les captures jugées redondantes et inutiles [21].

- **Energie de traitement** : le traitement inclut le contrôle des composants de capture et l'exécution des protocoles de communication sur les données collectées. Les microprocesseurs sont à la charge de ce traitement. Le choix de ces derniers est en fonction du scénario de l'application et la consommation d'énergie souhaitée, et en général un compromis est réalisé entre le niveau de performance et la consommation d'énergie [2].

Pour conserver l'énergie consommée par un capteur lors du traitement de données, deux approches sont utilisées pour cette finalité :

- L'approche DVS (Dynamic Voltage Scaling) : consiste à ajuster de manière adaptative selon la charge de travail, la tension d'alimentation et la fréquence du microprocesseur afin d'économiser la puissance de traitement sans dégrader les performances [35].
- L'approche partitionnement du système : consiste à transférer le traitement vers la station de base qui est généralement non contraignante en termes de ressources énergétiques et possède une grande capacité de calcul.

- **Energie de communication** : l'énergie de communication représente la plus grande quantité d'énergie totale consommée au niveau d'un noeud capteur. La communication est principalement liée aux protocoles développés pour la couche liaison et la couche réseau. Pour minimiser l'énergie consommée par la communication, on doit intervenir au niveau de ces deux couches. Pour ce faire, plusieurs approches de communication avec conservation d'énergie ont été proposées pour la couche réseau ainsi que pour la couche liaison et en particulier pour la sous couche MAC [21]. Dans le reste de cette section, nous présentons

quelques exemples d'approches utilisées au niveau de ces deux couches afin d'optimiser l'énergie de communication.

a) La couche MAC: Medium Access Control

Les protocoles MAC dédiés aux réseaux de capteurs devraient être efficaces en terme d'énergie, stables lorsque la taille du réseau augmente et adaptatifs aux changements de la topologie et de la connectivité du réseau lorsque les capteurs cessent de fonctionner, ou de se déplacer en cas de présence de mobilité.

Pour réduire la consommation d'énergie, les protocoles MAC doivent permettre de garder la radio éteinte le plus longtemps possible. Pour cela, les messages de contrôles sont allégés ou carrément annulés et plusieurs techniques permettant la mise en veille des nœuds à des temps précis sont utilisées. Le cas parfait est de n'allumer la radio que pour recevoir et émettre les paquets de données utiles. En pratique cela est impossible car les nœuds ne communiquent que via leurs radios [21].

On peut distinguer principalement deux approches MAC: les protocoles ordonnancés et les protocoles de contention.

Les protocoles ordonnancés

Ces protocoles adoptent des techniques de synchronisation entre les nœuds du réseau pour éviter les collisions et l'overhearing, ce qui les rend très efficaces en consommation d'énergie. De plus, ils sont bien adaptés au routage hiérarchique (voir § 2.3.2.1) parce qu'ils exigent généralement aux nœuds de former des clusters et ne permettent pas les communications pair à pair.

Pour effectuer les communications inter-cluster et intra-cluster entre les différents noeuds du réseau les approches: TDMA, FDMA et CDMA [36, 37] peuvent être utilisées.

L'adoption de FDMA (*Frequency Division Multiple Access*) pour les communications intra-cluster évite les collisions car le canal est divisé en plages de fréquences allouées de manière équilibrée aux nœuds communicants. Alors que dans TDMA, le temps est divisé en trames et chaque trame se compose d'un certain nombre de slots de temps. À chaque noeud est attribué un slot par trame durant lequel il peut émettre/recevoir des paquets de/vers d'autres noeuds. De même, CDMA évite les collisions par l'utilisation d'un code pseudo aléatoire. Ainsi, des nœuds peuvent émettre leurs données continuellement, mais le calcul de ces codes consomme suffisamment de l'énergie.

Cependant, ces approches sont complexes, peu flexibles et présentent des problèmes lors du passage à l'échelle et l'adaptabilité aux changements de la topologie. En outre, ils requièrent une synchronisation très fine et ils sont très sensibles aux interférences.

Pour remédier aux limitations présentes dans ces protocoles MAC, d'autres techniques ont été proposées dans cette direction.

SMACS (Self-organizing Medium Access Control for Sensor Networks) [38] est un protocole de scheduling proposé pour organiser le réseau en synchronisant les noeuds. Il implique une méthode d'accès au canal combinant TDMA et FDMA, dans laquelle les nœuds voisins choisissent aléatoirement un slot et une fréquence qui définit un lien. Tous les nœuds s'allument en même temps, s'envoient les paquets nécessaires à maintenir leur synchronisation puis envoient leurs données. Ce protocole permet donc d'éteindre la radio régulièrement mais ça n'est peut être pas encore suffisant s'il y a très peu de trafic. Dans ce cas, les nœuds s'allument périodiquement pour n'échanger que des paquets de contrôle afin de rester

synchronisés, mais n'envoient pas de données. Le nombre de paquets émis contenant de l'information utile pour l'application devient faible par rapport au nombre total de paquets émis.

EAR (Eavesdrop And Register) [21] est un autre protocole proposé pour cette famille. Ce protocole intègre les nœuds mobiles dans son mécanisme en introduisant une table de voisins stockée au niveau de chaque nœud, et en échangeant des messages entre ces noeuds pour mettre à jour la topologie du réseau.

Ces deux protocoles sont très utiles, lorsque la transmission périodique de l'information exige la maintenance continue du réseau. Cependant, ils exigent un overhead important pour construire la structure. De plus, la bande passante n'est pas bien exploitée et le passage de l'état actif /sommeil est très coûteux en consommation d'énergie (voir § 2.2.2).

Protocoles de contention

Les protocoles basés sur la contention sont les plus populaires et représentent la majorité des protocoles MAC proposés pour les réseaux de capteurs sans fil. Ces protocoles sont plus flexibles aux changements de la topologie. Ils assurent la communication pair-à-pair et n'exigent aucune synchronisation. Néanmoins, ils n'utilisent pas souvent efficacement les ressources à cause des collisions et de l'écoute inutile. Dans la suite de ce paragraphe, quelques protocoles de cette famille sont présentés.

Le protocole S-MAC (Sensor MAC) [39] utilise le CSMA/CA comme méthode d'accès au médium avec RTS/CTS (Request-To-Send, Clear-To-Send) pour éviter les collisions et le problème du nœud caché. Ce protocole instaure un mécanisme de mise en veille distribué à chaque nœud, dans le but de réduire sa consommation d'énergie et prolonger sa durée de vie. Chaque nœud devrait coordonner et échanger des informations avec ses voisins pour choisir son propre cycle "Sommeil/Actif ". Pour réaliser une synchronisation entre ces noeuds, S-MAC permet à chacun des nœuds d'envoyer un paquet SYNC pour générer un ordonnancement à tous ses voisins. S-MAC est particulièrement conçu pour les systèmes d'alerte dans lesquels les applications ont de longues périodes d'inactivité et peuvent tolérer la latence. Son but principal est de maximiser la durée de vie du réseau en dépit des autres critères de performances. Toutefois, l'utilisation intensive du cycle sommeil augmente considérablement la latence, car il est difficile de synchroniser les nœuds de telle sorte que la communication soit toujours possible. En outre, les cycles "Sommeil/Actif " sont constants et ne peuvent pas être modifiés après le déploiement.

T-MAC (Timeout MAC) [40] est une version améliorée de S-MAC, conçu spécialement pour une charge de trafic variable. Dans T-MAC, chaque nœud se réveille périodiquement pour communiquer avec ses voisins en utilisant le mécanisme RTS/CTS. Ainsi, le nœud écoute le canal radio et il ne peut transmettre que s'il est dans la période active. Cette période se termine si au bout d'un certain temps TA, aucun événement d'activation ne se produit, telle que la détection d'une communication sur le canal ou le déclenchement d'un temporisateur de frame. Cependant, T-MAC souffre de la latence du sommeil, où un noeud doit attendre que le récepteur se réveille avant qu'il puisse acheminer un paquet. Cette latence augmente avec le nombre de sauts qui le sépare du noeud récepteur. En outre, les performances de ce protocole se dégradent quand le trafic est unidirectionnel, par exemple seulement les capteurs remontent leurs données collectées au nœud puits. Les données sont généralement transmises par diffusion vers le nœud puits ce qui engendre un problème d'interruption du à la sensibilité de la

radio qui limite la portée de l'overhearing. Les noeuds en dehors de la portée de l'émetteur ne peuvent donc pas entendre la transmission en cours et se rendorment.

B-MAC (Berkeley MAC) [73] a été développé par l'université de Berkeley pour les capteurs compatibles ZigBee, assurant une faible complexité et une faible consommation induite par le système d'exploitation TinyOS. B-MAC se base sur deux principes : l'analyse du bruit sur le canal et l'écoute à basse consommation d'énergie. Ainsi, un nœud voulant transmettre des données devrait écouter le canal. Si le canal est libre, il devra émettre au début un préambule. Les nœuds sont souvent dans l'état "sommeil" et ils se réveilleront à des intervalles réguliers. A leur réveil, ils écouteront le canal s'il y a du bruit, cela signifie qu'il y a des données qui vont arriver sinon ils retourneront à l'état sommeil. Le principal avantage du B-MAC est qu'il ne nécessite pas de synchronisation entre les nœuds. Cependant, l'utilisation de préambule avant tout envoi pourrait avoir des conséquences sur la consommation d'énergie et la quantité des messages échangés dans le réseau.

b) La couche réseau

Pour prolonger la durée de vie d'un réseau de capteurs, les protocoles de routage au niveau de la couche réseau doivent être conçus de manière à présenter de bonnes performances sur le plan du choix des routes ainsi que sur celui de leur coût de fonctionnement, en particulier en émettant le moins de paquets possibles. Pour cela, plusieurs techniques sont proposées afin de conserver la consommation d'énergie : l'agrégation de données, le routage adaptatif, le clustering, le contrôle de la puissance de transmission...etc. Ces techniques et bien d'autres types de solutions seront exposés dans le point suivant qui étudie en détail le routage dans les réseaux de capteurs sans fil.

2.3 Le routage dans les réseaux de capteurs

Le routage constitue un service important dans les réseaux de capteurs. Il permet le transfert des paquets de données à partir des nœuds capteurs vers le nœud puits. Ces nœuds sont déployés dans la région à surveiller d'une manière très dense, ce qui les rend très proches les uns des autres. Pour ce genre de situations, plusieurs techniques de communication sont utilisées. De plus, ces nœuds doivent consommer moins d'énergie en transmettant les données vers le noeud puits. Par conséquent, l'utilisation des protocoles de routage conçus pour les réseaux ad hoc traditionnels est inappropriée pour ces réseaux de capteurs. Ceci, est du principalement aux caractéristiques qui distinguent ces deux types de réseaux. Il a donc fallu concevoir de nouveaux protocoles de routage spécifiques aux réseaux de capteurs. Pour cela, de nombreux travaux ont été effectués, et plusieurs protocoles ont été proposés pour permettre un routage à basse consommation d'énergie. Par ailleurs, ces protocoles sont influencés par de multiples contraintes dans leur conception. Il est à noter que plusieurs classifications ont été dédiées à ces protocoles. Tous ces points, seront détaillés dans les sections suivantes.

2.3.1 Contraintes à tenir en compte lors de la conception des protocoles de routage

L'un des défis de conception le plus important dans les réseaux de capteurs consiste à assurer le bon fonctionnement du réseau tout en essayant d'augmenter sa durée de vie. Ainsi, l'efficacité des protocoles de routage visés à ce type de réseaux dépend essentiellement de plusieurs facteurs [1, 22, 21].

Ces protocoles doivent impérativement utiliser le minimum de ressources énergétiques de façon à assurer une vie plus longue pour le réseau car les noeuds sont des micro-capteurs dotés de ressources limitées. Offrir une bonne gestion par ces protocoles devrait exploiter positivement les ressources de ces nœuds en énergie, capacité de stockage et de calcul.

D'autre part, ces protocoles doivent prendre en considération la nature de la topologie. Cette dernière peut être déterministe ou bien aléatoire.

Dans le cas d'une topologie déterministe, les noeuds sont distribués manuellement et le routage est prédéfini. Par contre dans une topologie aléatoire, les nœuds doivent s'auto-organiser afin de construire une structure virtuelle nécessaire pour le routage.

Un autre facteur très important est le type d'application. La nature du protocole dépend de l'application en question. Par exemple, dans des applications soumises à contraintes de temps, les données doivent être fournies au cours de certaines périodes du moment où elles sont capturées, sinon elles sont inutiles. Par conséquent, la latence devrait être soigneusement définie lors de la livraison des données.

La tolérance aux pannes est un autre facteur à prendre en compte pendant cette phase de conception. Les protocoles proposés doivent être capables de s'adapter à la défaillance des nœuds et le changement de la topologie, en maintenant par exemple des chemins alternatifs pour router les données en cas de défaillance, ce qui permet de garantir le fonctionnement du réseau et la connectivité des nœuds.

En outre, la mobilité de la station de base ou des nœuds capteurs intervient aussi lors de la conception de ces protocoles. Par exemple, un réseau avec une station de base fixe qui reçoit les données à partir d'un noeud distant nécessite des protocoles de routage différents de celui d'un réseau ayant une station de base mobile.

La forte densité qui caractérise les réseaux de capteurs engendre une grande redondance dans les données circulant à travers le réseau, ce qui entraîne une consommation excessive de l'énergie pour transmettre ces informations redondantes. De ce fait, l'agrégation des données est la meilleure solution à considérer pour remédier à cette contrainte. L'agrégation peut être effectuée par suppression de données redondantes ou par fusion (en prenant le maximum, le minimum, ou bien la moyenne) [21]. Dans le cas de la fusion, le nœud reçoit les messages de ses voisins et les combine en un seul message selon la technique choisie. Ceci réduit considérablement la quantité des messages échangés entre les noeuds, mais engendre une latence accrue car le nœud doit attendre les messages de ses voisins (expiration d'un timeout par exemple) avant d'envoyer les données. Lors de la conception d'un protocole de routage, il faut prévoir des techniques d'agrégation de donnée et éventuellement choisir des nœuds concernés par cette opération. A titre d'exemple, les approches de clustering utilisent largement ces techniques, en confiant la tâche d'agrégation seulement aux nœuds cluster-heads.

2.3.2 Classification des protocoles de routage

Dans les réseaux sans fil, trois grandes familles d'approches de routage peuvent être distinguées : protocoles proactifs, protocoles réactifs et protocoles hybrides. Les protocoles proactifs anticipent la demande d'acheminement des paquets, où chaque nœud maintient les informations concernant tous les autres nœuds du réseau dans des tables de routage qui sont mis à jour périodiquement. Alors que, les protocoles réactifs réagissent à une demande d'acheminement en calculant la route suite à cette requête. En combinant ces deux approches,

on trouve la famille des protocoles hybrides qui possèdent des propriétés à la fois de proactivité et de réactivité, de telle sorte que le réseau est décomposé en un ensemble de zones. Le routage à l'intérieur des zones est assuré par un protocole de routage proactif, tandis que le routage entre les zones est réalisé par un protocole de routage réactif.

Pour les réseaux de capteurs, plusieurs approches de routage ont été proposées. Les protocoles publiés dans ce domaine ont été largement étudiés et peuvent être classifiés selon plusieurs critères. Al-Karaki et al [1] ont proposé une classification (la plus cité dans la littérature) à ces différents protocoles. Ces derniers, sont classés selon leur principe de fonctionnement et la structure du réseau (voir figure 2.2).

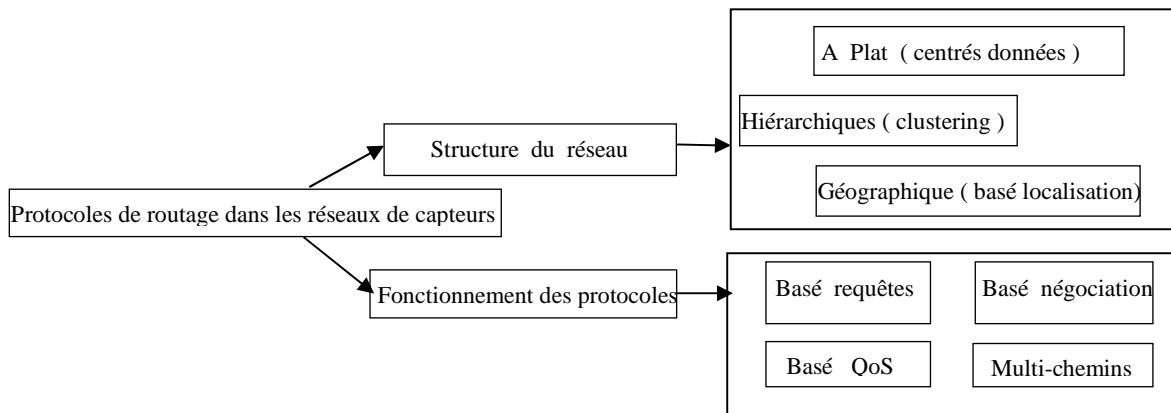


Fig.2.2 Classification de protocoles de routage.

2.3.2.1 Routage basé sur la structure du réseau

Cette section décrit différents protocoles de routage basés sur la topologie du réseau et les informations d'emplacements géographiques des capteurs. Ces protocoles sont divisés en trois classes : routage à plat et centré données, routage hiérarchique (basé clustering) et routage géographique ou basé localisation. Plus de détails pour les mécanismes et les protocoles, seront apportés à la classification basée sur la structure du réseau car une grande quantité de travaux a été publié dans cette classe.

1. Routage à plat et centré donnée

Dans un réseau à structure plate, tous les noeuds jouent le même rôle et coopèrent ensemble pour réaliser la tâche du routage. En raison du grand nombre de nœuds capteurs, il n'est pas faisable d'assigner un identifiant global à chaque nœud (tel que l'adressage IP), due à la difficulté d'obtention et de gestion d'adresses qui nécessitent généralement une étape de configuration qui peut être complexe. De plus, étant donné la nature des applications des réseaux de capteurs, où l'utilisateur ne s'intéresse pas à communiquer avec un nœud particulier dans le réseau, mais se concentre plutôt sur sa donnée. Dans ce cas la donnée est plus importante que le nœud lui-même et toutes les communications sont identifiées par leurs données. Il est donc nécessaire de concevoir une approche de routage pour répondre à ces exigences. D'où l'apparition des protocoles de routage centrés-données (Data Centric) [74]. Dans ce type de protocoles, le routage ne se fait pas en fonction d'une adresse de destination, mais suivant les données disponibles au niveau des capteurs.

Dans le reste de cette section, nous donnerons un résumé de quelques protocoles de cette famille et nous mettrons l'accent sur leurs avantages, limites et performances.

a) Exemple de protocoles de cette famille

Parmi les protocoles publiés dans la littérature selon le routage à plat et centré données ont peut citer : Direct Diffusion [75], Rumor Routing [76]. Les caractéristiques, ainsi que le principe de fonctionnement de chacun de ces protocoles vont être présentés dans ce qui suit.

Diffusion dirigée Ou Direct Diffusion (DD)

Le protocole diffusion dirigée proposé par Intanagonwiwat et al [1, 21, 75] est considéré comme une référence dans le routage centré-donnée. C'est un protocole réactif qui est simple dans son principe. Il repose sur quatre phases dans son fonctionnement : propagation d'intérêts, propagation des gradients, propagation des données et renforcement des routes (chemins).

-*Propagation des intérêts* : DD procède à une nomination de données pour les informations générées par les noeuds capteurs. Ces dernières sont décrites par des paires attribut-valeur. Ces attributs servent par exemple, à détailler le type de phénomène (donné ou variable physique) à capturer, sa position géographique, sa durée, etc. Par la suite, le nœud puits commence à diffuser vers tous les nœuds un message de type intérêt sous forme de requêtes, pour interroger le réseau sur une donnée bien déterminée. L'intérêt est basé sur la représentation attribut-valeur afin de designer les données requises et le débit de transmission vers le nœud puits.

-*Etablissement des gradients* : chaque noeud maintient localement un cache d'intérêt, contenant des informations pour décrire les intérêts et les gradients. Un gradient est un lien de réponse construit par le voisin recevant le message intérêt. Il est caractérisé par le taux de transfert de données, la durée et le temps d'expiration. De plus, le capteur recevant le message intérêt, ajoute dans son cache une entrée pour l'intérêt reçu si cette entrée n'existe pas déjà.

Après la mise à jour de la table d'intérêt, le noeud rediffuse l'intérêt reçu à tous ses voisins.

-*Propagation des données* : une fois les requêtes contenant les intérêts atteignent les nœuds sources destinataires, les nœuds capteurs commencent la récolte d'informations selon le débit le plus élevé de chaque intérêt. Pour chaque intérêt, les données sont transmises vers tous les voisins figurant dans l'ensemble des gradients relatifs à cet intérêt (chaque voisin reçoit les données avec son propre débit). Lorsqu'un nœud reçoit une donnée, il consulte sa table d'intérêt. Si aucune entrée relative à l'intérêt n'existe, la donnée est automatiquement supprimée. Sinon, le nœud vérifie si la donnée figure dans le cache des données récemment émises. Si elle n'y est pas, le nœud retransmet la donnée aux voisins vers lesquels des gradients ont été établis suivant le débit de chacun d'eux et ajoute la donnée à son cache de données. L'utilisation du cache de données permet de garder une trace des chemins vers les sources car la phase d'établissement des gradients ne crée que les routes vers le puits.

-*Renforcement des chemins* : durant les phases précédentes, plusieurs chemins peuvent être établis entre le noeud puits et la source. L'un de ces chemins est sélectionné par renforcement positif, afin d'être utilisé plusieurs fois. Une fois le noeud puits reçoit les premières données, il doit renforcer le chemin vers le voisin émetteur, en lui réclamant les mêmes données avec un taux plus élevé. Le renforcement doit se propager éventuellement jusqu'aux sources.

Par ailleurs, si ce chemin échoue un nouveau ou un alternatif doit être identifié. Pour ce faire, le protocole réinitialise une diffusion des paires attributs-valeurs afin de trouver d'autres routes, mais les données seront transmises dans ce cas à des débits plus faibles. Ceci est appelé renforcement négatif. Il est initialisé par le noeud puits quand ce dernier reçoit un faible taux de

transfert d'événements à partir d'un nœud voisin. Le nœud puits envoi un message de renforcement négatif à ce nœud voisin. De la même façon, le nœud voisin du nœud puits envoi ce message à un de ses voisins jusqu'à ce que le message arrive au nœud source d'événements. Les nœuds qui reçoivent le message de renforcement négatif arrêtent leur transmission des données vers les voisins émetteurs de ce message.

Le protocole Direct Diffusion présente de nombreux avantages en termes d'efficacité énergétique et type de communication adoptée (il évite l'adressage global dans le réseau, en utilisant une communication voisins-voisins). Ainsi, l'utilisation du cache pour stocker les données permet à ce protocole de minimiser la consommation d'énergie et les délais de transmission. D'autre part, l'utilisation d'un renforcement négatif permet d'éviter les routes qui engendrent des délais. En outre, les requêtes par diffusion directe sont également efficaces en consommation énergétique puisqu'il n'y a aucun besoin de maintenir la topologie globale du réseau.

Toutefois, Direct Diffusion ne présente pas un bon choix comme protocole de routage pour les applications de surveillance, tels que le contrôle de l'environnement et les processus industriels car DD est basé sur un modèle à base d'interrogation pour la délivrance de données qui ne fonctionne pas efficacement avec la livraison continue ou périodique des données au noeud puits. En outre, le critère de choix utilisé pour le renforcement positif des chemins influe sur ses performances (selon ce protocole, le renforcement est basé sur le taux de transfert). De plus, l'utilisation de l'inondation pour la propagation des intérêts et l'envoi de données consomme considérablement l'énergie des nœuds et surcharge le réseau.

Rumor Routing

Rumor Routing [1, 76] proposé par D.Braginsky et D.Estrin est une variante de la diffusion dirigée. Il est conçu pour les réseaux dans lesquels le routage géographique (type de routage décrit plus tard) n'est pas applicable. A la différence de la diffusion dirigée qui utilise une inondation totale de requêtes dans le réseau, même dans le cas où seulement une faible quantité de données est nécessaire, Rumor Routing essaye de trouver un compromis entre l'inondation des intérêts et la propagation des données. Ce protocole opte vers l'idée d'inondation par les évènements qui se produisent (dans le cas où le nombre d'évènements est petit et le nombre de requêtes est grand). Par conséquent, plutôt que d'inonder le réseau en entier, ce protocole transmet les requêtes aux nœuds qui ont observé un évènement particulier pour rechercher des informations sur les occurrences des évènements qui se sont produisent.

Dans le but de diffuser des événements à travers le réseau, Rumor Routing emploi des paquets à longue vie nommé : agent. Quand un nœud détecte un événement il l'ajoute à sa mémoire locale (cache, table d'événement) et génère un agent. L'agent circule dans le réseau pour disséminer l'information à propos de l'événement local vers des nœuds distants. D'ailleurs, lorsqu'un nœud génère une requête pour un événement, un nœud qui connaît le chemin peut répondre à la requête en se référant à sa table d'événement. Par conséquent le coût d'inondation est évité. L'algorithme maintient un seul chemin entre la source et la destination contrairement à Direct diffusion où les données sont envoyées par plusieurs chemins avec un débit réduit.

Avec Rumor Routing, le coût d'inondation du réseau en entier est évité. Ceci se traduit, par les résultats de simulation qui ont montré que le routage avec ce protocole permet de réaliser des économies énergétiques significatives par rapport aux autres protocoles de diffusion. Encore, dans la diffusion dirigée, les données peuvent être envoyées par des routes multiples à bas débit alors que, le principe utilisé par Rumor Routing est différent. Il lui permet de maintenir uniquement une route entre la source et la destination.

Néanmoins, ce type de routage n'est efficace que lorsque le nombre d'évènements est petit. Pour un grand nombre d'évènements, le coût de maintien des agents et des tables d'évènements au niveau de chaque nœud est considérable et en particulier dans le cas où le nœud destinataire n'est pas intéressé par ces évènements (il n'y a pas assez d'intérêt envers ses événements).

b) Avantages et inconvénients du routage à plat

Selon l'architecture plate du réseau, tous les noeuds sont au même niveau et peuvent participer de la même manière dans le routage de données. Le routage à plat est très simple dans son fonctionnement et les protocoles à plat ne nécessitent pas beaucoup de communications pour établir les routes. Ces dernières, sont généralement formées uniquement dans les régions où il y a des données à transmettre.

Les protocoles à plat offrent la possibilité d'agrégation des données. Cette opération s'effectue par les nœuds capteurs qui se trouvent sur le chemin du routage vers la destination, où les données des autres noeuds voisins sont accumulées avec une élimination de redondances.

Néanmoins, les nœuds proches de la station de base vont participer plus que les autres, et donc vont épuiser leur énergie beaucoup plus rapidement que les autres nœuds capteurs. Ce phénomène est connu sous le nom de points chauds (Hot-Spot). Cela est du au fait, que le trafic de tous le réseau est concentré sur ces noeuds. D'autre part, le fait d'adopter un routage multi-sauts où tous les nœuds participent dans la tâche du routage engendre des délais considérables causés par l'activation des noeuds en mode veille et l'établissement de ces routes.

Quoique la topologie plate assure une grande tolérance aux pannes, la faible scalabilité est l'une des ses limites. En effet, le routage à plat exige que tous les nœuds opèrent de la même façon et d'une manière distribuée, ce qui nécessite un grand nombre de messages de contrôle pour le bon fonctionnement des protocoles et la maintenance de la topologie.

2. Routage hiérarchique

Pour remédier aux problèmes du routage à plat, cette approche a été proposée afin de maintenir la conservation d'énergie au niveau des nœuds et essayer de prolonger la durée de vie du réseau. D'ailleurs, ce routage est considéré comme étant l'approche la plus favorable en termes d'efficacité énergétique et d'agrégation de données. Les protocoles selon cette approche procèdent à un partitionnement du réseau en clusters. Dans chaque cluster, des cluster-heads sont désignés, soit par élection par les autres nœuds capteurs, sinon ils sont assignés par une autorité centralisée (nœud puits par exemple).

Une fois que le réseau est structuré, des méthodes de communication inter et intra-clusters sont définies. Généralement, les cluster-heads sont responsables de router les données collectées des noeuds membres à la station de base via un seul saut, ou par l'intermédiaire des autres cluster-heads, si nécessaire. La tâche d'un cluster-head ne consiste pas seulement à router les données, mais elle peut être complétée par une agrégation ou une fusion de ces données. De ce fait, des changements de rôle membre/cluster-head sont souvent appliqués pour répartir la charge du routage de données entre les nœuds du réseau.

a) Protocoles de cette famille

Dans cette section, un seul protocole hiérarchique est présenté. Le protocole choisi est considéré parmi les plus récents protocoles de clustering publiés dans la littérature. La raison de ce choix revient à ses caractéristiques et son mécanisme permettant de visionner les différentes étapes nécessaires pour structurer le réseau en clusters selon cette approche.

Le prochain chapitre est consacré à la présentations détaillée de l'approche clustering, où nous décrivons son principe, ses différentes techniques ainsi nous exposons un panorama de protocoles proposés selon cette famille comme par exemple LEACH[33], HEED[58], ACE[71], WCA[70]..ect.

PEBECS: Partition Energy Balanced and Efficient Clustering Scheme

Ce protocole [77] a été récemment proposé par Y.wang et D.zhang. Il essaye de répondre à deux questions pertinentes: comment choisir les cluster-heads (CHs) pour trouver un nombre optimal de clusters dans le réseau, et quelle est la bonne méthode pour assigner les noeuds non cluster-heads aux différents clusters.

PEBECS essaye d'apporter un partitionnement équilibré avec des tailles inégales pour les différents clusters générés dans le réseau. Pour ce faire, il emploie un processus qui permet de produire des clusters ayant des tailles différentes. La taille d'un cluster diminue en fonction de sa distance à la station de base : les clusters plus loin, on un éventail plus grand que ceux plus proches de cette dernière. Ceci permet d'avoir des tailles plus petites dans les partitions proches de la station de base, ce qui donne la possibilité de conserver plus d'énergie pour les communications inter-clusters. Ce protocole suppose que les noeuds dispersés dans la zone d'intérêt sont subdivisés en K partitions. Telle que K est un paramètre du système. Chaque partition comprend un certain nombre de clusters. Dans chaque cluster un cluster-head est élu. Les cluster-heads élus n'assurent pas les mêmes rôles. Par exemple, dans la k^{Emme} partition, les cluster-heads ne sont responsables que de la collecte des données de leurs noeuds membres pour les router, par la suite, vers leurs cluster-heads voisins. Alors que, dans les autres partitions ils ont en plus, les responsabilités d'agrégation de données des autres cluster-heads et de routage inter-clusters.

Pour structurer le réseau et construire les clusters, cet algorithme s'exécute sur deux phases : l'élection des cluster-heads et la formation des clusters.

Dans la phase d'élection de cluster-heads, un poids est attribué à chaque noeud. Les cluster-heads sont élus à l'aide de ce poids. Un noeud prend la décision de devenir cluster-head en fonction de certaines métriques combinées pour calculer le poids. Ce poids est dépendant des paramètres suivants : le seuil de consommation d'énergie, le degré du noeud, une métrique de sa localisation, ainsi que le nombre de noeuds dans sa partition.

Le niveau de consommation d'énergie permet l'élection de cluster-head avec plus d'énergie résiduelle. Un noeud disposant de plus d'énergie résiduelle aura une forte probabilité d'être choisi. Le second paramètre, permet d'assurer une distribution uniforme des clusters dans la partition afin d'équilibrer la charge des cluster-heads choisis en termes de noeuds à gérer. Alors que le troisième et le quatrième, sont généralement utilisés pour apporter plus de conservation d'énergie lors du choix des cluster-heads. En effet, le troisième paramètre permet de favoriser le noeud disposant de plus grand nombre liens de communication avec ses noeuds voisins. Alors que le quatrième, est utilisé pour localiser le noeud le plus proche/ loin des autres noeuds de partitions voisines. Eventuellement, le noeud disposant de plus proches voisins dans les partitions adjacentes aura une forte probabilité d'être élu cluster-head.

Dans la phase de formation de clusters, cet algorithme introduit un mécanisme pour résoudre le problème de hot spot : chaque noeud calcule son rayon de concurrence afin de

limiter le nombre de noeuds dans chaque cluster de façon à confier les clusters de petites tailles pour les cluster-heads proches de la station de base.

Le routage multi-saut est adopté comme mode de communication entre les différents cluster-heads lors de l'envoie des données collectées des autres cluster-heads à la station de base.

Cette approche présente des résultats de simulation remarquables en termes de conservation d'énergie pour les nœuds capteurs et par conséquent, l'augmentation de la durée de vie du réseau. Ceci est dû, d'une part, à la prise en charge des métriques liées à la consommation d'énergie, le degré d'un nœud, et la distance aux autres nœuds adjacents dans la fonction de calcul de poids lors de la phase d'élection de cluster-heads. D'autre part, la technique de formation de clusters adoptée par PEBECS permet de réaliser un équilibrage de la consommation d'énergie entre les cluster-heads (clusters plus proches de la station de base ont des tailles plus petites que ceux éloignés). De plus, à travers le mécanisme de routage multi-sauts introduit par les cluster-heads, moins d'énergie est consommée pour atteindre la station de base, ce qui offre de meilleures performances pour le réseau.

Cependant, les auteurs supposent que le réseau est subdivisé en k partitions sans donner des détails sur la façon de choisir ce paramètre k .

En outre, la quantité de messages de communication générée dans le réseau est très élevée car les cluster-heads relais ne sont responsables que du transfert des données vers la station de base ou vers d'autres cluster-heads, et donc aucune fonction d'agrégation de données n'est réalisée sur les données issues des autres cluster-heads (chaque cluster-head ne combine que les données collectées de ses nœuds membres), ce qui va augmenter le nombre de messages échangés dans le réseau. Comme solution possible : introduire des mécanismes d'agrégation de données au niveau des cluster-heads relais pour diminuer le nombre et la taille des messages transférés vers la station de base.

b) Avantages et inconvénients du routage hiérarchique (approche de clustering)

Le routage hiérarchique permet de remédier au problème de hot spot (points chauds) rencontré dans le routage à plat car les cluster-heads, selon cette approche, sont les seuls à envoyer les données vers la station de base. De même, plusieurs mécanismes ont été proposés pour le routage hiérarchique afin de résoudre complètement ce problème qui pourra à ce moment se rencontrer avec les cluster-heads proches de la station de base (voir la technique utilisée par le protocole PEBECS, en construisant les clusters avec des tailles inégales en fonction de la distance à la station de base). En effet, cette approche présente deux grands avantages: la scalabilité et la simplicité des mécanismes d'agrégation de données. L'agrégation de données est l'une des techniques largement utilisée par cette architecture. Cette tâche est confiée aux cluster-heads qui procèdent à l'agrégation/fusion des données collectées dans leurs clusters pour les envoyer par la suite vers la station de base. Ceci permet d'alléger le trafic circulant dans le réseau en diminuant le nombre de messages transmis à la station de base et en présentant ainsi, une manière efficace pour minimiser la consommation d'énergie dans le réseau et le passage à l'échelle.

Cependant, l'approche de clustering provoque une consommation déséquilibrée en énergie dans le réseau causée par la surcharge des cluster-heads, car ces derniers consomment plus d'énergie par rapport aux autres nœuds du réseau. Pour remédier à ce problème, des techniques de sélection dynamique sont adoptées pour le changement de rôle des nœuds cluster-heads/membres. Cette solution permet d'assurer un équilibrage de consommation d'énergie entre les nœuds du réseau. D'autre part, une autre solution qui nécessite des exigences matérielles pour les cluster-heads peut être utilisée. Selon cette solution, les cluster-heads sont des capteurs spécifiques avec plus de ressources énergétiques et plus de capacités de traitement.

3. Routage géographique (basé sur la localisation)

Avec ce type de routage, les informations de localisation des capteurs: positions géographiques, sont utilisées pour effectuer l'acheminement des données. Plusieurs solutions ont été proposées selon cette approche. Généralement, les méthodes employées supposent que chaque capteur a la connaissance exacte de sa position, soit grâce à la technologie GPS (Global Positioning System) dans le cas où les capteurs sont équipés par des récepteurs GPS de faible énergie et permettant ainsi la communication avec un satellite. Sinon, grâce à des méthodes de localisation basées sur l'estimation de la distance aux autres noeuds. Dans de telles méthodes, la localisation est assurée par une estimation de la distance séparant deux capteurs, en fonction des propriétés du signal reçu (temps de propagation, atténuation du signal, etc). Soit en fonction de l'angle d'arriver du message.

Dans ce qui suit, nous présentons un protocole de routage efficace en conservation énergétique et qui sera basé sur un système de localisation.

a) Exemple de protocoles de cette famille

Parmi les protocoles basés sur la localisation , on retrouve le protocole GAF [78]. Ce protocole est le plus répandu dans les applications utilisant des informations sur la position géographique des capteurs. Son principe et ses caractéristiques vont être détaillés dans la suite de ce paragraphe.

GAF: Geographic adaptive fidelity

Le protocole GAF [78] est un protocole de routage basé sur la localisation. Il était conçu comme approche de routage pour les réseaux Ad hoc, néanmoins il peut être applicable aux réseaux de capteurs. Pour conserver l'énergie dans le réseau, il utilise un mécanisme de mise en veille des nœuds non utilisés, sans affecter les performances du routage.

Ce protocole procède à une décomposition de la zone de capture en de petites cellules virtuelles de sorte que tous les nœuds dans une cellule peuvent communiquer ensemble. Ces cellules forment une grille virtuelle. Où chaque nœud, emploie sa position indiquée par le service GPS pour s'associer à un point de la grille. A tout moment, un seul nœud dans une cellule est actif à la fois, tandis que les autres sont en état sommeil ou en état de découverte. Telle que, les nœuds liés au même point sur la grille sont considérés équivalents en termes de coût lié au routage des paquets. Cette équivalence est exploitée en maintenant les nœuds situés dans le même secteur (au même point) de la grille, à l'état de sommeil.

La figure 2.3 illustre une situation simple du protocole GAF. Dans cette figure, le nœud 1 peut atteindre 2, 3 et 4. Les nœuds 2, 3 et 4 peuvent atteindre 5. Mais 2, 3 et 4 sont équivalents (en termes de consommation d'énergie durant le routage) donc deux d'entre eux seront mis en veille.

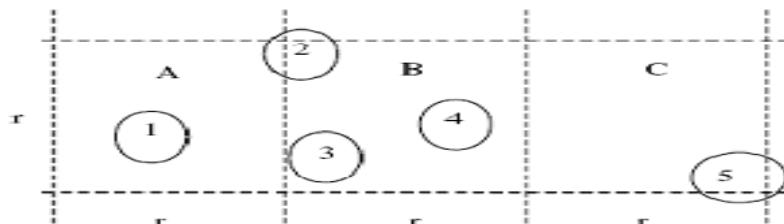


Fig.2.3 Exemple d'une grille virtuelle dans GAF

Selon GAF, l'état de découverte est l'état initial dans lequel un noeud identifie tous les nœuds qui se trouvent dans sa cellule en échangeant des messages de découverte. Le noeud qui participe dans la tache du routage est considéré comme étant un nœud dans l'état actif. Or, un nœud dans l'état de découverte ou dans l'état actif entre dans l'état sommeil, caractérisé par l'interface radio éteinte, s'il trouve qu'il y a déjà dans sa cellule un nœud actif qui tient la tâche du routage. Les nœuds équivalents peuvent basculer de l'état "Sommeil" vers l'état "actif" par tour de rôle. Généralement, les nœuds les plus riches en énergie sont plus prioritaires pour entrer dans l'état actif. De cette façon, la stratégie GAF équilibre la charge de la consommation d'énergie entre les différents nœuds du réseau.

Les auteurs montrent, par les résultats de simulation, que le gain en consommation d'énergie avec l'utilisation du protocole GAF augmente proportionnellement avec la densité des nœuds. De plus, ce protocole fournit des performances en termes de temps d'attente et perte de paquets.

Bien que, ce protocole soit basé sur la localisation, il peut être considéré comme un protocole de routage hiérarchique, où la formation des clusters (zones de la grille) est basée sur les positions des nœuds. Pour chaque zone particulière de la grille, le noeud actif agit comme étant un cluster-head responsable de la transmission des données vers d'autres noeuds du réseau.

Cependant, l'adaptation de ce protocole à un réseau de capteurs exige un modèle de consommation d'énergie approprié. En outre, GAF suppose que les nœuds doivent connaître leurs positions en utilisant un système GPS, qui est coûteux en énergie et peu précis.

a) Avantages et inconvénients du routage géographique

Le routage géographique utilise la position des nœuds comme moyen principal de routage et d'adressage de données. Il s'effectue grâce à des techniques géométriques (application des systèmes de positionnement, tel que le GPS) afin d'acheminer l'information d'une zone géographique vers une autre, ce qui permet de faciliter le contrôle de la topologie et la puissance de transmission entre les différents nœuds. De plus, ce mécanisme réduit considérablement le nombre de transmissions dans le réseau, parce que les informations sur les régions de capture sont disponibles, et donc la station de base peut envoyer directement la requête vers la région désirée. Ainsi l'espace de stockage nécessaire pendant le routage est très réduit car il ne nécessite aucune table de routage : le nœud achemine ses paquets seulement en se basant sur la position de la destination et sur des informations locales sur ses voisins directs. Un autre avantage important est l'indépendance de la mobilité. Cette approche est utilisable même si les nœuds changent de position d'une façon continue ce qui le rend adapté aux environnements mobiles et les applications militaires.

Cependant, ce type de routage ne convient pas aux applications qui exigent une livraison fiable de données à des intervalles réguliers (applications de surveillances, par exemple) vu les délais de transmissions élevés et le besoin d'informations de positionnement pour effectuer le routage. De plus, l'utilisation du système de positionnement GPS est inapproprié pour ce type d'applications : exigence d'équiper les capteurs par les récepteurs GPS, qui sont coûteux en énergie et peu précis ce qui diminue leur degré de fiabilité.

2.3.2.2 Routage basé sur le fonctionnement des protocole

Selon le fonctionnement et les avantages visés par les protocoles, quatre classes de routage peuvent être distinguées : routage basé sur la Qualité de Service (QoS ou QoS pour Quality Of Service), routage basé sur les multi-chemins, routage basé sur les requêtes et routage basé sur la

négociation de données. Dans cette section, les principales caractéristiques de ces différentes classes sont décrites, avec une brève présentation des principes et caractéristiques des protocoles exposés comme exemples dans chaque classe.

1. Routage basé sur la qualité de service

N'importe quel protocole de routage selon cette famille tend à satisfaire quelques métriques de qualité de service (QoS) comme par exemple : le délai d'acheminement de bout en bout, la bande passante et le taux d'erreurs, afin de réaliser un routage équilibré en termes de consommation d'énergie et délivrance de données en temps réel. Pour cela, le réseau doit s'équilibrer entre la consommation d'énergie et la qualité des données. Nous discutons quelques exemples de ces protocoles dans le paragraphe suivant.

a) Protocoles de cette famille

Dans ce qui suit, le principe de deux protocoles appartenant à cette famille va être brièvement présenté.

SAR : Sequential Assignment Routing

SAR [80] est considéré parmi les premiers protocoles élaborés pour les réseaux de capteurs. Il prend en considération la consommation d'énergie et la qualité de service lors de la prise de décision sur chaque route candidate pendant le processus de routage.

Une approche multi-chemins (classe détaillée plus tard) est utilisée pour éviter la défaillance d'une route. Cela est possible à travers la création d'un arbre avec multiples chemins reliant le nœud source au destinataire. Ainsi, à partir de chaque nœud jusqu'au nœud puits, plusieurs arbres d'une longueur d'un saut de ce nœud sont initialement construits. Chaque arbre est par la suite développé en allant successivement du nœud puits vers les autres nœuds, tout en évitant ceux qui ont les valeurs faibles de QoS et d'énergie disponible. A la fin de ce processus, chaque nœud fera partie des chemins multiples et pourra savoir le nœud voisin a utilisé pour transmettre les messages.

SAR dépend de trois facteurs : les ressources énergétiques acquises, la QoS dans chaque chemin, et le niveau de priorité des paquets. De ce fait, les chemins de l'arbre prennent en compte les métriques QoS, la ressource énergétique sur chaque chemin et le niveau de priorité de chaque paquet. Une ou plusieurs routes peuvent alors être empruntées.

Un processus de calcul de chemins est relancé périodiquement par le nœud puits afin de prendre en compte tout changement dans la topologie du réseau.

SAR présente plusieurs avantages. Il assure une bonne fiabilité du routage grâce à l'utilisation de plusieurs chemins plutôt qu'un chemin simple ce qui le rend très efficace pour la tolérance au pannes. De plus, la prise en compte des métriques de qualité de service et des ressources énergétiques dans le processus de routage (les routes sont sélectionnées en se basant sur ces deux métriques) augmente la performance du réseau.

Cependant, le maintien de multiples chemins entre la source et la destination impose une consommation importante en énergie et nécessite un envoi périodique de messages de contrôle engendrant ainsi, une grande quantité de messages échangés et donc surcharger le réseau.

SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks

Avec SPEED [81], chaque noeud maintient des informations de localisation sur ses voisins. Il utilise ces informations géographiques pour router les paquets au noeud puits. Sa caractéristique fondamentale est de garantir un délai de livraison de bout en bout optimale. Grâce à cette spécification, SPEED est le protocole le plus approprié pour les applications temps réel. En effet, ce protocole essaye de garantir une vitesse constante pour la transmission de paquets dans tout le réseau. Cette vitesse est estimée en divisant la distance entre la station de base par la vitesse du paquet. De plus, ce protocole offre la possibilité d'équilibrage de charge du trafic entre les noeuds du réseau. L'équilibrage de charge est réalisé via la technique adoptée par le module de routage appelé SNGF (Stateless Non-deterministic Geographic Forwarding) responsable de la sélection du prochain saut en offrant la vitesse de livraison de paquets souhaités. Ceci permet de disperser les paquets sur un domaine de routage plus vaste afin de répartir le trafic uniformément à travers tout le réseau.

SPEED propose une approche très utile pour les applications temps réel, mais son utilisation reste très limitée dans les réseaux de capteurs car il ne considère aucune métrique pour l'optimisation de la consommation d'énergie. Alors que l'énergie est le défi critique pour ce type de réseau, par conséquent, et pour une meilleure compréhension des capacités du protocole SPEED, en termes de conservation de la ressource énergétique, ce protocole a besoin d'être comparé avec d'autres protocoles de routage spécifiques aux réseaux de capteurs et efficaces en consommation d'énergie. Cependant, ce protocole présume que tous les nœuds connaissent leurs positions par rapport au noeud puits. Cela nécessite l'utilisation des techniques de localisation comme le GPS qui est très gourmand en énergie et créant ainsi une source additionnelle de perte d'énergie. De plus, la quantité de messages de contrôle échangé entre les différents nœuds du réseau est très considérable (les messages périodiques contenant les positions des noeuds, par exemple). Ces messages surchargent le réseau et engendrer des retards dans les délais de livraison des données vu les congestions créées, alors que le but principal pour ce protocole est d'optimiser le délai de bout en bout.

b) Avantages et inconvénients du routage basé QoS

La qualité de service est nécessaire dans les applications de sécurité, de surveillance et les applications temps réel : centres nucléaires, monitoring médical, applications militaires, etc. Certains protocoles de routage proposés pour les réseaux de capteurs (comme SPEED et SAR, par exemple) essayent de remplir des paramètres de QoS, afin de réaliser une livraison de données fiable et avec une qualité satisfaisante aux exigences des applications temps réel. L'utilisation de ces paramètres de QoS dans les réseaux de capteurs assure la fiabilité des transmissions grâce à la qualité des liaisons réalisée lors de la communication entre les différents nœuds, qui se reflète par une augmentation de taux d'arrivée des paquets au noeud puits. Généralement, il est difficile de satisfaire les critères de QoS en tenant compte de la contrainte énergétique qui est toujours présente dans le développement de n'importe quel protocole de routage pour les réseaux de capteurs.

2. Routage basé sur la négociation de données

En détectant le même phénomène, les nœuds capteurs inondent le réseau par les mêmes paquets de données. Ce problème de redondance peut être résolu en employant des protocoles de routage basés sur la négociation. Avant de transmettre, les nœuds capteurs négocient entre

eux leurs données en utilisant des messages de négociation afin d'éliminer la transmission des données redondantes. Cette procédure garantit que seules les informations utiles seront transmises ce qui élimine la redondance des données et économise l'énergie.

a) Protocoles de cette famille

La famille de protocole SPIN est un bon exemple illustratif du routage basé sur la négociation.

SPIN: Sensor Protocol for Information Negotiation

Ce protocole [1, 21, 82] est considéré parmi les premiers protocoles de routage à plat conçu pour les réseaux de capteurs. Il représente une amélioration des techniques d'inondations Flooding et Gossiping.

Avec Flooding chaque capteur retransmet les données reçues par diffusion à ses voisins. Cependant, cette technique présente un inconvénient majeur qui est la duplication des paquets (un nœud peut recevoir plusieurs copies de la même donnée). Gossiping est la version modifiée de la technique d'inondation Flooding, dans laquelle chaque capteur ne diffuse pas les paquets reçus mais retransmet ces derniers à un des voisins sélectionné de façon aléatoire. Les voisins font suivre les paquets de la même manière jusqu'à l'arrivée de l'information à destination. Le Gossiping réduit les redondances de données mais l'arrivée de ces données à destination est retardée.

Pour palier aux problèmes d'inondation (comme la duplication de paquets et la latence élevée de données), SPIN est proposé comme nouvelle approche. Ce protocole utilise un mécanisme de négociation (échange d'informations) entre les nœuds capteurs avant de transmettre n'importe quelle donnée. Dans une négociation, les nœuds capteurs utilisent des méta-donnée (descripteurs de données de haut niveau). Ceci consiste à nommer les données capturées: les décrire avant de les émettre. Cela permet aux éventuels récepteurs de décider de recevoir ou pas le paquet de données, ce qui évite le problème des données redondantes.

Le protocole SPIN utilise trois types de messages: ADV, REQ, DATA. Quand un noeud capteur a une donnée à partager, il l'annonce par la transmission d'un message ADV incluant une méta-donnée. Chaque nœud, s'intéressant à la donnée référencée par ce paquet meta-donnée, peut la récupérer en envoyant un message requête (REQ). À la réception du paquet requête, un message DATA (données) est envoyé au noeud intéressé. Ce message contient la donnée réelle du capteur, avec la méta-donnée comme entête. De cette manière, seuls les nœuds capteurs qui sont intéressés par la donnée annoncée auront une copie.

Les changements de la topologie n'ont qu'une influence locale sur SPIN, car chaque nœud doit uniquement connaître ses voisins à distance d'un seul saut et donc il est concerné que par un traitement local d'informations, ce qui permet de diminuer par un facteur de 3,5 la dissipation d'énergie par rapport à l'inondation classique (Flooding et Gossiping). De plus, le mécanisme de négociation utilisé contribue d'une manière efficace à la réduction du taux de transmission des données redondantes. Cependant, le mécanisme d'annonce (via les messages ADV) utilisé par SPIN, ne peut garantir un acheminement fiable de données vers la destination. Dans le cas où les noeuds intéressés se trouvent relativement loin du noeud source et les capteurs situés entre la source et la destination ne sont pas intéressés par ces données.

b) Avantages et inconvénients du routage basé négociation

Le routage basé sur la négociation apporte plus de conservation à l'énergie consommée dans la communication des données, mais aussi et surtout à l'optimisation du trafic circulant

dans le réseau en limitant le nombre de messages échangés: les informations semblables seront réduites en les remontant une seule fois au nœud puits, ce qui permet d'économiser plus d'énergie dans les capteurs et réduire ainsi le taux de données redondantes transmises et aussi offrir une utilisation efficace de la bande passante en évitant les échanges superflus des messages avec le nœud puits. Néanmoins, ce mécanisme de négociation entre les nœuds produit des retards non tolérables pour délivrer les données au nœud puits. D'autre part, les messages de contrôle échangés entre les nœuds causent une charge supplémentaire dans le réseau, ce qui induit à des situations de congestion et à une perte additionnelle d'énergie.

3. Routage basé sur les requêtes

Dans ce type de routage, le noeud puits génère une requête de données (portant sur une donnée particulière) propagée à travers le réseau afin d'interroger les nœuds capteurs. Le nœud qui détient les données requises doit répondre au nœud demandeur, en envoyant la donnée à travers le chemin inverse de la requête. Souvent, ces requêtes sont exprimées en langage naturel, ou décrites dans des langages d'interrogation à niveau élevé.

a) Protocoles de cette famille

Les deux protocoles: la diffusion dirigée ‘DD’ et Rumor Routing présentés précédemment dans la section exemples de famille de protocoles à plat et centré donnée, sont deux exemples de ce type de routage (les intérêts générés sont sous forme de requêtes). Dans la suite de ce paragraphe on va décrire brièvement deux autres protocoles de cette famille : COUGAR et ACQUIR.

COUGAR et ACQUIR,

Les deux protocoles COUGAR [72] et ACQUIR [48], considèrent le réseau comme un énorme système de base de données réparties, où chaque noeud correspond à une base de données. Cette idée innovatrice a donné lieu à une adaptation des techniques de BDD distribuées pour les réseaux de capteurs. L'idée principale du protocole COUGAR, est d'employer des requêtes déclaratives afin d'abstraire le traitement des requêtes de la couche réseau. L'abstraction est assurée par une nouvelle couche de requête (couche d'interrogation) introduite entre la couche réseau et application. Ce protocole utilise aussi des techniques d'agrégation de données afin d'économiser l'énergie. COUGAR confié les opérations d'agrégation et de transmission de données captées vers le nœud puits à un nœud leader choisi par les nœuds capteurs. Le nœud puits est responsable de la génération d'un plan de requêtes. Ce plan sert à décrire la méthode de sélection du nœud leader pour une requête donnée. Ainsi, il permet de spécifier les informations nécessaires sur le flux de données et son traitement au niveau des nœuds intermédiaires, et aussi l'affectation des requêtes dans le réseau. Par la suite, ce plan est envoyé aux nœuds concernés.

Le deuxième protocole : ACQUIR, est similaire à COUGAR. A son tour, il perçoit le réseau comme une base de données répartie. ACQUIR est bien adapté pour les requêtes complexes constituées de plusieurs sous-requêtes. Le mécanisme d'interrogation du réseau se déroule comme suit: le nœud puits expédie des requêtes à travers le réseau. Chaque nœud recevant la requête peut répondre partiellement, tout en utilisant les données disponibles dans son propre cache. Une fois la requête est complètement résolue, elle est envoyée aux nœuds voisins. Si les données disponibles ne sont pas à jour, le nœud collecte de nouvelles informations à partir des voisins qui se trouvent dans un périmètre de \mathbf{d} sauts (par rapport à la station de base). Une

fois la requête est totalement traitée, elle est renvoyée à la station de base suivant la route inverse de la requête ou via le chemin le plus court.

L'architecture proposée par COUGAR offre une efficacité en consommation d'énergie, en particulier lorsque la quantité des données générée par les nœuds capteurs est très élevée. Ceci est grâce à l'indépendance des interrogations des capteurs dans la couche réseau. Toutefois, l'ajout de la couche additionnelle consacrée aux requêtes (couche d'interrogation) induit plusieurs inconvénients. Par exemple, il introduit une charge supplémentaire en termes de consommation d'énergie et de capacité de stockage au niveau de chaque nœud. Or, le protocole ACQUIRE a présenté une technique élaboratrice pour des requêtes plus complexes. Le mécanisme appliqué par ACQUIRE garantit une interrogation efficace en ajustant la valeur du paramètre **d**. Mais dans le cas, où **d** est égal au diamètre du réseau, le comportement du protocole serait similaire à une inondation. D'autre part, si le paramètre **d** est très petit, la requête passera à travers plusieurs nœuds avant d'être complètement résolue. De plus, l'introduction de la couche supplémentaire utilisée pour répondre aux requêtes, produit des problèmes similaires à COUGAR (tous les deux souffrent des mêmes inconvénients provoqués par la couche supplémentaire).

a) Avantages et inconvénients du routage basé requêtes

L'utilisation du routage orienté requêtes, permet aux protocoles adoptant ce principe de routage de se différencier aux autres protocoles lors de l'acheminement des données captées, car les protocoles utilisant ce mécanisme procèdent à une diffusion directe des requêtes via le nœud puits à travers tout l'ensemble du réseau. Ces requêtes interrogent les nœuds sur une donnée bien déterminée. Par la suite, seulement les nœuds qui détiennent ces données les envoient au nœud demandeur. Cette approche permet d'abaisser le trafic circulent dans le réseau en diminuant la quantité des messages échangés entre les nœuds. En outre, l'utilisation des techniques d'agrégation de données permet de diminuer la consommation d'énergie (par exemple: la suppression des redondances).

Cependant, les protocoles de cette famille sont inappropriés pour les applications de surveillances, telles que le contrôle de l'environnement et les processus industriels, car le routage n'est plus efficace lorsque le nombre de requêtes est grand, engendrant ainsi des retards non tolérables pour délivrer les données au noeud puits. De plus, le coût de maintien des tables de routages au niveau de chaque nœud devient très considérable.

4. Routage multi chemins

Le routage multi chemins permet de faire face à l'un des problèmes les plus rencontré avec les protocoles de routage actuels. Le fait de trouver un simple chemin optimal en énergie et l'utiliser fréquemment pour toutes les communications provoque l'épuisement des batteries des nœuds qui constituent ce chemin. Par conséquent, le réseau sera partitionné et ses performances en termes de tolérance aux pannes seront dégradées. La tolérance aux pannes d'un protocole est mesurée par la probabilité qu'un chemin alternatif existe entre une source et une destination quand le chemin primaire soit défaillant. De ce fait, le routage adopté par les protocoles de cette famille consiste à maintenir plusieurs chemins depuis chaque capteur vers le nœud puits, ce qui permet de garantir la présence de plus d'un chemin fiable pour la transmission, et offre une reprise rapide du transfert de données en cas de panne (choisir l'un des chemins qui restent).

a) Protocoles de cette famille

EAR est présenté comme exemple de protocole de cette famille. Son principe ainsi que ses caractéristiques sont détaillés dans la suite de cette sous section.

EAR: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks

L'utilisation répétée des routes à consommation d'énergie optimale, épuisera rapidement l'énergie des noeuds faisant partie de ces routes. Pour remédier à cette lacune, un chemin alternatif est utilisé avec une certaine probabilité. EAR [79] adopte ce principe en maintenant un ensemble de routes entre la source et la destination, mais n'utilise qu'une seule route à chaque fois. A chaque route est assignée une probabilité dépendante d'une métrique d'énergie. Les noeuds du réseau relaient les paquets reçus en choisissant une route selon les probabilités affectées. EAR s'exécute sur trois phases: initialisation, communication de données et maintenance de route.

-*Phase d'initialisation:* le nœud puits initie les communications en effectuant une inondation pour découvrir tous les chemins vers le nœud destinataire qui intéresse l'utilisateur. À ce niveau, une élection des chemins est établie selon une fonction de coût énergétique, ce qui permet de créer une table de routage. Uniquement, les chemins avec les meilleurs coûts sont gardés dans la table, alors que ceux ayant des coûts élevés en énergie sont abandonnés. Par la suite, chaque nœud assigne une probabilité à chacun de ses voisins correspondant à un chemin dans sa table de routage.

-*Phase de communication des données:* en fonction des probabilités attribuées dans la phase précédente, chaque nœud envoie le paquet vers la destination en choisissant aléatoirement un nœud de sa table de routage.

-*Maintenance des routes:* une inondation est lancée périodiquement afin de maître jour à les tables de routage (les chemins entre le nœud puits et les nœuds du réseau) tout en reprenant les étapes de la phase initialisation.

A la différence du protocole Direct Diffusion décrit précédemment, où les données sont envoyées par des routes multiples et l'une d'entre elles était renforcée pour envoyer les données à des débits plus élevés, avec EAR une route unique est choisie aléatoirement à partir des solutions multiples afin d'économiser l'énergie. Ceci permet à EAR, en le comparant avec Direct Diffusion, de réduire la consommation d'énergie par 21,5%, et augmenter la durée de vie du réseau par 44%.

Toutefois, il est à noter que ce protocole nécessite l'utilisation d'informations de localisation lors du mécanisme d'adressage des nœuds (inondation localisée pour atteindre les noeuds sources), ce qui complique d'avantage l'initialisation des routes et la structure générale du réseau. D'autre part, la construction de ces chemins alternatifs nécessite plus de ressources en termes: d'énergie, quantité de messages de contrôle circulant dans le réseau ainsi que l'espace nécessaire pour stocker les tables de routage, etc.

b) Avantages et inconvénients du routage multi-chemins

Les réseaux de capteurs sont caractérisés par un taux élevé de pannes. Afin d'assurer une bonne fiabilité du routage, certains protocoles construisent plusieurs chemins (le protocole

EAR décrit en haut, par exemple). Cela permet d'obtenir une conservation d'énergie vu l'existence de plusieurs routes entre la source et la destination ce qui augmente la fiabilité et la robustesse du réseau et garantit ainsi une rapidité remarquable lors de la reprise en cas de panne. Cependant, la construction de ces chemins alternatifs requiert plus de ressources énergétiques, messages de contrôle et espace de stockage, car la présence de plusieurs chemins nécessite de les maintenir actifs, ce qui induit une consommation importante en énergie. Les messages échangés périodiquement pour maintenir les chemins alternatifs engendrent, un overhead et une perte additionnelle d'énergie.

2.4 Conclusion

L'énergie est une ressource précieuse dans les réseaux de capteurs, sa conservation est indispensable pour garantir une longue durée de vie pour le réseau, puisque les capteurs sont de petits composants avec une faible capacité de stockage, de calcul et sont alimentés par des batteries dont la capacité est limitée, et généralement non rechargeable. Par exemple, dans certaines applications où les capteurs sont déployés dans des zones d'intérêt hostiles ou toute intervention humaine après déploiement est exclue, il est difficile voire impossible de changer les batteries. Pour cela, les contraintes imposées aux réseaux de capteurs dépendent en premier lieu de la nature des capteurs ainsi que du type de l'application.

Ce chapitre nous a permis de visualiser les différents facteurs de consommation d'énergie dans les réseaux de capteurs et les approches et techniques appliquées pour sa conservation. Parmi les techniques proposées pour diminuer la consommation de l'énergie (énergie de communication en particulier) les mécanismes utilisés dans les protocoles de routage. Une étude critique a été apportée aux principaux protocoles de routage sous contrainte énergétique proposée dans la littérature afin de tirer profit de leurs avantages et voire leurs limites. Par exemple, dans les architectures à plat, la majorité des protocoles de routage conçus pour les réseaux de capteurs de petite ou moyenne taille fournissent de bonnes performances. Cependant, lorsque le nombre de nœuds augmente, le trafic de contrôle prédomine les communications réelles. Ce qui conduit à une augmentation dans la latence et à une explosion dans les tables de routage. Afin de pallier à ces limites, l'architecture hiérarchique s'est considérée comme l'une des solutions communément efficaces pour assurer un bon routage dans les réseaux de capteurs. Pour cela, la technique de clustering évoquée précédemment dans ce chapitre est une solution qui se révèle intéressante pour confronter le défi de conservation d'énergie dans les réseaux de capteurs.

Dans le chapitre suivant, nous présentons en détail l'approche de clustering, où nous allons énumérer d'autres protocoles de cette famille et exposer leurs principes de fonctionnement et caractéristiques.

Chapitre 3

État de L'art sur Les Protocoles de Clustering Pour Les Réseaux De Capteurs Sans fil

3.1 Introduction

Les réseaux de capteurs sans fil sont fortement influencés par la limitation de la ressource énergétique disponible au niveau de chaque nœud. Actuellement, les principaux travaux de recherche dans ce domaine portent sur la prolongation de la durée de vie du réseau, via la conception de nouveaux protocoles de routage. Divers mécanismes peuvent intervenir en vue de l'optimisation de la consommation d'énergie lors de la transmission des données captées vers l'utilisateur final. La structuration hiérarchique du réseau s'est présentée, dans ce contexte, comme une approche de routage qui contribue d'une manière efficace à l'économie d'énergie, mais aussi à la réduction des messages de contrôle, ainsi qu'à l'augmentation de la réutilisabilité de la bande passante et l'attribution de ressources dans le réseau.

Certes, le routage hiérarchique est une stratégie qui vise à organiser le réseau d'une manière favorable à la réduction de la dissipation d'énergie afin de préserver cette ressource épuisable et difficilement rechargeable. Selon un processus de clustering, les nœuds du réseau sont groupés en clusters. Cela est établi par l'élection d'un chef pour le cluster, puis à déterminer les nœuds qui feront partie de ce dernier. Plusieurs techniques peuvent être utilisées pour le partitionnement du réseau en clusters. En effet, ces techniques permettent à classifier les différents protocoles proposés selon cette approche de routage.

Ce chapitre est organisé comme suit. Dans la section 3.2, seront définis l'ensemble des concepts relatifs au mécanisme de clustering ainsi que les motivations qui nous ont amené à choisir ce type de routage pour les réseaux de capteurs. Dans la section 3.3, nous passons en revue les contraintes qui interviennent lors de la conception d'un protocole de routage hiérarchique basé sur le mécanisme de clustering. Les principales techniques de clustering et une description de chacune des catégories sont présentées dans la section 3.4. Dans la section 3.5, nous exposons un ensemble de solutions hiérarchiques, protocoles de clustering, proposées dans la littérature, tout en expliquant leurs caractéristiques et les étapes de leurs constructions. Dans la section 3.6, nous dressons une table comparative qui résume les différentes caractéristiques de ces protocoles de clustering. La conclusion de ce chapitre est donnée en section 3.7.

3.2 Notions de base et motivations

Le type d'application affecte grandement les protocoles de routage dans un réseau de capteurs. La stratégie de déploiement des nœuds a un impact sur ces protocoles. Généralement le déploiement des nœuds dans ces réseaux peut être déterministe ou aléatoire [1].

Lors du déploiement déterministe, les capteurs sont placés manuellement et les données peuvent donc être acheminées via des routes pré-déterminées. Par contre, lors du déploiement aléatoire, les capteurs sont épargnés (ex. lâchés d'un avion). Pour cette répartition non uniforme, une stratégie de clustering s'avère souvent nécessaire. Le clustering, permet de mieux gérer le trafic du réseau et d'alléger la quantité d'information qui circule en effectuant des traitements au sein de chaque cluster avant de propager les données vers le reste du réseau pour les transmettre à la station de base.

3.2.1 Définitions

Dans des milieux hostiles et dangereux où l'intervention humaine est impraticable, il n'est pas toujours faisable d'organiser manuellement les nœuds en groupes. Pour cette raison, il y a eu une grande quantité de travaux de recherche sur des manières de créer des structures logiques d'organisation. Le clustering est considéré comme un point clé dans la conception de ces réseaux. Avec cette méthode l'ensemble des capteurs est divisé en groupes (ou clusters), et pour chaque cluster un chef pour le cluster ou cluster-head est élu [44]. Dans chaque cluster, les nœuds envoient leurs données au nœud cluster-head (voir figure 3.1) qui transmet à son tour la donnée à la station de base (si nécessaire, par l'intermédiaire des cluster-heads des autres clusters). La tâche d'un cluster-head ne consiste pas seulement à router les données, mais elle peut être complétée par un traitement et une agrégation/fusion de données issues des autres capteurs afin de faciliter la gestion du réseau et en particulier l'optimisation des dépenses énergétiques. Les cluster-heads sont supposés avoir un niveau d'énergie supérieur et une plus grande capacité de stockage et de calcul que les autres noeuds capteurs [42,45], alors que les nœuds plus contraints en ressources énergétiques se consacrent uniquement à la capture.

En effet, l'utilisation du mécanisme de clustering dans les réseaux de capteurs facilite l'agrégation de données, ce qui conduit à diminuer les redondances et les transmissions inutiles vers la station de base. D'autre part, le clustering permet aux nœuds d'effectuer des transmissions sur des courtes distances avec leur cluster-head, ce qui minimise la consommation d'énergie et optimise l'utilisation des ressources du medium de communication [42, 43].

Nous présentons ci-dessous les principaux composants d'une architecture hiérarchique pour un réseau de capteurs [42]:

- **Nœud capteur** : Le noeud capteur est le composant noyau du réseau. De plus, ces capteurs peuvent jouer des rôles multiples, tel que la sensation simple, stockage, conduite de données et le traitement (voir §1.1.2.1). Cela dépend de leurs capacités en termes de calculs et puissance énergétique sans oublier l'influence du mécanisme d'affectation des rôles adapté par le protocole de clustering.
- **Cluster** : Le cluster est un ensemble de noeuds capteurs qui forme l'unité d'organisation d'un réseau de capteurs. La nature dense de ses réseaux exige de les décomposer en cellules ou clusters afin de simplifier les tâches de communication et de répondre aux différentes contraintes.
- **Cluster-heads (CHs, leaders, chefs de groupes)** : Se sont les représentants des clusters. Ils sont souvent nécessaires pour l'organisation des activités dans les clusters. Leur tâche ne se limite pas à l'agrégation de données, elle s'étend aussi à l'organisation de la communication intra-cluster et même inter-clusters. Ces chefs peuvent être élus par les autres nœuds ou bien

sont pré-assignés par le concepteur du réseau. Ils peuvent aussi être juste des nœuds ordinaires, ou bien des noeuds dotés de plus d'énergie.

- **Station de base :** La station de base se situe à un niveau supérieur de l'hiérarchie d'un réseau de capteurs. Elle permet de fournir une liaison entre le réseau de et l'utilisateur final.
- **L'utilisateur :** Les données générées par un réseau de capteurs sont employées dans une large gamme d'applications [1]. Ces dernières sont accessibles via Internet, à l'aide d'un PDA ou d'un ordinateur de bureau. Une simple requête générée par l'utilisateur final est diffusée dans le réseau permet de recueillir toutes les données souhaitées.

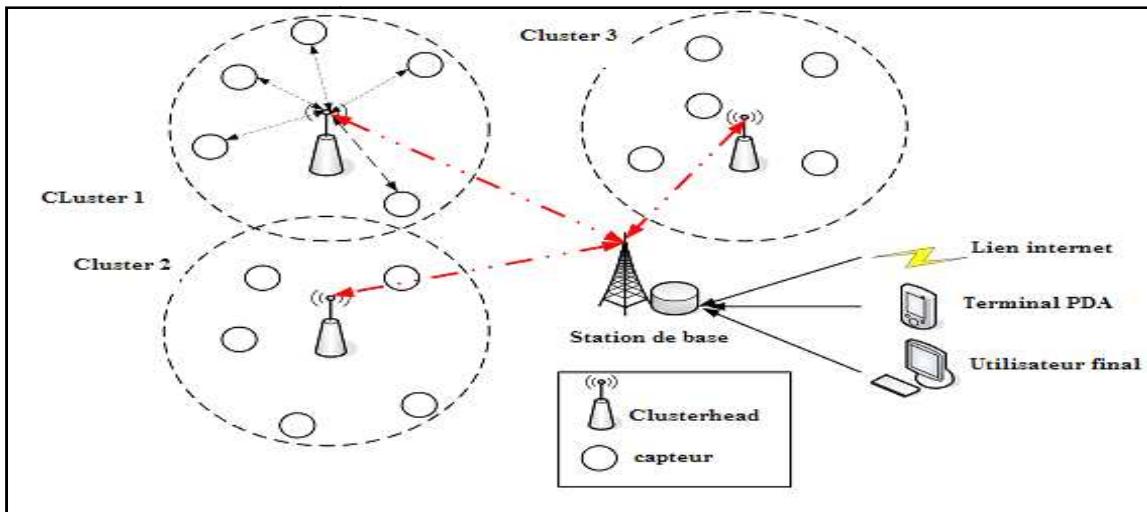


Fig. 3.1 Hiérarchie d'un réseau de capteurs sans fil [42].

3.2.2 Formation de clusters

L'efficacité d'un algorithme de clustering est évaluée en termes de nombre de clusters formés et de leurs stabilités en fonction de la mobilité des noeuds (la capacité de s'adapter au changement de la topologie du réseau). Il existe plusieurs méthodes de formation de clusters. Mais la plus répandue s'exécute sur trois phases :

Election de cluster-head

Appelée aussi la phase Set-up : chaque nœud prend la décision de devenir CH ou bien de se joindre à un CH selon sa connaissance locale de la topologie en utilisant une métrique spécifique ou une combinaison de métriques tels que le plus grand/petit ID dans son voisinage, le degré de connectivité, la puissance de transmission, l'énergie résiduelle..., ou bien selon un poids qui représente une combinaison de quelques métriques.

Communication intra-cluster et inter-cluster

Chaque cluster-head se charge des communications à l'intérieur de son cluster et maintient les informations de routage lui permettant de joindre la station de base. Pour atteindre la station de base, les cluster-heads communiquent directement à cette dernière sinon une communication multi-saut est adoptée en utilisant d'autres cluster-heads comme noeuds relais.

Maintenance des clusters

Dans le but de s'adapter aux changements de la topologie du réseau, une mise à jour des clusters est dynamiquement réalisée pour reconnaître les nouveaux noeuds ajoutés ou ceux disparus (les noeuds qui ont consommé leurs énergies). D'autre part, si le cluster-head garde son statut le plus longtemps possible, même s'il ne possède pas par exemple le poids maximum dans son propre cluster alors il perdra son rôle une fois sa batterie sera épuisée.

3.2.3 Avantages de l'approche clustering

Le mécanisme de clustering permet de [42, 43, 44] :

- Faciliter le partage des ressources et /ou la synchronisation au sein d'un cluster.
- Optimiser l'utilisation de la bande passante en limitant les interactions inter-clusters et évitant les échanges superflus des messages avec la station de base ; alléger le flux de communication dans le réseau.
- Réduire la taille des tables de routage stockées au niveau des nœuds en localisant le chemin, car la mise en place des routes se fait seulement au niveau des cluster-heads.
- Agrégation des données transmises au collecteur. Les données qui sont collectées à partir d'un ensemble de nœuds d'un cluster peuvent être fusionnées par un cluster-head et par la suite envoyées vers la station de base ce qui permet la réduction du nombre de paquets échangés.
- Stabiliser la topologie au niveau des nœuds et donc minimiser les frais de sa maintenance. Un nœud ne s'inquiète que du choix du cluster-head auquel il va être relié. Par conséquent il n'est pas affecté par les changements au niveau supérieur.
- Implémenter des mécanismes par les CHs afin d'augmenter la durée de vie de la batterie, ce qui permet de prolonger la durée de vie du réseau. Par exemple, le cluster-head programme les activités dans le cluster de sorte que les nœuds puissent dormir le plus de temps possible et donc diminuer le taux de communication ainsi que le taux de collisions pour y accéder au canal de communication.

3.3 Contraintes et facteurs de conception lors du processus de clustering

L'approche de clustering joue un rôle important dans l'organisation, non seulement du réseau, mais il affecte nettement les performances de ce dernier. Il y a plusieurs contraintes et facteurs qui doivent être considérés lors de la conception des clusters et la structuration du réseau. Ces facteurs importants servent comme directives pour le développement des protocoles de clustering utilisés dans les réseaux de capteurs.

Énergie limitée vs Durée de vie du réseau

Les nœuds capteurs ont un stockage d'énergie limité, et l'utilisation efficace de cette dernière est un défi à relever. Cette contrainte énergétique doit être prise en compte dans le processus de clustering afin de réduire la dissipation d'énergie entre les nœuds du réseau. Par ailleurs, ce défi

énergétique a un impact direct sur la durée de vie du réseau en entier. Pour cela, grouper les nœuds dans des clusters devrait organiser la topologie du réseau et contribuer à réduire la consommation d'énergie de ses nœuds afin d'augmenter sa durée de vie [1, 42].

Capacités limitées

La taille physique des capteurs et la petite quantité d'énergie stockée dans un nœud, limitent beaucoup la capacité de ces capteurs en termes de communication et puissance de traitement. Un bon protocole de clustering devrait se servir des ressources partagées entre les nœuds tout en considérant la limitation de leurs capacités [42].

Domaine d'application vs Choix de Cluster-heads et formation de clusters

Le type de l'application a une grande influence sur l'organisation en clusters. Étant donné un protocole de clustering, il doit être en mesure de s'adapter à la variété des conditions rencontrées et même d'essayer de rependre aux différents besoins de l'application. En effet, lors de la conception d'un réseau de capteurs, les constructeurs doivent examiner soigneusement la formation des clusters et le choix de cluster-heads dans ce réseau. Pour certaines applications, telle que la surveillance et le contrôle d'un champs de bataille où les nœuds sont exposés à un grand risque d'endommagement par des actions hostiles et leurs informations captées sont très critiques, des exigences concernant le nombre de nœuds dans un cluster ou la taille physique d'un capteur sont très importantes pour assurer un bon fonctionnement du réseau. Ces conditions et bien d'autres comme l'énergie résiduelle au niveau de chaque nœuds, le chemin à faible consommation d'énergie pour router les données et aussi les différentes techniques utilisées pour l'agrégation de données collectées, peuvent aussi avoir un impact sur la façon dont les cluster-heads sont choisis ainsi que sur l'organisation du réseau en clusters [42,46].

Agrégation de données

Une des caractéristiques majeure des réseaux de capteurs est la densité des nœuds. Cependant, il y a souvent de multiples capteurs qui recueillent la même information dans le réseau. L'agrégation sert à combiner les données qui proviennent de plusieurs nœuds en une information significative ce qui éliminera la redondance. Entre autre, l'utilisation des techniques d'agrégation de données dans les protocoles de clustering réduit la quantité des données transmises, minimise considérablement la communication entre les nœuds, résoud le problème d'implosion dans les tables de routage et allège en conséquence la congestion du réseau.

Tolérances aux pannes

La petite quantité d'énergie dans les nœuds est parmi l'une des limitations primaires dans les réseaux de capteurs. Ces derniers sont souvent enclins à la mort et les interférences. Ces situations peuvent entraîner l'échec de liens de connectivité du réseau. Il est donc important de prévoir lors de la conception des protocoles de clustering des mécanismes de recouvrement après panne (algorithmes curatifs : qui utilisent une approche optimiste ne masquant pas les pannes, où le mécanisme de tolérance aux pannes implémenté n'est exécuté qu'après la détection de pannes) ou bien implémenter des techniques tolérantes aux pannes, approche préventive, qui tente à retarder ou éviter tout type d'erreur afin de garder le réseau fonctionnel le plus longtemps possible tout en essayant de consommer le moins d'énergie et donc éviter par la suite une

extinction prématuée de batterie, ce qui augmente la durée de vie des nœuds et assure une communication fiable même après la faille de certains de ces composants (voir §3.4.1.2)

Qualité de service (QoS) et les opérations temps-réel

Quelques applications, exigent que les données fournies au collecteur doivent être de bonne qualité, et par conséquent de nombreuses exigences de QoS dans les réseaux de capteurs dépendent bien du type d'application. Par exemple, dans les applications de surveillance d'habitats, la réception des données est suffisante pour l'analyse, et la latence de l'opération n'est pas critique. Tandis qu'aux applications militaires, l'acquisition des données en temps réel est très critique. Par ailleurs, les protocoles de clustering doivent accorder une attention particulière à la tolérance aux pertes des paquets et aux délais de création des clusters. En addition, le temps nécessaire pour les mécanismes de rétablissement du processus de clustering nécessite également une prise en compte. Il est donc envisageable de se pencher sur ces paramètres au moment du choix d'un schéma de clustering. Les implémentations peuvent varier considérablement en fonction de ces paramètres, et en conséquence, le processus de conception doit tenir compte de tous ces aspects [42].

3.4 Techniques de clustering

Les techniques de clustering proposées dans la littérature pour les réseaux de capteurs sont généralement classées selon [43, 49] :

- L'architecture du réseau et le modèle de fonctionnement.
- Les objectifs du processus de clustering.
- Taxonomie d'attributs de clustering y compris le nombre de clusters désirés et les propriétés générées de ces derniers.
- Les paramètres de prise de décision.

Cette classification de différentes méthodes de clustering utilisées dans la littérature va permettre de fournir une vue générale sur ce mécanisme, mais aussi sur les multiples choix possibles lors de la conception des clusters, les objectifs à atteindre, les propriétés des clusters / cluster-heads à satisfaire, et les paramètres intervenants dans la prise de décisions lors de ce processus. Chaque critère va être étudié en détail avec une description des catégories associées et ses caractéristiques ainsi que son influence sur l'organisation du réseau en cluster.

3.4.1 L'architecture du réseau et le modèle de fonctionnement

Différentes architectures et conceptions ont été considérées pour les applications des réseaux de capteurs. Dans la suite de ce paragraphe, les principaux paramètres architecturaux sont mis en évidence ainsi que leurs incidences sur le mécanisme de clustering.

3.4.1.1 Traitement de données dans le réseau

Dans un réseau de capteurs, plusieurs noeuds peuvent générer des données dupliquées. Cependant, les paquets semblables peuvent être agrégés de sorte que le nombre de transmissions

soit réduit. L'agrégation combine les données de différentes sources en employant des fonctions telles que la suppression (élimination de redondances), minimum, maximum et la moyenne [52]. Généralement, toutes les fonctions d'agrégation sont assignées à des noeuds plus puissants et plus spécialisés ; il s'agit des noeuds élus comme cluster-heads (CHs). Cette contrainte restreint le choix des CHs, et limite le nombre des noeuds par cluster afin de ne pas surcharger ces CHs [43]. Par ailleurs, il est intéressant de choisir un autre noeud comme back up d'un CH, un noeud de remplacement en cas de panne (voir §3.4.2.3), ou bien adapter un mécanisme de choix rotationnelle et aléatoire pour rôle de CH entre les différents noeuds du réseau afin de prolonger leur durée de vie et par conséquent celle du réseau. La conception d'un schéma de clustering est influencée par tels choix /contraintes.

3.4.1.2 Mobilité du réseau

La majorité des architectures réseaux supposent que les noeuds capteurs sont stationnaires et peu d'entre elles utilisent des capteurs mobiles. Mais pour certaines applications, il est important de supporter la mobilité des noeuds [51]. Sous de telles conditions, le clustering devient une tâche très ardue. Les noeuds changent leurs positions et les clusters se voient obligés de s'adapter aux différentes configurations du réseau. D'autre part, les événements contrôlés par un capteur peuvent être relevés périodiquement ou continuellement dépendamment du type d'application. Par exemple, dans des applications de contrôle industriel : le réseau travaille en mode réactif, il suffit de générer du trafic lorsqu'un événement survient. Alors que dans la plupart des applications nécessitant un suivi continu : comme dans les réseaux surveillant des données environnementales, la soumission des rapports périodiques est exigée, ce qui génère un trafic important routé vers la station de base. Bien que les applications à suivi continu permettent d'avoir des groupes stables, elles peuvent augmenter considérablement la charge des CHs en fonction de nombre des noeuds présents dans les clusters. Dans ce cas, une attribution rotationnelle (round-robin par exemple) du rôle de CHs est potentiellement nécessaire si ce dernier est choisi aléatoirement [43].

3.4.1.3 Mode de déploiement et capacité d'un noeud

Le déploiement des noeuds capteurs est soit déterministe ou auto-organisationnel. Dans les situations déterministes, les capteurs sont placés manuellement et les données sont acheminées par des routes pré-déterminées. Toutefois, dans les systèmes à auto-organisation, les noeuds sont disséminés de manière aléatoire, créant ainsi une infrastructure ad-hoc. Dans cette distribution, la position de la station de base ou du CH est cruciale en termes d'efficacité énergétique et de performance du réseau. Par exemple dans le cas d'un réseau à capteurs homogènes, ayant la même capacité en termes de calcul, communication et puissance, les CHs sont sélectionnées par les capteurs déployés. Ces derniers sont chargés soigneusement, exclus des fonctions de capture, afin d'éviter l'épuisement rapide de leur énergie. En outre, les portées de la communication et la proximité relative de ces CHs à la station de base sont des enjeux qui doivent être considérés. Habituellement, un capteur a une portée limitée. Par conséquent, un CH peut ne pas atteindre la station de base. Il est donc préférable d'utiliser des routes multi-saut. D'autre part, les noeuds hétérogènes peuvent imposer plus de contraintes sur le processus de clustering. Certains noeuds peuvent être désignés pour des tâches spéciales ou habilité à capacités distinctes. De ce fait, il est requis de limiter le choix des CHs à un sous-ensemble de ces noeuds de sorte à conserver leurs ressources le plus longtemps possible [4, 43, 53].

3.4.2 Objectifs du clustering

Les protocoles de clustering dans la littérature varient selon leurs objectifs. Souvent, l'objectif est défini en vue de répondre aux exigences des applications. Par exemple l'application peut être sensible à la latence des données, la durée de vie de la batterie et bien d'autres. Ces recommandations sont considérées comme des critères pour le choix des CHs et la construction des clusters. Dans ce qui suit nous présentons quelques objectifs visés par le processus de clustering.

3.4.2.1 Minimiser la consommation d'énergie

La batterie d'un capteur est un composant de petite taille. Elle fournit une quantité d'énergie très limitée. Par conséquent, elle influe sur la durée de vie du réseau et particulièrement dans les environnements difficiles. Et donc, le facteur de consommation d'énergie est d'une importance primordiale dans ce type de réseau. Il est souhaitable de réduire au minimum l'énergie de la communication en adoptant des stratégies pour placer les CHs près de la plupart des nœuds d'un cluster ce qui permet de minimiser l'énergie de communication intra-cluster. Dans ce cas, le clustering adaptatif⁵ d'un réseau peut s'avérer comme un choix viable pour la prolongation de la durée de vie de ce dernier [43].

3.4.2.2 Équilibrage de la charge “Laod balancing”

La répartition équilibrée des capteurs sur les clusters est généralement un objectif pour les configurations où les CHs sont sélectionnés par ces capteurs, afin d'effectuer des fonctions de traitement de données et de gestion d'intra-clusters. Cependant, la mise en clusters à taille égale devient cruciale pour l'allongement de la durée de vie du réseau. Ceci empêche l'épuisement de l'énergie d'un sous-ensemble de CHs à taux élevés de nœuds, ou de les rendre prématurément dysfonctionnelles. Il est impératif de disposer d'un nombre identique de nœuds dans les clusters quand les CHs effectuent l'agrégation de données de sorte que les rapports combinés deviennent prêts au même temps pour un traitement ultérieur à la station de base, ou au niveau supérieur dans la hiérarchie du réseau [43].

3.4.2.3 Tolérance aux pannes

Des les environnements difficiles où toute intervention humaine après déploiement est exclue, les nœuds capteurs sont généralement exposés à un risque accru de dysfonctionnement liés aux dommages physique. La tolérance à l'échec des nœuds CHs est généralement nécessaire dans de telles situations afin d'éviter la perte des données sensées. Cependant, le re-clustering du réseau est une solution pour répondre à cette contrainte. Toutefois, une nouvelle organisation n'est pas que bénéfique pour le réseau, mais elle est souvent source de perturbation pour l'opération en cours. Par conséquent, les nouvelles techniques contemporaines de tolérance aux pannes seront plus appropriées. L'assignation de CH backup est la méthode la plus utilisée dans la littérature pour le rétablissement de l'échec d'un CH. Le choix et le rôle que ces CHs backups joueront pendant le fonctionnement normal du réseau varient. Par ailleurs, la rotation du rôle de

⁵ C'est un clustering dynamique qui doit être capable de prendre en compte l'évolution de la topologie en cas de panne et d'expiration d'énergie de certains capteurs, etc.

CH entre les nœuds du réseau peut également être un moyen pour la tolérance aux pannes, de plus que leur avantage dans l'équilibrage de charge [43, 54].

3.4.3 Taxonomie d'attributs de clustering

Un ensemble d'attributs (voir figure 3.2) est potentiellement identifiable pour classer par catégorie et différencier les protocoles de clustering dédiés aux réseaux de capteurs sans fil.

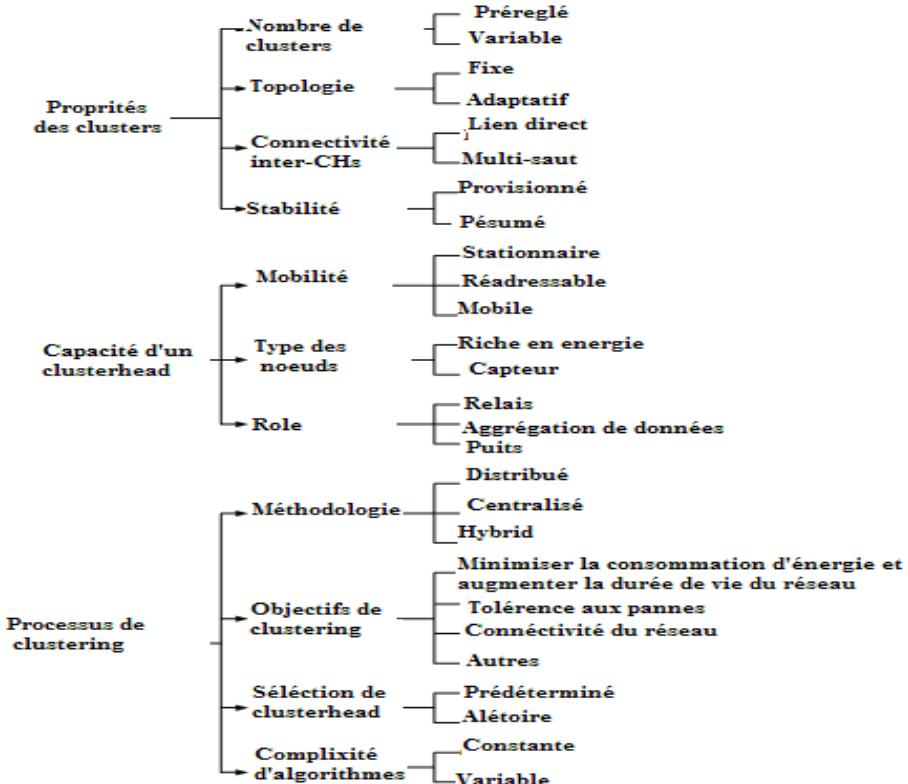


Fig.3.2 Taxonomie d'attributs de clustering dans un réseau de capteurs.

3.4.3.1 Propriétés des clusters

Souvent, le processus de clustering s'efforce à parvenir à un certain nombre de caractéristiques pour les clusters générés. Ces caractéristiques peuvent être liées à la structure interne du cluster ou à la façon dont il se rapporte avec d'autres clusters. Ci-dessous les attributs les plus pertinents [43, 49] :

-*Nombre et taille de Clusters* : Si l'ensemble des CHs est prédéterminé, le nombre de clusters dans le réseau sera prédefini. Toutefois, la sélection aléatoire des CHs par les capteurs déployés résulte en un nombre variable de clusters. Dans certains cas, il est souhaitable d'avoir un petit nombre de clusters afin d'acheminer les paquets rapidement entre ces clusters, mais dans d'autres cas (en présence de mobilité par exemple) il est important de garder une petite taille pour les clusters et par conséquent avoir un plus grand nombre de clusters pour pouvoir gérer les changements de la topologie. [33, 58].

-*Stabilité* : Si le nombre de cluster et l'appartenance d'un nœud à un cluster variaient tous le temps, le processus de clustering serait adaptatif. Autrement, il est considéré comme stable,

fixe et les capteurs ne commutent pas entre les clusters. De même, le nombre de ces clusters reste fixe tout au long de la durée de vie du réseau.

-*Topologie intra-clusters* : La communication entre un nœud et son CH dans un cluster est directe. Néanmoins, la connexion via multi-sauts entre les nœuds et leur CH est parfois exigée; particulièrement quand la portée de la communication dans le cluster est limitée.

-*Connectivité inter-CHs* : Une grande partie des travaux publiés sur les réseaux de capteurs présume que le CH peut atteindre directement la station de base. D'autre part, si le CH n'a pas la capacité de communication à longues distances, la connexion des CHs à la station de base doit être assurer par l'établissement d'une route inter-CHs jusqu'à la station de base.

3.4.3.2 Capacités d'un cluster-head

Le modèle du réseau influe sur le processus de clustering ; en particulier sur la capacité d'un nœud et sa portée de transmission. Les attributs de CHs suivants sont les facteurs de différenciation entre les schémas de clustering :

-*Mobilité* : Lorsque le CH est mobile, les nœuds membres sont modifiés dynamiquement et les clusters doivent être maintenus tout le temps. Autrement, les cluster-heads stationnaires tendent à avoir des groupes stables et à faciliter la gestion du réseau. Parfois, le CH peut se déplacer sur des distances limitée et à se repositionner afin d'avoir une meilleure performance pour le réseau.

-*Type des nœuds*: Comme mentionné précédemment (voir §3.4.1.3), dans certaines configurations, parmi les capteurs déployés, un sous-ensemble est désigné comme CHs. Tandis que dans d'autres, les CHs sont des capteurs dotés de plus de ressources de calcul et de communication (capteurs dédiés).

-*Rôle*: Un CH peut simplement agir dans son cluster en tant que point de relais pour le trafic produit par les nœuds, ou exécuter l'agrégation/fusion des données rassemblées. Il peut également jouer le rôle d'un nœud puits et agir selon les phénomènes détectés.

3.4.3.3 Le processus de clustering

Les caractéristiques du processus de clustering varient de manière significative selon: la méthodologie adoptée, les objectifs à satisfaire, critères de sélection de CHs ainsi que la complexité de ses protocoles. Ces paramètres influent sur le classement des protocoles que nous présentons dans la section suivante.

- *Méthodologie* : Dans peu d'approches, une autorité centralisée, la station de base par exemple, affecte le rôle de chaque nœud et commande la formation des clusters. Autrement, si les CHs sont juste des capteurs ordinaires, la construction des groupes doit être faite d'une façon autonome sans coordination. Or que dans le cas de cluster-heads riches en ressources, des techniques hybrides peuvent être utilisés. Cependant, la coordination inter-CHs est assurée d'une manière distribuée, alors que chaque CH se charge individuellement de la formation de son propre cluster. Une autre méthode est potentiellement employée pour avoir une architecture hiérarchiques; clusters de clusters (voir § 3.5.2.4).

-Objectifs de clustering : Comme déjà mentionné dans 3.4.2, la formation des clusters et le choix de CHs dans un réseau de capteurs est influencée par plusieurs objectifs. En effet, un processus de clustering peut avoir de multiples objectifs. Par exemple la minimisation de la consommation d'énergie dans le réseau, la tolérance aux pannes, l'équilibrage de charge, etc.

-Choix de cluster-heads : Un CH peut être pré-assigné ou bien sélectionné aléatoirement de l'ensemble des nœuds déployés. Généralement, les idées principales que l'on retrouve pour l'élection de CHs sont basées sur différentes métriques. L'énergie résiduelle⁶ au niveau d'un nœud, le voisinage peut aussi être envisagé comme critère de choix : le nœud ayant le plus grand nombre de voisins est désigné CH. La mobilité est pratiquement un autre paramètre de choix : le nœud de plus faible mobilité est élu tête de cluster. Le fait de prendre en compte la mobilité pour l'élection de CH assure une certaine stabilité lors du découpage du réseau. En effet, si on élit comme tête de cluster un nœud trop mobile, à chaque période, ses voisins auront changé et il devra très souvent relancer la procédure d'élection. Par ailleurs, le poids attribué à chacun des nœuds utilisés pour l'élection de CHs.

-Complexité du protocole : Selon l'objectif et la méthodologie, de nombreux protocoles de clustering ont été proposés. La complexité et le temps de convergence (voir 3.4.1) de ces protocoles peuvent être constants, ou dépendant du nombre de CHs et/ou de capteurs. Alors, la complexité d'un protocole de clustering est essentielle pour l'estimation de la latence des messages échangés et les frais généraux de construction et d'entretien des clusters.

3.4.4 Paramètres de décision

La décision de devenir nœud CH, ou pour rejoindre un cluster existant et construire des nouveaux clusters est généralement basée sur les paramètres suivants : le temps de convergence, le poids assigné à un nœud, l'émergence du processus de clustering et les heuristiques. Ces paramètres vont être utilisés pour la classification des différents protocoles de clustering mentionnés dans la section suivante [49].

3.4.4.1 Décision basée sur le temps de convergence

Si on considère que le point de départ d'un algorithme de clustering est l'état stable. Un seul événement, dans ce réseau (par exemple, un lien ou un capteur ajouté ou supprimé) va déclencher une série de mesures de restructuration de l'organisation du réseau [49]. Le temps nécessaire pour que l'algorithme termine cet événement afin de réaliser une structure valide de clusters est noté la convergence du temps. Ce temps de convergence peut être constant: indépendant de la taille du réseau, ou variable selon la méthodologie suivie pour le choix de cluster-heads et/ou la formation de clusters.

3.4.4.2 Décision basée sur les heuristiques

Une heuristique est un algorithme qui fournit rapidement une solution réalisable, pas nécessairement optimale pour un problème d'optimisation NP-complet⁷. Par exemple déterminer le nombre de clusters/cluster-heads dans le réseau. Une heuristique ou méthode approximative,

⁶ Quantité d'énergie restante au niveau d'un nœud

⁷ En ligne <http://fr.wikipedia.org/wiki/Heuristique>.

est donc le contraire d'un algorithme exact qui trouve une solution optimale pour un problème donné. Il est généralement plus judicieux de faire appel à des méthodes heuristiques pour des problèmes difficiles ou l'usage de cette heuristique est pertinent pour calculer une solution approchée telle que le problème de clustering, et ainsi accélérer le processus de résolution (voir 3.5.3).

3.4.4.3 Décision basée sur le poids

Un poids est une fonction spécifique assigné à chaque noeud dans le réseau. Ce poids dépend de plusieurs métriques impliquées dans son calcul comme l'énergie acquise par un nœud, le degré et la densité des nœuds, la distance aux voisins...etc.

3.4.4.4 Décision basée sur l'émergence

L'émergence d'un algorithme se base sur un principe d'interactions subtiles entre éléments simples (nœuds) vers une organisation complexe (hiérarchie de clusters) possible qu'à grande échelle⁸. Ce principe peut être approché analogiquement par la construction d'un avion où tous les éléments : ouvrier, techniciens, interagissent pour la construction complexe de cet avion sans que chacun eu un plan.

3.5 Approches de clustering pour les réseaux de capteurs sans fil

Afin d'avoir une topologie plus stable et plus scalable pour le réseau de capteurs, de nombreuses solutions basées sur le clustering ont été proposées dans la littérature. En outre, ces protocoles permettent aussi la réutilisation spatiale de la bande passante ainsi que son contrôle. Ainsi, plusieurs classifications ont été apportées à ces protocoles.

Dans cette section, cinq grandes familles vont être exposées : les protocoles hiérarchiques à temps de convergence constant et variable, les protocoles heuristique et pondérés et enfin les protocoles émergents (voir tableau 3.1).

Algorithmes de clustering				
Les protocoles hiérarchiques à temps de convergence constant	Les protocoles hiérarchiques à temps de convergence variable	Les protocoles heuristiques	Les protocoles pondérés	Les protocoles émergents
<ul style="list-style-type: none"> - LEACH - HEED - EECS - EEUC 	<ul style="list-style-type: none"> - EEHC - RCC - Clubs - Hierarchical clustering 	<ul style="list-style-type: none"> - LCA2 - Max-Min-D cluster - Highest Connectivity Cluster 	<ul style="list-style-type: none"> - WCA - DCA 	<ul style="list-style-type: none"> - ACE

Tab 3.1 Quelques Algorithmes de Clustering pour les réseaux de capteurs

⁸ En ligne <http://en.wikipedia.org/wiki/Emergence>.

3.5.1 Les protocoles hiérarchiques à temps de convergence constant

La formation des clusters et le choix de leurs chefs dans ce type de protocoles s'achèvent en un temps de convergence constant. Généralement, la construction des groupes et l'élection des CHs sont basées sur une probabilité en fonction du nombre de nœuds dans le réseau, nombre de fois que le nœud a été choisi CH, la taille du cluster, l'énergie résiduelle locale des capteurs et enfin sur la proximité de ces derniers du CH à élire. Des exemples sur ce type de protocoles seront présentés dans le paragraphe suivant. Plus de détail sera apporté au protocole LEACH [33] qui fait partie des premiers protocoles de clustering proposés pour les réseaux de capteurs, et considéré comme protocoles de référence dans le domaine du routage hiérarchiques. Il a constitué par la suite une source d'inspiration pour plusieurs autres protocoles dont quelques uns seront l'objet de sa sous-section améliorations.

3.5.1.1 LEACH: Low-Energy Adaptive Clustering Hierarchy

Parmi les premiers protocoles de routage hiérarchiques conçus pour les réseaux de capteurs sans fil; on retrouve le protocole LEACH proposé par Heinzelman et Chandrakasan [33, 53]. Pour une meilleure allocation de ressources est contrôle d'énergie, ce protocole s'appuie sur le principe de partitionnement du réseau en clusters, de façon similaire aux réseaux cellulaires selon l'amplitude du signal reçu. Les nœuds organisés en clusters transmettent leurs données aux CHs, qui à leur tour, fusionnent ces données rassemblées et les envoient vers la station de base. De ce fait, les CHs doivent posséder plus d'énergie que les nœuds membres. Par conséquent, une rotation aléatoire de la position du CH est utilisée entre ces membres, afin d'assurer une répartition efficace de la consommation énergétique et l'utilisation de la bande passante entre tous les nœuds du réseau.

Le système fixe le nombre de clusters en fonction de quelques paramètres comme la topologie du réseau, le coût des opérations de communication et de calcul. Généralement la valeur optimale de CHs représente 5% des noeuds du réseau.

Le fonctionnement de ce protocole est faite cycliquement où chaque cycle est décomposé en deux grandes phases: initialisation et transmission (état-stable).

Initialement, les clusters sont formés et les cluster-heads sont sélectionnés. Les nœuds s'auto-élisent pour être des CHs en se basant sur le pourcentage de CHs désiré pour ces derniers, déterminé a priori, et le nombre de cycles pendant lequel un nœud a joué le rôle de CH. Ainsi, un nœud n choisit une valeur aléatoire entre 0 et 1. Si ce nombre est inférieur à une valeur seuil $T(n)$, ce nœud se déclare CH pour le cycle courant.

$$\text{La valeur de } T(n) \text{ est calculée comme suit : } T(n) = \frac{P}{1 - P(r \bmod (1/P))} \text{ si } n \in G$$

Où : P : pourcentage désiré de CHs. r : itération actuelle, Et G : ensemble des nœuds qui ont été sélectionnés comme CH durant les dernières (1/ P) itérations.

Les nœuds qui ont été élus comme cluster-heads informeront alors leur voisinage de leur élection par envoie de messages tout en utilisant le protocole CSMA non persistant de la couche MAC. A la réception des annonces faites par ces CHs, les nœuds non cluster-heads, selon la puissance du signal reçu, déterminent le cluster auquel ils veulent appartenir puis communiquent leurs choix au CH choisi.

Après la réception de tous les messages des nœuds désirant être membres d'un cluster, les CHs élus peuvent alors déterminer le nombre de ces membres et les ordonnancés par la suite, en créant des tables TDMA 'Time Division Multiplexed Access' (allouer à chaque membre un slot de temps de transmission) afin d'éviter les collisions intra-cluster. En outre, un code CDMA est choisi par les CHs afin d'être utilisé par les nœuds de son cluster lors de la phase de transmission de données.

Transmission : une fois que les intervalles de temps des tables TDMA et les codes CDMA sont déterminés et annoncés aux différents membres, la deuxième phase celle de transmission est lancée. Chaque nœud ne peut émettre ses données vers le CH de son cluster que durant l'intervalle alloué. Ce dernier va collecter tous les données reçues, les agréger et les router à la station de base.

Discussion

LEACH est un protocole décentralisé, distingué par un temps de convergence constant, sa capacité de minimiser la consommation énergétique des nœuds et d'allonger la durée de vie du réseau, ainsi que par sa particularité qui réside dans la non nécessité de communication entre les différents noeuds pour l'élection des CHs.

Cependant, vu le manque de connaissance de la topologie globale, ce protocole n'assure pas le nombre de CHs ni leurs positions. Donc les CHs peuvent être non uniformément distribués, et concentrés dans une seule partie du réseau. Le fait d'avoir une décision probabiliste pour choisir le CH, permettrait à un nœud avec une puissance énergétique basse d'être élu comme CH. En conséquence, quand ce nœud meurt, la cellule entière devient dysfonctionnelle. D'autre part, ce protocole présume certaines suppositions qui ne sont pas toujours évidentes. Par exemple, il prétend que les CHs peuvent transmettre avec une quantité d'énergie suffisante pour atteindre directement la station de base (via un seul-saut). Ce qui n'est pas toujours réalisable.

Améliorations

Une multitude de protocoles a été proposée en s'inspirant du protocole LEACH, afin d'apporter des améliorations à ce dernier tout en essayant de pallier à ses lacunes. Dans ce qui suit, nous présentons trois protocoles comme exemples d'améliorations.

1. LEACH-C : Low-Energy Adaptive Clustering Hierarchy Centralized

Ce protocole est une version améliorée du protocole LEACH [33] qui adopte presque les mêmes étapes que lui mais avec une méthodologie de structuration totalement différente. Cependant, sa phase d'état stable est la même que celle de LEACH. Par contre, lors de la phase initialisation, ce protocole agit différemment et essaye d'assurer une distribution uniforme de CHs.

Tout nœud du réseau transmet à la station de base des informations sur sa localisation (en utilisant GPS) ainsi sur son niveau d'énergie. Dès lors, la station de base va utiliser ces informations globales sur le réseau pour produire une meilleure structuration en clusters, en calculant le nombre optimal de ces clusters par un protocole d'optimisation, tout en se basant sur la moyenne des niveaux d'énergie. Et donc, elle va affecter pendant chaque itération, des rôles pour les différents nœuds; CH ou capteur simple. Les nœuds ayant un niveau d'énergie supérieur à la moyenne sont élus comme CHs. Dès que le nombre optimal de clusters est déterminé, la station de base annonce les identités de CHs par une diffusion de messages dans le réseau. Pour continuer le fonctionnement de ce protocole de la même manière que LEACH.

Cette version centralisée de LEACH permet une diminution remarquable de la consommation énergétique, mais en revanche elle n'est pas adaptée aux réseaux de grande dimension, due à l'overhead engendré dans le réseau suite au grand nombre de messages échangés.

2. PEGASIS: Power-Efficient Gathering in Sensor Information Systems

Dans [68], Lindsey et Raghavendra ont proposé une version améliorée de LEACH appelée PEGASIS. L'idée principale de ce protocole est de former une chaîne entre les nœuds de sorte que chaque nœud reçoit et communique à un voisin proche (les voisins proches sont localisés via l'utilisation de la force du signal reçu). Les données collectées sont agrégées et transmises d'un nœud à un autre jusqu'à ce qu'elles arrivent à un nœud particulier nommé « leader » qui les transmet à la station de base. Les nœuds qui transmettent les données à la station de base, sont désignés parmi les membres de la chaîne et sont choisis tour à tour selon un round-robin dans le but est de réduire l'énergie dépensée par un nœud durant une période (cycle). Contrairement à LEACH, PEGASIS évite la formation des clusters et procure l'envoi de données à la station de base à un seul nœud dans la chaîne.

Les résultats de simulation ont montré que PEGASIS peut prolonger de deux à trois fois la durée de vie d'un réseau de capteurs relativement à LEACH. Un tel gain de performance est réalisé par : l'élimination de l'overhead causé par le processus de formation de clusters dans LEACH (ce protocole n'utilise qu'un seul nœud dans une chaîne afin de transmettre à la station de base au lieu d'en utiliser plusieurs), ainsi que par l'utilisation de l'agrégation de données qui réduit le nombre de transmissions, de réceptions, et la largeur de la bande consommée dans la communication.

Bien que l'overhead du clustering soit évité, PEGASIS exige toujours un ajustement dynamique de la topologie puisqu'un nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données. Cependant, un tel ajustement de la topologie pourrait causer un overhead important. En outre, PEGASIS suppose que tout nœud est capable de communiquer directement avec la station de base. Or, cette supposition est loin de la réalité car les capteurs communiquent généralement en mode multi-sauts pour atteindre la station de base.

3. TL-LEACH : Two level hierarchy LEACH

TL-LEACH [57] est une extension du protocole LEACH. Il adopte une technique de rotation pour les CHs, avec l'utilisation de deux niveaux de clustering : primaires et secondaires, en plus des autres nœuds capteurs. Ce protocole est basé sur une formation aléatoire à temps de convergence constant, auto-configurable et adaptative pour les clusters. De plus, un contrôle localisé pour le transfert de données dans le réseau est aussi exercé.

La vie du réseau est alors segmentée en cycles caractérisés par un choix de différent CHs pour le premier niveau ainsi que pour le deuxième. Lors d'un cycle, quatre grandes phases peuvent être distinguées : découverte, installation des clusters, organisation et finalement la transmission de données.

- *Découverte*: chaque noeud décide selon une probabilité calculée a priori, s'il va être dans le cycle courant comme CH primaire, CH secondaire ou bien un noeud simple. Un noeud qui s'est élu comme CH primaire, annonce cette information aux autres nœuds en utilisant le protocole CSMA de la couche MAC. Egalement les nœuds qui se sont déterminés CHs secondaires communiquent cette information aux autres nœuds simples.

- *Installation* : l'organisation des noeuds dans des clusters à deux niveaux comprend la sélection des CHs primaires et secondaires, de telle sorte que chaque CH secondaire choisi le CH primaire auquel il va se rattacher et lui transmet un message de rattachement. De même, les noeuds simples prennent à leur tour une décision sur les CHs secondaires pour lesquelles des messages de rattachements seront envoyé.
- *Organisation* : chaque CH primaire a une connaissance sur ses membres , CHs secondaires et nœuds simples appartenant à son cluster, ce qui leur permet d'élaborer un plan de communication (une table TDMA) pour allouer aux différents noeuds membres des slots de temps de transmission pendant lesquelles ils peuvent émettre. En outre, un code CDMA est choisi par les CHs primaires, afin d'être utilisé par les nœuds de son cluster lors des transmissions.
- *Transmission* : les données sont communiquées à la station de base à partir des noeuds sources.

Cette transmission est faite selon deux étapes :

1. Les nœuds secondaires rassemblent les données des autres nœuds simples membres. A cette étape une fusion/agrégation de données peut être établie.
2. La collecte des données des CHs primaires à partir de leurs CHs secondaires est faite à ce niveau. La fusion de données peut être effectué à ce niveau également.

La structure hiérarchique en deux niveaux de TL-LEACH, réduit la quantité des informations qui doit être transmise à la station de base. Par conséquent, la consommation énergétique de la communication est ainsi diminuée. Avec cette technique, moins de nœuds sont nécessaires pour transmettre les données à la station de base.

Mais le problème qui se pose avec ce protocole est la non scalabilité de sa structure dans les réseaux à grande échelle. C'est vrai que l'organisation du réseau en deux niveaux présente de meilleur résultats en performance que celle à un seul niveau, mais dès que la densité du réseau s'accroît la distance vers la station de base augmente, ce qui engendre une dissipation d'énergie lors de la communication de données dans les deux niveaux de clusters et rend par conséquent, cette structure non scalable. Comme solution, on peut envisager l'utilisation de plusieurs niveaux pour les réseaux à grande échelle. De plus, ce protocole ne présente aucune restriction sur le nombre de CHs primaires et de CHs secondaires choisis dans le réseau ni leurs positions ce qui provoquent une distribution non uniforme de ces CHs (les auteurs n'ont pas donner des détails sur la façons dont les probabilités d'éligibilité sont calculées).

3.5.1.2 HEED: A Hybrid Energy Efficient Distributed Clustering Approach For Ad-hoc Sensor Networks

Younis et Fahmy ont présenté le protocole hiérarchique HEED [58]. Ce dernier est une autre variante des protocoles de clustering, il adopte une stratégie périodique pour choisir les CHs en se basant sur l'énergie résiduelle et un coût de communication telle que la proximité d'un noeud à ses voisins ou son degré (généralement utilisé comme deuxième paramètre de sélection).

Dans le processus de clustering utilisé par HEED, chaque nœud prend sa décision selon ses informations locales. Par ailleurs, les deux paramètres motionnés précédemment sont utilisés pour l'élection des CHs. Le premier critère qui est l'énergie résiduelle est employé pour la

sélection d'un ensemble de CHs initial ou expérimental. Cependant, un nœud avec une grande quantité d'énergie résiduelle a plus de chance de s'élire CH. Le second paramètre sert à résoudre les conflits; lorsqu'un nœud se trouve dans la portée de plusieurs CHs. Dans ce cas, si le niveau de puissance utilisé pour la communication intra-cluster est fixe pour tous les nœuds, alors le coût de communication est proportionnel au degré du nœud. Cela signifie qu'un nœud joint le CH avec le degré minimum pour distribuer la charge ou bien il joint celui avec le degré max pour créer des clusters denses. Sinon, des portées de transmission variables sont permises au sein d'un même cluster et chaque nœud a la possibilité de choisir le niveau de puissance approprié pour atteindre son CH, donc AMPR donné par :

$$AMRP = \frac{\sum_{i=1}^M MinPwr_i}{M}$$

Et qui représente la puissance moyenne pour la communication intra-clusters, est utilisée comme étant un troisième paramètre dans la fonction du coût de communication.

La conception de ce protocole a été faite de façon qu'il se termine en un nombre limité d'itérations $N_{iter} = O(1)$.

Durant chaque période $Tcp + Tn$ un nouveau CH est choisi où : Tcp est l'intervalle de temps du processus de clustering, et Tn est le temps qui sépare la fin d'un intervalle Tcp et le début d'un autre. Mais avant qu'un nœud ne commence l'exécution de HEED il place sa probabilité pour devenir CH comme le suivant: $CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}}$, Où C_{prob} est une probabilité supposée et

ne peut être calculée à priori. Elle détermine le nombre optimal de CHs afin de limiter les annonces de cluster-heads. $E_{residual}$: correspond à l'énergie résiduelle estimée dans les nœuds. E_{max} : Énergie maximum de référence qui correspond à une batterie entièrement chargée, généralement elle est identique pour tous les nœuds: homogènes (sinon on parle de nœuds hétérogènes avec des puissances énergétiques différentes).

Le fonctionnement de HEED est résumé en trois grandes phases décrites dans le code suivant:

- **Initialisation // Distribution expérimentale de cluster-heads**

Découvrir les voisins à la portée du nœud.

Calculer la probabilité initiale de devenir cluster-head : $CHprob = f(\frac{Er}{Emax})$

Annonce l'information de devenir CH candidat dans le rayon de sa communication.

- **Répétition // Élection itérative et équilibrage de charge**

Si v a reçu des annonces de CHs, choisir un avec le coût minimal // v : un nœud ordinaire
Si v n'a pas de CH, il décide de devenir CH avec $CHprob$.

$CHprob = \min(CHprob * 2, 1)$

Répéter jusqu'à ce que $CHprob$ atteigne 1.

- **Finalisation // Achèvement et établissement de clusters et CHs.**

Si un CH est trouvé, se joindre à son cluster.

Sinon, décider de devenir CH.

Une fois les clusters sont formés, les CHs affectent aux membres de leurs groupes des tranches de temps différentes pendant lesquelles ils peuvent transmettre leurs données.

Après la collecte des données, les CHs acheminent l'information agrégée directement vers la station de base via un seul saut. Ces CHs peuvent alternativement communiquer avec l'un ou l'autre, pour agréger leur informations par des multiples sauts vue la connectivité assuré par le protocole HEED.

Discussion

HEED présente plusieurs arguments en sa faveur. C'est un protocole non centralisé convergeant; le clustering se termine dans $O(1)$ itérations. Il prolonge la durée de vie du réseau en distribuant l'énergie de la communication et le nombre de CHs d'une façon uniforme et donc produire des clusters compact tout en minimisant la charge des messages de contrôle. Son processus d'itérations est indépendant de la topologie ou de la taille du réseau. En outre, il n'indique aucune supposition sur la distribution ou la densité des noeuds, ainsi que leurs capacités.

Le fait, que le choix des CHs est une décision qui ne se base que sur des informations locales, des insuffisances dans la fonction du coût seront présentes, telle le cas de la communication inter-clusters qui n'est pas prise en considération par cette fonction. L'étendue du réseau est posée comme un autre problème d'adéquation de ce protocole pour les réseaux à grande échelle, vu la stratégie adoptée pour la communication entre les CHs est la station de base qui se fait via un seul saut, sans oublier l'overhead produit dans la deuxième phase (étape de répétition) avec un nombre $N_{iter} \times N$ itérations inutiles. D'autre part, les clusters générés avec HEED ne sont pas tellement équilibrés en taille.

3.5.1.3 EECS: Energy Efficient Clustering Scheme in WSN

Dans EECS [59], les auteurs ont proposé une nouvelle technique pour l'élection des CHs et la formation de clusters. Selon cette technique, les CHs candidats doivent rentrer en concurrence pour devenir des CHs. Ces CHs candidats sont choisis parmi les noeuds du réseau selon une probabilité prédéfinie.

Dans la phase d'élection de CHs, un nombre constant de CHs candidats est déterminé. Par la suite ces CHs candidats diffusent des messages à leurs voisins via le support radio selon leur rayon de compétition, annonçant ainsi leurs volontés de devenir des CHs. A la fin, le CH ayant plus d'énergie résiduelle est choisi.

La formation des clusters adoptée par ce protocole est bien différente que celle de LEACH. Avec EECS chaque noeud choisi le cluster auquel il va se joindre selon deux paramètres : la distance qui le sépare du CH, et selon la distance qui sépare le CH de la station de base.

Les tailles des clusters seraient dynamiques, et donc ce protocole permet de construire des groupes qui ont un éventail plus large dès qu'on s'éloigne de la station de base. Par conséquent, ces groupes vont exiger plus d'énergie pour la transmission de données que ceux, qui sont plus proches de la station de base.

Discussion

Ce protocole permet d'utiliser des tailles dynamiques pour les clusters. Il offre une faisabilité similaire à celle de LEACH en termes de mécanisme de recouvrement, avec une meilleure utilisation d'énergie dans le réseau et donc assurer une meilleure connexion. D'ailleurs, les

résultats de simulation d'EECS ont présentés de meilleures performances que LEACH, en termes de conservation d'énergie et prolongation de la durée de vie du réseau.

Mais malgré ça, on peut rapprocher à ce protocole sa méthodologie de communication employée pour router les données à la station de base. Généralement un seul saut est utilisé. Ceci consomme beaucoup d'énergie par les noeuds distants pour atteindre la station de base.

3.5.1.4 EEUC: An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks

Li et al ont proposé le protocole EEUC [60] qui aborde le problème des points chauds 'hot spot': les CHs qui entourent la station de base vont mourir plus vite que les autres CHs, car tout le trafic du réseau destiné à la station de base passe par ces noeuds et donc ont plus de charge que les noeuds distants, en particulier lors de l'adoption du routage multi-sauts comme mécanisme de transmission. L'idée principale derrière EEUC est d'utiliser un mécanisme de partitionnement de taille inégale pour grouper les noeuds lors du processus de clustering afin d'équilibrer la consommation d'énergie des CHs, et résoudre le problème des points chauds. Pour ce faire, les clusters les plus proches de la station de base ont des tailles plus petites que ceux qui sont plus loin, ce qui va permettre une diminution de la charge des CHs entourant la station de base et donc leur offrir la possibilité de conservation d'énergie.

EEUC procède d'une manière concurrente pour choisir les CHs. Initialement, tous les noeuds ont la même probabilité pour devenir des CHs candidats, où chaque noeud tire un nombre aléatoire entre 1 et 0. Si le nombre tiré est inférieur à un certain seuil prédéfini T , le noeud s'endormira. Sinon, il devient CH candidat et diffuse par la suite un message de compétition pour annoncer sa décision. Ce message contient l'identité du noeud, le diamètre du rayon de concurrence et son niveau d'énergie résiduelle. Le diamètre du rayon de concurrence est calculé en fonction de sa distance à la station de base.

La figure 3.3 illustre une topologie de clusters où les cercles représentent les différents diamètres de concurrence. En fonction de ces diamètres, une répartition contrôlée de CHs dans le réseau est exercée.

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

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

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

Une fois les CHs finaux sont sélectionnés, ils diffusent des messages d'annonce à travers le réseau. Et selon la force du signal reçu, chaque noeud rejoint le cluster du CH le plus proche, tout en confirmant son appartenance par un message de jointure.

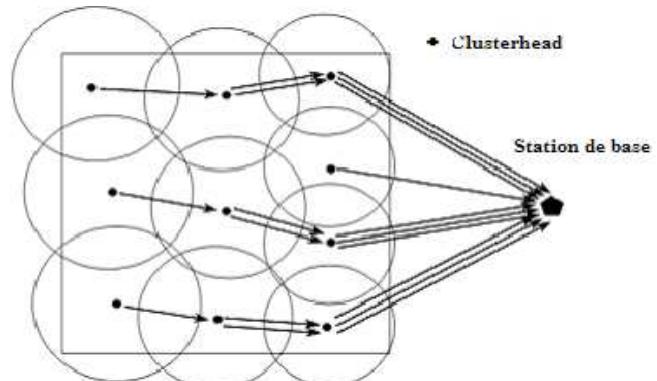


Fig. 3.3 Le mécanisme utilisé par EEUC.

Pour organiser la communication intra-clusters EEUC adopte le même mécanisme utilisé dans LEACH (combiné les deux protocoles TDMA et CDMA pour collecter les données des noeuds membres). Or que pour le routage inter-clusters, une nouvelle méthode est introduite. Selon un seuil calculé, les CHs envoient leurs données à la station de base. Si la distance qui leur séparent à la station de base est inférieur à ce seuil, les données sont transmises directement à cette dernière, sinon les CHs devront trouver le meilleur noeud relais (le plus adjacent et celui doté de plus d'énergie résiduelle) pour router les données et y parvenir à la station de base.

Discussion

EEUC réalise un équilibrage de la consommation d'énergie entre les CHs via la technique de formation de clusters adoptée; clusters plus proches de la station de base ont des tailles plus petites que ceux éloignés. De plus, à travers le mécanisme de routage multi-sauts introduit une meilleure préservation de la ressource énergétique est offerte dans le réseau; moins d'énergie est consommée pour atteindre la station de base.

En contre partie, le fait de choisir les plus proches CHs adjacents est ceux dotés de plus d'énergie résiduelle comme noeuds relais lors du routage multi-sauts sans prendre en compte la taille des clusters dans la fonction du choix ni le nombre de fois que ces CHs ont été déjà choisi comme relais, pourraient causer beaucoup de charge supplémentaire à ces CHs ce qui épuise leurs batteries rapidement. Ainsi, le nombre de messages de communication généré dans le réseau est très élevé car les CHs relais ne sont responsables que du transfert de données vers la station de base ou vers d'autres CHs relais et donc aucune fonction d'agrégation de données issue des autres CHs n'est réalisée ce qui va augmenter le nombre de messages échangés dans le réseau. Comme solution possible : introduire des mécanisme d'agrégation de données au niveau des CHs relais pour diminuer le nombre et la taille des messages transférés vers la station de base.

3.5.2 Protocole hiérarchique à temps de convergence variable

Le temps est un facteur important dans la convergence des protocoles de clustering. Certains de ces protocoles tels que RCC [63] et CLUBS [64] ont $O(n)$ comme temps de convergence, où n représente le nombre de capteurs dans le réseau. Il est donc plus pratique de mettre en œuvre ce type de protocole pour les réseaux de capteurs de petite ou moyenne taille car ce temps de convergence variable va permettre à ces protocoles plus de contrôle pour les propriétés de clusters que ceux à temps de convergence constant.

3.5.2.1 EEHC : Energy Efficient Hierarchical Clustering

EEHC proposé par Bandyopadhyay et Coyle [62] est un protocole aléatoire avec un temps de convergence variable. Ce protocole s'exécute sur deux phases : initial et prolongée.

Phase initial (appelée aussi clustering à un seul niveau) : chaque noeud s'annonce comme un CH, avec une probabilité p_1 , aux autres noeuds voisins situés dans sa portée de communication. Ces CHs sont nommés les volontaires. Tous les noeuds qui sont à k sauts d'un CH reçoivent cette annonce, soit par communication directe ou par retransmission. N'importe quel noeud qui reçoit un tel message et qui n'est pas un CH, s'affilie au plus proche CH et devient un noeud membre dans son cluster. Si l'annonce ne parvient pas à un noeud au bout d'un intervalle de temps préréglé et calculé en fonction de la durée d'un paquet pour atteindre un capteur à k sauts, ce

noeud qui n'est pas CH et qui n'est à la portée d'aucun des CHs volontaires et ni membre d'aucun cluster est forcé à être CH.

Phase prolongée: le processus de clustering est prolongé afin de permettre la construction de plusieurs niveaux, c'est-à-dire la structuration de H niveaux d'hiérarchie. Les CHs répètent récursivement l'exécution du processus de clustering afin de former un palier supplémentaire de clusters. Et pour transmettre les données à la station de base, ce protocole essaye d'assurer la connectivité en H sauts entre les CHs et la station de base. H est supposé comme le niveau le plus haut. Pour ce faire, EEHC procède comme suit : les nœuds capteurs transmettent les données collectées au CHs de niveau 1 (niveau le plus bas). Les CHs du niveau 1 transmettent à leur tour les données agrégées aux CHs de niveau 2 et ainsi de suite jusqu'au niveau le plus supérieur de la hiérarchie des clusters. Une fois arrivée, les CHs de haut niveau transmettront les rapports de données à la station de base. Cela permet à EEHC d'avoir une complexité temporelle de O ($k_1 + k_2 + \dots + k_h$). Cette complexité est considérée comme amélioration importante par rapport aux nombreux protocoles de clustering qui ont O (n) comme temps de convergence.

Discussion

EEHC permet de conserver la ressource énergétique au niveau du réseau en réalisant une meilleure complexité temporelle, comparé aux autres travaux (voir RCC et CLUBS, protocoles détaillés plus tard, par exemple). Ce protocole essaye d'assurer un certain équilibrage de charge car chaque nœud dans le réseau a une chance de devenir CH. De plus, la communication réalisée par EEHC est meilleure qu'une communication directe avec la station de base. Cela est grâce à l'adoption de plusieurs sauts pour atteindre la station de base, ce qui permet d'économiser l'énergie lors de la communication des données. Il faut noter que la collecte des données et la transmission agrégée des rapports à la station de base dépendent des paramètres p et k adoptés par le protocole EEHC.

Cependant, le niveau de CHs a un impact direct sur l'énergie : plus le niveau est élevé plus la consommation d'énergie augmente car les CHs d'haut niveau dépensent plus d'énergie pour la construction et la réexpédition des annonces. Par conséquent, les CHs de haut niveau mourront plus rapidement que les autres nœuds. En outre, lors d'appels fréquents pour la sélection du CH, le réseau en entier est instable et donc plus d'énergie est gaspillée pour l'entretien de la topologie et l'opération de maintenance s'avère un peu coûteuse.

3.5.2.2 RCC : Random Competition based Clustering

Bien que RCC [63] soit conçu pour les réseaux ad-hoc, il est également applicable aux réseaux de capteurs. Ce protocole se focalise principalement sur la stabilité des clusters. Il applique un principe basé sur la victoire des nœuds qui ont les premières déclarations. De ce fait, cette stratégie permettrait à n'importe quel noeud de déclencher le processus de formation de clusters dans le réseau. Or, un nœud qui ne fait partie d'aucun cluster se déclare comme étant CH et diffuse cette information à ses voisins. Le premier nœud qui annonce cette information sera élu par ses voisins comme CH. Après avoir entendu cette déclaration, les nœuds voisins envoient leurs accords et joignent ce CH comme noeuds membres. Des paquets d'annonces de CHs sont relancés périodiquement afin de maintenir les clusters. Le délai entre la diffusion d'un paquet d'annonce et sa réception peut créer des conflits, beaucoup de nœuds voisins peuvent émettre des paquets d'annonce de CHs; concurremment.

Pour pallier à un tel conflit, RCC utilise explicitement un temporisateur aléatoire. Chaque nœud dans le réseau remis à zéro son compteur de valeur aléatoire avant de se déclarer CH. Cette

solution de temporisateur aléatoire n'est pas complète car le problème de conflits peut persister. Pour le résoudre, ce protocole fait recours à l'identité des nœuds : celui ayant le plus petit identifiant deviendra CH.

Discussion

Le choix aléatoire du CH ne prend pas en considération son niveau d'énergie résiduelle. Par conséquent un nœud à basse puissance énergétique peut être choisi comme CH et donc sa durée de vie limitée, limitera par la suite la durée de vie du réseau en entier.

En outre, le fait de relancer le processus de formation pour remédier aux différents conflits consomme à son tour la bande passante et l'énergie. De plus, l'utilisation du critère d'identité comme deuxième paramètre limitera la scalabilité et l'extension du réseau.

RCC n'admet aucune limite sur les tailles des clusters générés. Par exemple dans les réseaux denses tel que les réseaux de capteurs, la taille des clusters peut être importante ce qui peut épuiser rapidement les batteries des CHs, et par la suite causer des goulets d'étranglement dans le réseau.

3.5.2.3 CLUBS : Algorithm For Group Formation In An Amorphous Computer

Nagpal et Coore ont proposé le protocole CLUBS [64], un protocole de clustering qui converge en un temps proportionnel à la densité des noeuds. Il partitionne le réseau en clusters

selon la diffusion locale tout en assurant trois caractéristiques :

-Dans le réseau, chaque nœud doit être connecté à un cluster.

-Tous les clusters du réseau devront avoir le même diamètre maximum.

-Les nœuds dans un cluster doivent être en mesure de communiquer les uns avec les autres.

Avec ce protocole, la formation des clusters se fait avec un maximum de deux sauts. Chaque noeud dans le réseau participe à ce processus de formation tout en choisissant un nombre aléatoire dans un intervalle fixe de nombres entiers. Le noeud commence à décompter à partir de ce nombre. Si le compte à rebours n'a pas été interrompu par un autre noeud voisin et atteint le zéro, ce noeud se déclare CH et diffuse un message d'invitation à ses voisins.

Une fois que ses voisins à deux sauts reçoivent le message diffusé, ils arrêtent le compte à rebours, acceptent l'invitation et se joignent au cluster. Un noeud qui a rejoint un cluster est appelé 'membre' et il n'est plus autorisé à concourir pour devenir CH.

Ces nœuds membres peuvent rester à l'écoute des messages d'invitations additionnels et donc ils ont la possibilité d'être membres de plusieurs clusters. Par ailleurs, si un noeud qui est en concurrence pour devenir CH détecte une collision ou reçoit un message défectueux, devient un noeud membre et suppose que plusieurs CHs ont tenté de l'atteindre au même moment. Son CH sera découvert plus tard. Par conséquent, ce protocole ne se termine pas à moins que tous les noeuds dans le réseau joignent un certain cluster comme CHs ou comme nœuds membres.

Discussion

L'efficacité et la simplicité de CLUBS lui permettre d'être implémenté dans des réseaux de capteurs asynchrones car le mécanisme d'élection de CH et celui de formation de clusters ne

nécessite aucune synchronisation entre les noeuds. En outre, ce protocole assure de nombreuses contraintes qui sont communes à d'autres environnements distribués, telles la connaissance limitée de la topologie ou l'accès aux identifiants globales.

L'inconvénient majeur de CLUBS se résume dans les clusters ayant leur CH à un saut dans la portée des autres clusters. Si tel est le cas, les deux clusters vont s'effondrer et le processus de sélection de CHs va être relancé à nouveau. Par ailleurs, dans un environnement où la cause principale de la défaillance des capteurs est l'épuisement des batteries, il est souhaitable d'équilibrer la consommation d'énergie parmi les nœuds du réseau. Pour cela, la sélection des CHs devrait se faire sur la base de plusieurs facteurs comme l'énergie et la densité du réseau.

3.5.2.4 Hierarchical Control Clustering

À la différence de la plupart des travaux déjà vu, l'objectif de Hierarchical Control Clustering est la formation de plusieurs niveaux hiérarchiques de clusters ; clusters de clusters [65].

La figure 3.4 illustre la notion d'hiérarchie de clusters. Lors du processus de clustering, certaines propriétés utiles pour la gestion et la scalabilité de la hiérarchie tels que la taille du groupe et le degré de chevauchement sont pris en considération.

D'après le schéma proposé, n'importe quel noeud dans le réseau peut lancer le processus de formation de clusters. Si plusieurs nœuds commençaient le processus de formation de clusters en même temps, le noeud initiateur avec le plus petit identifiant 'ID' sera favorisé.

Ce protocole procède en deux phases : arbre de découverte et formation de clusters.

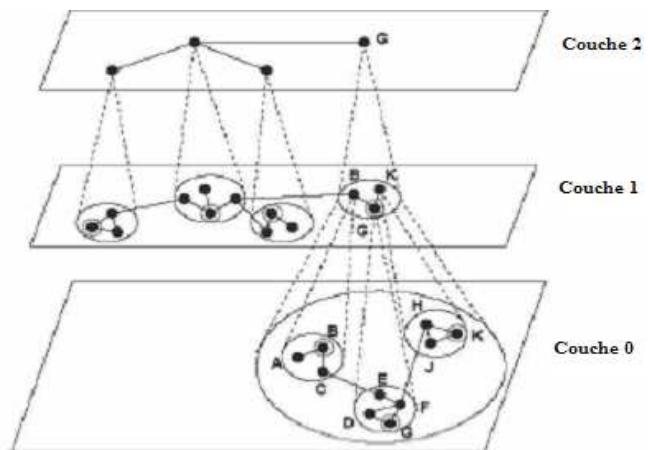


Fig.3.4 Hiérarchie de clusters.

La phase d'arbre de découverte est une formation distribuée selon BFS : Largeur-First-Search⁹ dont la racine est le nœud initiateur. Chaque nœud diffuse un signal, une fois tous les p unités de temps, annonçant des informations relatives aux sauts de plus courte distance à la racine. Dans le cas où un nœud voisin représente le plus court chemin vers la racine pour un autre nœud source, ce nœud voisin sera choisi comme parent et par conséquence le nœud source mettra à jour sa distance de saut vers la racine. Le signal de diffusion annonce des informations sur l'identité de source, ID de parent, ID de la racine, et la taille du sous-arbre. Ce dernier doit être mis à jour par chaque noeud quand la taille de ses fils change.

Quand un sous-arbre d'un noeud atteint un paramètre de taille k, la phase de formation de clusters est entamée. Alors, cette phase de formation de clusters est lancée par le noeud racine sur son sous-arbre. Un seul cluster est construit pour l'ensemble du sous-arbre, si sa taille est < 2k. Dans le cas contraire, de multiples clusters seront formés. Après la phase de création de clusters, le maintien des informations est crucial pour les clusters, ainsi que pour la maintenance des arbres BFS.

⁹ Permet le parcours d'un graphe de manière itérative, en utilisant une file FIFO dans laquelle il prend le premier sommet et place en dernier ses voisins non encore explorés tout en marquant les sommets déjà visités pour que l'algorithme puisse se terminer. En ligne http://fr.wikipedia.org/wiki/Algorithme_de_parcours_en_largeur.

Discussion

Hierarchical Control Clustering est généralement utilisé pour gérer des environnements dynamiques, par exemple : la présence de noeuds mobiles dans le réseaux. Ce protocole permet de construire k-clusters limités par leur taille car la cardinalité de ces clusters à la fin de l'exécution du processus de clustering est compris entre k et $(2 * k - 1)$. Cependant, l'échange excessif d'informations pour la formation ou le maintien des clusters consomme de l'énergie et limite la durée de vie du réseau. De même, la phase de construction de l'arbre recouvrant peut créer un trafic de contrôle important qui se répercute sur le temps de convergence du protocole.

3.5.3 Protocoles heuristiques

La sélection d'un ensemble optimal de clusters est équivalente au problème d'optimisation qui est un problème NP-complet. Plusieurs protocoles de clustering proposés dans la littérature sont basés sur différentes heuristiques. Une heuristique est un algorithme de résolution ne fournissant pas nécessairement une solution optimale pour un problème d'optimisation donné. Souvent, un algorithme heuristique a les objectifs suivants: trouver un algorithme avec un temps raisonnable (le temps favorable pour la construction de clusters) et essayer aussi de trouver la solution optimale (essayer de fournir la solution la plus proche de l'optimum). La plupart des protocoles heuristiques ont une complexité temporelle de $O(n)$, où n est le nombre total de noeuds. Ces protocoles permettent d'avoir des performances judicieuses. En plus, ils ne sont pas basés sur des métriques particulières car les CHs sont choisis selon différentes contraintes [43].

3.5.3.1 LCA2 : Linked Cluster Algorithm

LCA2 [66, 69] présenté par Baker et Ephremides est proposé comme amélioration à LCA [66, 67] l'un des plus anciens protocoles de clustering utilisé dans les réseaux ad-hoc.

LCA2 est un protocole essayant de satisfaire les objectifs suivants : la formation de clusters recouvrant assurant une meilleure connectivité des noeuds, et améliorer la durée de vie du réseau.

Un identifiant 'ID' unique est assigné à chaque noeud. Par la suite, chaque noeud diffuse son identité, et en écoutant les transmissions des autres noeuds construisent sa liste de voisins.

Avec ce protocole, un nouveau concept est introduit: noeud "couvert" ou "non couvert". Un noeud est dit couvert s'il se trouve dans le voisinage d'un saut d'un noeud s'est déclaré CH candidat. Un noeud est choisi CH s'il a la plus petite identité parmi les voisins non couverts qui se trouvent à un seul sauts (voisins n'appartenant à aucun CH).

Les communications intra-cluster sont gérées par le protocole TDMA de la couche MAC, où un slot de temps est affecté à chaque noeud membre du cluster. Un noeud membre ne peut envoyer ses données à son CH que pendant son slot de temps. Le CH envoie ses données à son tour à la station de base.

Discussion

LCA2 permet de construire des clusters recouvrant à un seul saut en produisant un nombre réduit de CHs dans le réseau.

Toutefois, le fait de se baser uniquement sur l'identité des noeuds pour sélectionner les CHs cause le choix de noeuds ayant de faible énergie car il suffit d'avoir le plus petit identifiant dans son voisinage pour devenir CH (même si sa réserve d'énergie est faible). De plus, les CHs consomment en moyenne plus d'énergie que les autres membres dans un cluster, puisqu'ils sont responsables de la coordination entre ses membres, l'agrégation de leurs données collectées et la transmission vers la station de base. Ainsi, si ces CHs gardent leur statut de CH pour une longue durée, il en résulte qu'ils consomment rapidement leurs énergies, et par conséquent dégradent les performances du réseau.

3.5.3.2 Highest Connectivity Cluster Algorithm

Afin de maximiser la connectivité du réseau, les auteurs du protocole : Highest Connectivity Cluster Algorithm [66] essayent de construire des clusters recouvrant (non disjoints).

Un identifiant unique est assigné à chaque capteur qui diffuse par la suite un message pour reconnaître le nombre de ses voisins (son degré). Et donc, le degré des nœuds est considéré comme paramètre de choix pour l'élection des CHs : le noeud ayant le plus fort degré dans son voisinage se déclare CH. Dans le cas d'un conflit (deux nœuds voisins ont le même degré) le nœud ayant le plus petit identifiant deviendra CH.

Discussion

Le protocole Highest Connectivity Cluster génère un nombre réduit de clusters puisqu'il favorise les nœuds ayant le plus fort degré pour être des CHs c'est-à-dire les nœuds qui couvrent plus de nœuds voisins.

Bien que ce protocole assure une bonne connectivité des nœuds dans le réseau, il ne peut être utilisé que dans les réseaux de petite ou moyenne taille (il est difficile d'assurer la connectivité quand le nombre de nœuds augmente). En outre, avec ce protocole, un noeud déjà élu CH a tendance à garder le statut de CH plus longtemps car il a une forte probabilité pour qu'il soit élu plus souvent CH par rapport à d'autres noeuds.

3.5.3.3 Max-Min-D Cluster

Dans Max-Min-D cluster [69], une nouvelle procédure est proposée pour l'élection de CHs. Cette dernière est développée pour étendre la notion de clusters à un seul saut et donc étaler la formation de clusters à d-sauts (construire d-clusters) où d est un paramètre de l'heuristique dont le nombre de clusters formés dépend. Les règles heuristiques utilisées par ce protocole permettent de converger rapidement vers une solution; au bout de $2d$ cycles d'échange de messages. Max-Min D-cluster utilise la métrique ID (identifiant) pour élire les CHs. Il s'exécute sur trois phases.

Lors de la première phase appelée aussi 1^{er} d-round, chaque nœud diffuse la valeur WINNER qui représente son identifiant à ses d-voisins (voisins se trouvant à d sauts de lui) et assigne à cette valeur le plus grand ID entendu parmi ses d-voisins. Puis, il diffuse cette nouvelle valeur WINNER à ses d-voisins (procédure Floodmax).

Au cours de la deuxième phase (2^{ème} d-round), chaque nœud garde le plus petit ID(WINNER) parmi ceux qu'il a reçu durant cette phase (le plus petit ID parmi les plus grands ID diffusés; procédure Floodmin).

La troisième phase consiste à choisir les CHs en se basant sur les identifiants enregistrés lors des deux phases précédentes (1^{er} d-round et 2^{ème} d-round). Un nœud **u** se déclare CH s'il reçoit

son identifiant ID (my_ID = WINNER) lors de la deuxième phase de la diffusion (2^{ème} d-round). Sinon, si le nœud **u** a reçu WINNER durant chacune des deux phases (1 et 2), alors il choisira le nœud dont l'identifiant est attribué à la valeur WINNER comme CH (WINNER correspond à la plus petite valeur reçue par le nœud **u**). Sinon, **u** choisit comme CH le nœud ayant le plus grand ID dans son d-voisinage.

Discussion

Ce protocole permet la création de clusters disjoints c'est-à-dire un nœud ne peut appartenir qu'à un seul CH et les clusters formés se trouvent à d-sauts loin du CH (d est une valeur sélectionnée pour l'heuristique). Par conséquent, l'équilibrage de charge entre les CHs est assuré. Cependant, cette technique augmente considérablement l'overhead du réseau puisqu'elle utilise un nombre important de messages échangés pour élire les CHs, ce qui induit à une latence non négligeable. En outre, l'utilisation de la métrique ID par Max-Min-D Cluster produit des CHs susceptibles de jouer leur rôle de CHs pour très longtemps puisque leurs identités ne vont pas changer (les CHs restent pour longtemps CHs). Par contre, les clusters ne restent pas pour autant adaptés, et alors la naissance de nouveaux clusters est plus visible plutôt que l'adaptation des anciens.

3.5.4 Protocoles pondérés

Plutôt que d'utiliser une seule métrique pour élire les CHs, certains protocoles de clustering proposés dans la littérature utilisent une somme pondérée de plusieurs paramètres pour l'élection de CHs. Ces protocoles associent un poids à chaque nœud. Ce poids est représenté par une somme pondérée des différentes métriques impliquées dans son calcul comme montré dans cette équation :

$$\text{Poids}(u) = \sum_{i=1}^k \alpha_i * P_i \quad \text{avec} \quad \sum_{i=1}^k \alpha_i = 1$$

Où :

α_i représente le coefficient de pondération (le degré d'implication de la métrique) et P_i la valeur de la métrique.

Le coefficient de pondération de chaque métrique dépend de l'application et reflète son degré d'implication dans le calcul du poids. Par exemple, dans les réseaux de capteurs où l'énergie est une ressource précieuse, il est nécessaire de faire associer à la métrique d'énergie résiduelle un coefficient de pondération très élevé.

3.5.4.1 WCA : Weighted Clustering Algorithm

Le protocole WCA [70] a motivé le développement de nombreux protocoles de clustering contemporain. Ce protocole a été proposé à l'origine pour les réseaux mobiles ad hoc, car la mobilité est l'une des métriques impliquées dans le calcul du poids. WCA combine trois autres métriques avec la mobilité pour calculer le poids attribué à chaque noeud. Ces métriques sont : le degré d'un noeud, puissance de transmission et le temps de service en tant que CH tel que : $W_v = w_1\Delta_v + w_2D_v + w_3M_v + w_4P_v$ avec $w_1 + w_2 + w_3 + w_4 = 1$

La première métrique : Δ_v représente la différence entre le degré d'un nœud **u** et une valeur seuil prédefini. Ce seuil est présumé être la taille idéale d'un cluster car chaque capteur est capable de gérer un nombre limité de noeuds au sein de son cluster en tant que CH. D_v est la distance

moyenne des voisins calculée à l'aide d'un GPS est strictement liée à la puissance énergétique consommée dans les transmissions. La mobilité relative M_v , est le rapport des niveaux de puissance des transmissions successives reçues par un nœud de ses voisins. Ainsi, le nœud ayant la plus faible mobilité dans son voisinage sera un bon candidat pour être CH puisqu'il gardera un voisinage plus stable au cours du temps et favorisera la stabilité des clusters. Le temps passé par un noeud à assumer la charge de CH depuis l'installation du réseau est comptabilisé. Ce temps représente la quatrième métrique dans la fonction du calcul de poids.

Le nœud ayant le plus petit poids dans son voisinage: la plus petite somme des distances, la plus faible mobilité et le plus petit temps en service en tant que CH, devient CH. Cette élection est assurée par un processus itératif initial où chaque nœud diffuse son poids et maintient une liste des poids reçus de ses voisins. Les informations stockées sont à nouveau échangées avec les voisins immédiats et le processus continu jusqu'à ce que tous les noeuds aient des informations sur les poids de tous les autres noeuds du réseau. Ceci leur permet de connaître le noeud qui a le plus petit poids. Une fois le CH déterminé, ses voisins s'y rattachent dans la limite de Δ_v et le processus est répété pour les itérations suivantes avec les capteurs non rattachés. Ainsi, la fonction pondérée est mise à jour par ces derniers tout en ignorant les capteurs attachés. Par la suite, les nouveaux poids mis à jour sont diffusés sur le réseau. Le temps requis pour que le processus converge est en fonction de la taille du réseau.

Discussion

Le fait de combiner plusieurs métriques dans le calcul du poids utilisé lors de l'élection des CHs, permet à WCA de produire des clusters recouvrant et de présenter de meilleures performances en termes de nombre de clusters produits et nombre de changements de CHs. En contre partie, l'exigence de WCA pour connaître le poids de tous les nœuds avant de commencer le processus de clustering induit à un overhead très élevé et il épouse rapidement les batteries des noeuds. De plus, l'implication de la métrique de mobilité dans la fonction de calcul de poids restreint l'utilisation de ce protocole pour un certain type de réseaux dans lesquels les capteurs sont mobiles.

3.5.4.2 DCA : Distributed Clustering Algorithm

DCA [56] est un protocole destiné aux réseaux dans lesquels les nœuds sont immobiles, ou se déplacent avec une faible vitesse. Ce protocole considère que chaque noeud a un poids unique et les CHs sont choisis à la base des poids assignés aux nœuds. Le poids est défini par l'inverse de la vitesse, afin de favoriser les nœuds les moins mobiles fournissant des CHs plus stables. Un nœud u est choisi pour être CH s'il possède le plus grand poids dans son voisinage. Autrement, il joint le CH qui se trouve dans son voisinage. DCA opère selon deux phases : formation de clusters et la maintenance des clusters générés.

Formation de clusters : DCA suppose que les informations de voisinage à un seul saut sont disponibles (obtenues par l'exécution d'une itération initiale de découverte du voisinage). De ce fait, les nœuds possédant le poids le plus élevé dans leur voisinage sont élus comme CHs et annoncent par la suite cette information à leurs voisins. Un nœud de poids plus petit ne peut prendre une décision sur le rôle qu'il pourra jouer, qu'une fois que tous ses voisins ayant le plus grand poids ont pris leurs décisions. Dans ce cas, le noeud attend les messages d'annonces que ces derniers pourraient les diffuser. Une fois qu'un nœud reçoit les messages annoncés par les noeuds de poids élevé s'y rattache au CH possédant le plus grand poids parmi ceux reçus et

confirme son appartenance autant que membre de son cluster par l'envoie d'un message de rattachement. Si aucun de ses voisins de poids élevé n'a pris la décision de jouer le rôle de CH, alors ce noeud devient CH.

Maintenance : en raison de la mobilité ou de l'échec des nœuds, cette phase de maintenance nécessite de reconfigurer la structure des clusters et relancer le processus de clustering à nouveau.

Discussion

DCA est un protocole qui offre la possibilité de supporter les nœuds mobiles se déplaçant avec une vitesse faible. En supposant que la topologie du réseau ne change pas pendant l'exécution du processus de clustering, ce protocole permet une construction de clusters à un seul saut parce que chaque nœud ne diffuse les messages que dans un voisinage à un seul saut.

Cependant, cette technique de formation de clusters n'utilise que des informations partielles sur le réseau (voisinage à un seul saut), et donc les clusters générés peuvent ne pas être optimales. De plus, cette technique itérative de clustering est une méthode trop sensible à la perte des paquets ou l'échec des noeuds. Par exemple, si un nœud v découvre que l'un de ses voisins u a un poids plus élevé que lui, alors il doit attendre jusqu'à ce que u décide pour qu'il puisse prendre sa décision. Le temps d'attente est en fonction du nombre de noeuds dans son voisinage. Ceci influera sur la vitesse de convergence de ce protocole qui dépendra dans ce cas, du diamètre du réseau ce qui rend l'utilisation de DCA souhaitable pour les réseaux de petite taille.

La méthode d'attribution des poids aux nœuds n'a pas été discutée dans l'protocole et il n'y a aucune optimisation sur les paramètres du système tels que le nombre de paquets échangés et le contrôle de la puissance énergétique dépensée.

3.5.5 Protocoles émergents

Un protocole émergent est n'importe quel calcul qui réalise formellement ou atteint stochastiquement des effets globaux prévisibles en communiquant seulement avec un nombre limité de voisins immédiats et sans l'utilisation de contrôle centralisé ou de visibilité globale. Par conséquent, un protocole émergent pour un réseau de capteurs est un protocole dans lequel la propriété globale désirée, ni n'est explicitement codée dans le protocole ni organisée par une autorité centrale, mais elle émerge en raison de l'interaction et de la rétroaction locales répétée entre les noeuds.

3.5.5.1 ACE: An Emergent Algorithm for Highly Uniform Cluster Formation

ACE [71] est un protocole utilisant l'émergence dans son processus de clustering. L'idée principale de ce protocole est de permettre à un nœud d'évaluer son potentiel en tant que CH. Pour un moment donné, un noeud soit est choisi comme étant un CH où bien il n'est plus CH et se retire devant le meilleur candidat CH. La sélection réalisée entre les CHs est basée sur un poids. Ce poids et en fonction de nombre de membre fidèles. Un membre est dit "fidèle" s'il n'est membre que d'un seul cluster. ACE fonctionne en plusieurs itérations asynchrones et locales à chaque nœud. Il s'exécute sur deux phases : formation de clusters et celle de migration.

Formation de clusters : dans cette phase de nouveau clusters sont générés sur la base d'une autoélection de noeuds. Le noeud désirant devenir CH diffuse un message d'annonce à ses voisins. Après avoir reçu le message d'annonce, un voisin s'affilie au CH et devient un membre dans son

cluster. À tout moment, un noeud peut être un membre de plus d'un cluster. Dans ce cas ce noeud est dit "clustered". Toutefois, il peut être un membre fidèle (n'appartient qu'à un seul cluster). Un noeud qui ne reçoit aucun message d'annonce est dit "unclustered" c'est-à-dire: n'appartenant à aucun cluster. Ce noeud unclustered engendre lui-même un nouveau cluster en se déclarant CH chaque fois qu'il estime qu'il peut gagner au moins un certain nombre de noeuds fidèles. Cette estimation est une fonction qui dépend du temps écoulé depuis que le processus de formation a été lancé par ce noeud.

Migration : lors de cette phase, les meilleurs candidats CHs sont sélectionnés. Chaque CH vérifie périodiquement la capacité de ses voisins d'être CHs afin de choisir le meilleur candidat pour jouer le rôle du nouveau CH. Cette sélection est faite selon le nombre de noeuds fidèles. De ce fait, le noeud qui a le plus grand nombre de noeuds fidèles dans son cluster et moins de chevauchement avec les clusters existants est considéré comme le meilleur CH candidat. Une fois que le meilleur CH candidat est déterminé par le CH actuel du cluster, ce dernier va se retirer et sa position de CH va migrer dans la direction du nouveau CH choisi. Par conséquent, certains membres de l'ancien cluster peuvent ne plus appartenir au nouveau cluster. Comme de nouveaux noeuds à proximité du nouveau CH vont devenir des membres dans son cluster.

Discussion

Une formation de clusters très uniforme résulte de l'utilisation du protocole ACE. Ce protocole fonctionne sans avoir besoin de la connaissance géographique des positions des noeuds, où de n'importe quel genre d'évaluation de distance entre les capteurs. ACE suppose que tout le réseau peut être couvert juste en trois cycles, tout en utilisant des communications intra-clusters. Ceci lui permet d'être indépendant de la taille du réseau (quelque soit le nombre total de noeuds dans le réseau). Il est à noter que ce protocole converge en un temps constant $O(d)$ où d est la densité des noeuds par unité de cluster.

Cependant, les auteurs d'ACE n'ont pas présenté des détails sur leur choix de trois cycles comme étant nombre total d'itérations permettant de couvrir tout le réseau. De même, le nombre total de clusters engendré par ce protocole est très élevé et donc les chevauchements entre leurs membres sont inévitable, malgré que les auteurs ont essayé de le diminuer par la technique utilisée dans la deuxième phase de migration.

C'est vrai qu'ACE est indépendant de la taille du réseau, mais la densité de ce dernier influera sur la vitesse de sa convergence.

3.6 Tableaux comparatifs pour les protocoles de clustering

Pour comparer et analyser les différentes philosophies des protocoles de routage conçus pour les réseaux de capteurs selon l'approche de clustering, il est important d'utiliser des critères de classification appropriés pour pouvoir les distinguer. En effet, la classification permet aux concepteurs de mieux comprendre les caractéristiques de ces protocoles, et de discerner les relations qui les relient.

Ci-dessous deux tableaux inspirés des travaux de [42] et [43] qui vont contenir les différentes analyses et comparaisons apportées aux protocoles détaillés précédemment dans la section 3.5 selon les paramètres de classifications mentionnées dans la section 3.4 (uniquement, les critères de classification estimés importants seront présentés dans les deux tableaux).

Critères de comparaison utilisés dans le premier tableau

Les métriques du tableau 3.2, montrent une comparaison entre les protocoles étudiés précédemment selon les aspects suivants :

-**Les métriques de prise de décision** : Les différents protocoles présentés dans le tableau sont distingués en premier, par les paramètres de prise de décision. D'ailleurs ces paramètres de classification sont les mêmes utilisés dans la section précédente (voir §3.5). Quatre métriques ont été utilisées pour cette classification : le temps de convergence constant et variable, les fonctions utilisées lors du processus de clustering (pour le choix du CH et la formation des clusters), établies selon les heuristiques, poids des noeuds et l'émergence. De ce fait, tous les protocoles qui vont être présentés dans le tableau sont classés selon ces métriques. Il est à noter qu'on va fusionner les protocoles classifiés selon le temps de convergence constant et ceux de temps de convergence variable dans une seule classe : métrique de décision basée sur le temps, tout en indiquant leur type dans le paramètre temps de convergence utilisé dans le tableau, et détaillé plus tard, afin de pouvoir comparer tous les protocoles étudiés selon ce paramètre.

-**Utilisation de puissance** : N/A signifie non applicable ou non disponible. Ce paramètre est employé pour indiquer que le paramètre de puissance n'est pas considéré comme un critère lors du processus de clustering ; il n'est pas traité par ce protocole.

-**Temps de convergence** : Le temps de convergence et l'un des paramètres de prise de décision utilisé dans la classification des protocoles décrits dans 3.5. Ce paramètre dépend de plusieurs facteurs comme la taille du réseau (nombre de noeuds), la méthodologie suivie pour le choix de cluster-heads et/ou la formation de clusters. Deux états sont distinguables: temps de convergence constant et variable (pour plus de détails voir § 3.4.1, §3.5.1, §3.5.2).

-**Mobilité des nœuds** : Un autre paramètre qui semble intéressant à prendre en considération dans la comparaison des protocoles de clustering : la mobilité des nœuds. La plupart des protocoles conçus pour les réseaux de capteurs supposent que les nœuds capteurs sont stationnaires. Cependant, il existe des environnements où les capteurs doivent être mobiles. Dans ce cas, la mobilité n'est plus considérée comme un défi secondaire. Elle est une question clé pour ces environnements mobiles. Pour cela, la propagation des informations sur la position des nœuds dans le réseau devient crucial pour s'adapter au changement de la topologie et donc comparer les protocoles selon cette métrique nous permet de voir l'efficacité de ces protocole à la mobilité des nœuds, ainsi qu'à la conservation d'énergie du réseau.

-**Stabilité des clusters** : La formation et la stabilité des clusters dépendent de plusieurs facteurs qui influent sur leurs comportements. Comme par exemple: l'extinction prématurée de la batterie de certains nœuds, l'ajout d'un nouveau nœud dans le réseau. L'efficacité d'un protocole de clustering se mesure en termes de stabilité des clusters construits, et donc sa capacité de s'adapter aux changements de la topologie sans perturber le fonctionnement du réseau.

-**Informations de localisation** : La localisation des emplacements des nœuds capteurs est un critère utilisé par certains protocoles de clustering afin de router les données d'une manière

efficace, ce qui diminue l'overhead généré par la grande quantité des messages échangés entre les nœuds.

-Équilibrage de charges : Le contrôle d'utilisation de la puissance énergétique ne présente aucune garantie pour que les nœuds soient équitablement utilisés. Cela peut provoquer l'épuisement de l'énergie de certains nœuds avant les autres, et par conséquence, le partitionnement du réseau. La satisfaction de ce paramètre évitera l'épuisement rapide des batteries (voir §3.4.2.2). Cependant, la mise en clusters à taille égale (le même nombre de nœuds dans chaque cluster), devient un objectif qui doit être rempli afin de réaliser la prolongation de la durée de vie du réseau.

Métriques de Décision	Critères de Classification Protocoles	utilisation de puissance	Temps de convergence	Mobilité des nœuds	Stabilité des clusters	Equilibrage de charges	Tolérance aux pannes	Informations de localisation
Basées sur le temps	LEACH	Maximale	Constant O(1)	Non	Moyenne	Oui	Oui	Non
	HEED	Oui	Constant O(1)	Non	Elevée	Bon	N/A	Non
	EECS	Oui	Constant O(1)	Non	Bonne	Très bon	N/A	Oui
	EEUC	Oui	Constant O(1)	Non	Bonne	Très bon	N/A	Oui
	EEHC	Oui	Variable O(n)	Non	N/A	Oui	N/A	Oui
	RCC	N/A	Variable O(n)	Oui	Moyenne	N/A	Qui	Oui
	CLUBS	N/A	Variable O(n)	Oui	Moyenne	N/A	Oui	Non
Heuristique	Hierarchical Control clustering	Oui	Variable O(k1+k2+...+kh)	Possible	N/A	Oui	Oui	Non
	LCA2	Non	Variable O(n)	Possible	Moyenne	Oui	Oui	Oui
	Max-Min-Dmax	Non	Variable O(n)	Possible	Moyenne	Oui	Oui	Oui
Poids	Highest connectivity	Non	Variable O(n)	Possible	Moyenne	Oui	Oui	Oui
	WCA	Oui	variable O(n)	Oui	Modeste	Non	Oui	Oui
	DCA	Non	variable O(n)	Oui	modeste	Non	Oui	Oui
Emergence	ACE	N/A	Constant O(d)	Oui	Elevée	Bon	Oui	N/A

Tab.3.2- Classification des protocoles de clustering présentés selon les métriques de décision, et certains critères qui portent sur les objectifs à atteindre, l'architecture du réseau et modèle de fonctionnement.

D'après le tableau ci-dessous établi depuis l'étude des protocoles exposés dans la section 3.5, les remarques suivantes sont faites :

- Chaque protocole est bon à certains critères de comparaisons. Cependant, il est mauvais par rapport à d'autres. De ce fait, il sera difficile de concevoir un protocole qui supporte tous les critères en même temps.
- L'architecture du réseau influe sur ses performances en termes de puissance utilisée, tolérance aux pannes, mobilité des nœuds, etc. Par exemple : le critère de mobilité, qui ne dépend pas de la structure du réseau mais plutôt de la stratégie suivie par le protocole, et le type de capteurs utilisés dans le réseau.

- Le temps de convergence d'un protocole, dépend des critères de prise de décision utilisés par un nœud. Un nœud dont la décision est indépendante des autres décisions des nœuds, permet d'avoir une complexité temporelle de $O(1)$; temps de convergence constant. Alors que, le temps de convergence des autres protocoles basés sur un paramètre de poids lors de la prise de décision, par exemple, ont typiquement une complexité temporelle élevée : $O(n)$ comme temps nécessaire pour converger.

Critères de comparaison utilisés dans le deuxième tableau

Ce deuxième tableau, comprend toujours une classification des protocoles de clustering selon les métriques de prise de décision. En outre, il introduit des nouvelles comparaisons entre ces protocoles en termes d'attributs de clustering: les propriétés de clusters générés, les capacités de cluster-heads souhaitées et le processus de clustering adopté. Dans le reste de cette section, nous discutons tous ces critères de comparaison.

- **Propriétés des clusters :** Les différents protocoles de clustering essayent de répondre aux questions suivantes :
 1. Le nombre optimal de clusters dans un réseau de capteurs. Souvent, le nombre de clusters dans le réseau peut être préréglé (déterminé a priori par le concepteur du réseau) ou bien il est variable.
 2. Le type de connectivité entre les CHs et les différents clusters. Soit le lien de connexion est direct ou via plusieurs sauts.
 3. Les clusters sont stables (fixes), si l'appartenance d'un nœud à un groupe ne varie pas tous le temps. Et donc on aura une topologie fixe et stationnaire. Sinon, ils sont adaptatifs : si le nombre de ces groupes n'est pas fixe et les capteurs ne commutent pas entre les clusters. La topologie du réseau dans ce cas est adaptative.
- **Capacités d'un cluster-head :** Le modèle du réseau influe sur la capacité des cluster-heads en termes : de mobilité des nœuds (les nœuds du réseau sont stationnaires et fixes, ou peuvent se déplaçait sur des courtes distances), ainsi que sur le type des capteurs utilisés (homogènes/ hétérogènes). De plus, le CH choisi peut jouer plusieurs rôles: un nœud relais entre la station de base et les nœuds membres de son cluster, et parfois un relais pour les CHs des autres clusters.
- **Processus de clustering:** Les caractéristiques du processus de clustering varient selon la méthodologie adoptée, les objectifs à satisfaire, le choix à réaliser pour l'élection du CH, et finalement la complexité obtenue (en termes de temps/messages échangés). Généralement le processus de clustering agit selon deux méthodologies. Soit de façon centralisée, sinon de manière distribuée afin de parvenir à la satisfaction des différents objectifs (voir § 3.4.) dont la minimisation de la consommation d'énergie est le but le plus requis.

Métrique de Décisions	Critères de Classification Protocoles	Propriétés des clusters				Capacités du cluster-head			Processus de clustering			
		Nombre de clusters	Topologie intra-clusters	Connectivité inter-clusters	Stabilité	Mobilité	Type de nœuds	Rôle	Méthodologie	Objectifs du clustering	Sélection du CH	Complexité d'algorithmes
Basée sur le temps	LEACH	variable	Fixé (1- saut)	Lien direct (1-saut)	Moyenne	Stationnaire	Capteurs	Relais/Agrégation	Distribuée	Conserver l'énergie	Prédéterminée	Constante
	HEED	variable	Fixé (1- saut)	Lien direct/ multi sauts	Assurer	Stationnaire	Capteurs	Relais/Agrégation	Distribuée	Conserver l'énergie	Aléatoire	Constante
	EECS	variable	Adaptative	multi -sauts	Assurer	Stationnaire	Capteurs	Relais/Agrégation	Distribuée	Conserver l'énergie	Concurrente	Constante
	EEUC	variable	Adaptative	multi -sauts	Assurer	Stationnaire	Capteurs	Relais	Distribuée	Conserver l'énergie/équilibrage de charge	Aléatoire	Constante
	EEHC	variable	Adaptative	Lien direct/ multi sauts		Stationnaire	Capteurs	Relais/Agrégation	Distribuée	Conserver l'énergie	Aléatoire	Variable
	RCC	variable	Adaptative	Lien direct	Moyenne	Mobile	Capteurs	Relais	hybride	Scalabilité	Aléatoire	Variable
	CLUBS	variable	Fixé (2- sauts)	multi -sauts	Moyenne	Stationnaire	Capteurs	Relais	Distribuée	Scalabilité	Aléatoire	Variable
	Hierarchical Control clustering	variable	Adaptative	multi -sauts		Possible	Capteurs	Relais/Agrégation	Distribuée	Scalabilité	Aléatoire	Variable
Heuristique	LCA2	variable	Fixé (1- saut)	Lien direct	Moyenne	Possible	Capteurs	Relais/Agrégation	Distribuée	connectivité	Aléatoire	Variable
	Max-Min-Dmax	variable	Adaptative	Lien direct	Moyenne	Possible	Capteurs	Relais/Agrégation	Distribuée	connectivité	Aléatoire	Variable
	Highest connectivity	variable	Adaptative	Lien direct	Moyenne	Possible	Capteurs	Relais/Agrégation	Distribuée	connectivité	Aléatoire	Variable
Poids	WCA	variable	Adaptative	Lien direct	Modeste	Mobile/stationnaire	Capteurs	Relais/Agrégation	Distribuée	Conserver l'énergie	Aléatoire	Variable
	DCA	variable	Adaptative	Lien direct	modeste	Possible	Capteurs	Relais/Agrégation	Distribuée		Aléatoire	Variable
Emergent	ACE	variable	Adaptative	Lien direct	Assurer	stationnaire	Capteurs	Relais/Agrégation	Distribuée	Conserver l'énergie/scalabilité	Aléatoire	Variable

Tab.3.3 - Classification des protocoles de routage présentés selon les paramètres de décision et les attributs du clustering.

En analysant le tableau 3.3, selon les critères de classification basés sur les paramètres de prise de décision et les attributs de clustering, les observations suivantes sont tirées :

-Les protocoles à temps de convergence constant et variable, commandent le nombre de clusters en plaçant la probabilité de devenir un CH selon les capacités des nœuds et les conditions d'application. Alors que, le nombre de clusters résultant des protocoles émergents dépend du nombre de propriétés existantes dans le réseau, tant dit que dans les protocoles pondérés c'est les différents paramètres intervenants dans la fonction de calcul de poids qui déterminent le nombre de clusters dans le réseau.

-Peu de protocoles sont conçus pour gérer la mobilité dans les réseaux de capteurs. Comme la majorité des nœuds capteurs sont stationnaires, les protocoles existants sont moins adaptés à des environnements mobiles. Donc l'application pour laquelle un réseau de capteur va servir déterminera les caractéristiques de ce dernier et le type de capteurs a déployé. Et donc, le type de l'application va imposer l'environnement de déploiement des capteurs et caractérisera leurs natures ainsi que la méthodologie suivie pendant le processus de clustering.

-La stabilité des clusters dépend de la/les métrique(s) utilisés dans l'élection des CHs (métriques de décisions) et donc pour construire des clusters plus ou moins stables, il est nécessaire de bien choisir ces métriques. En outre, ces métriques d'élection doivent tenir compte de la capacité des nœuds qui vont jouer le rôle de CH, ainsi que la topologie du réseau.

-Le type de communication (le mode d'envoi de données) adopté dans les clusters ou inter-clusters détermine le rôle jouer par le CH: collecte de données des nœuds membres et envoie direct à la station de base ou bien passer par d'autres CHs pour joindre la station de base (multi-saut). Ces CHs peuvent être autorisés ou non à appliquer des techniques d'agrégation/fusion sur les données collectées. De ce fait, la quantité des messages échangés dans le réseau dépendra du type de communication choisi.

-Selon le tableau, on peut remarquer que le processus de clustering peut avoir de multiples objectifs à saisir dont la conservation d'énergie est le plus important. De plus, on retrouve l'équilibrage de charge et la scalabilité comme deux autres objectifs qu'un protocole de clustering essaye de les remplir. Par exemple, la scalabilité du réseau dépend en premier lieu de la méthodologie suivie pour la formation de clusters ainsi que par les techniques de choix des CHs.

-La majorité des protocoles adoptent un mécanisme décentralisé pour la formation des clusters et chaque CH se charge individuellement de la formation de son propre cluster.

3.7 Conclusion

Ce chapitre a été axé sur l'approche de clustering dans les réseaux de capteurs sans fil. Ainsi, une définition de cette dernière et une présentation de quelques protocoles conçus selon cette approche ont été réalisées. Plusieurs méthodologies ont été employées pour offrir une structuration de clusters pour ces réseaux. Les principales techniques utilisées par les protocoles de clustering publiés dans la littérature ont été énumérées. Une classification a été établie pour définir les caractéristiques essentielles des protocoles étudiés. Par la suite, une étude critique a été menée sur les différents protocoles présentés dans ce chapitre, afin qu'on puisse tirer profit de leurs avantages en conservation énergétique.

Bien que la technique de routage hiérarchique semble prometteuse, ils restent encore de nombreux défis à relever par de nouvelles approches de routage efficaces en énergie. L'approche de routage que nous avons proposé est nommée : LMEER(*Layered Multi-hop Energy Efficient clustering Routing protocol*). Ce nouveau protocole va constituer l'objet principal du prochain chapitre. Son fonctionnement ainsi que ses caractéristiques sont exposées en détail dans le chapitre suivant.

Chapitre 4

LMEER : Un protocole de clustering à basse consommation d'énergie pour les réseaux de capteurs sans fil

4.1 Introduction

La majorité des travaux de recherche menés actuellement dans le domaine des réseaux de capteurs se concentrent sur le problème de conservation d'énergie afin de prévoir des algorithmes et protocoles spécifiques à ce genre de réseaux consommant le minimum d'énergie. Ainsi, le traitement de ce défi dépend des contraintes imposées par les capteurs : petite capacité de stockage, basse capacité de traitement, durée de vie limitée de la batterie et portée radio réduite. Le développement d'une technique efficace permettant d'économiser la ressource énergétique est un objectif primordial pour ce type de réseaux. L'architecture basée sur le clustering est considérée comme une approche prometteuse pour cette finalité. Le clustering consiste à partitionner virtuellement le réseau en un ensemble de clusters mais aussi sélectionner des nœuds spéciaux nommés « CHs », il sont chargés d'organiser la communication dans les clusters ainsi que dans le réseau.

Bien que plusieurs travaux dans le domaine des réseaux de capteurs ont été publiés sur le routage selon l'approche de clustering, plusieurs défis en termes de gestion de la consommation d'énergie des nœuds capteurs restent présents. Pour cela, il est intéressant de concevoir de nouveaux protocoles et techniques de clustering plus rigoureux pour la gestion de la ressource énergétique et l'optimisation de sa consommation, afin de permettre à ces réseaux d'accomplir au mieux leurs tâches sans que leurs performances soient dégradées lors du passage à l'échelle ou la présence de pannes. Ceci nous a mené à proposer un nouveau protocole de routage selon l'approche de clustering essayant d'offrir de meilleures performances pour le réseau en particulier en termes de conservation d'énergie.

Ce chapitre est structuré comme suit : dans la première partie nous présentons notre nouveau protocole de routage proposé selon l'approche de clustering. Ce protocole est nommé LMEER : *Layered Multi-hop Energy Efficient clustering Routing protocol*. Une description détaillée de ce nouveau protocole est donnée ainsi qu'à travers les différentes phases qui le constituent. La seconde partie de ce chapitre est consacrée à l'évaluation des performances du protocole LMEER. La simulation est utilisée comme outil d'évaluation de ses performances. Les résultats obtenus et leurs interprétations ainsi que les paramètres utilisés dans la simulation de ce nouveau protocole seront données dans la section 4.4. Et pour illustrer les atouts de LMEER, nous comparons les différents résultats obtenus à ceux obtenus par le protocole LEACH (voir § 3.5.1.1). La conclusion de ce chapitre est présentée dans la section 4.6.

4.2 Motivations

Le routage est un service très important dans les réseaux de capteurs. Il doit permettre l'arrivée des données à la station de base avec le minimum de perte et de dissipation d'énergie. Les approches de routage utilisant un seul saut pour communiquer les paquets de données

consomment beaucoup d'énergie pour atteindre la station de base si les noeuds chargés de lui router les paquets sont très éloignés. Pour remédier à cette lacune, nous avons utilisé le routage à plusieurs sauts dans notre protocole. LMEER est donc adapté au routage multi-sauts réalisé par les CHs des différents clusters. C'est également dans ce but qu'une structuration en couches est appliquée au réseau afin de faciliter le choix des nœuds assurant le routage multi-sauts. De plus, cette topologie en couches est exploitée dans la politique de formation de clusters car LMEER exécute l'algorithme de formation de clusters d'une façon adaptative selon les caractéristiques de la couche d'un noeud donné (sa distance par rapport à la station de base), ce qui lui permet une certaine autonomie dans la gestion des capteurs situés dans chaque couche.

L'adoption d'un mécanisme d'équilibrage de la charge permet une optimisation de la durée de vie de chaque nœud et par conséquent, la durée de vie de l'ensemble du réseau. Et pour équilibrer la consommation d'énergie entre les nœuds choisis CHs, il est primordial d'équilibrer leur charge de travail ; nombre de membres à gérer dans leurs clusters et paquets de données à router. En d'autres termes, il est recommandé d'éviter toute situation où certains CHs sont surchargés alors que d'autres le sont légèrement ou sont complètement libres. C'est sur ce principe que nous nous sommes basés en adoptant le degré d'un nœud comme métrique dans le poids d'élection de son CH à joindre, avec également l'utilisation d'informations sur les énergies résiduelles. Ceci permet à LMEER de produire un partitionnement de taille inégale pour grouper les nœuds lors du processus de clustering et donc équilibrer la charge de capteurs à gérer et la consommation d'énergie au niveau des noeuds CHs. Ce partitionnement du réseau en clusters de tailles inégales permet aussi de résoudre le problème des points chauds (Voir § 3.5.1.4). Ce problème survient dans les couches hautes: les CHs qui entourent la station de base épuisent plus rapidement leur énergie que les autres CHs des couches supérieures, car tout le trafic du réseau destiné à la station de base passe par ces CHs et donc ces noeuds ont plus de charge que les nœuds distants. La construction des clusters les plus proches de la station de base avec un petit nombre de membres à gérer et ceux qui sont plus loin avec un plus grand nombre de nœuds membres dans leurs clusters, permet de garantir la stabilité de la topologie et d'améliorer les performances du réseau en offrant une possibilité de conservation d'énergie pour les nœuds sélectionnés CHs.

4.3 Principe et fonctionnement

L'énergie reste le défi critique dans les réseaux de capteurs car ils sont dotés de batteries non rechargeables ni remplaçables. Ainsi, l'objectif principal des protocoles dédiés aux réseaux de capteurs est de minimiser la consommation d'énergie tout en assurant le bon fonctionnement de ces réseaux. LMEER est proposé comme un nouveau protocole de clustering suivant cet objectif. Ce protocole adopte un mécanisme permettant de configurer le réseau en couches afin d'assurer un routage multi-sauts entre ces différentes couches. Entre autres, un mécanisme pour le groupement des nœuds en clusters est aussi utilisé. Ce mécanisme essaye d'assurer une distribution de la charge de travail et donc garantir une exploitation plus équitable de l'énergie des nœuds.

Le fonctionnement de LMEER est complètement décentralisé. Il s'exécute sur trois phases. La première porte sur la configuration du réseau. Alors que la deuxième assure la formation des clusters et la communication de données. Finalement, la maintenance de la topologie est considérée comme sa troisième phase. Le détail de chaque étape est donné dans le reste de cette section.

4.3.1 Configuration du réseau

Cette phase permet la structuration du réseau en couches ainsi que la découverte des voisins et donc permettre à chaque nœud de construire ses listes de voisins.

La découverte des voisins est lancée par la station de base tout en utilisant la diffusion. Un nœud est considéré comme un voisin d'un autre nœud si ce dernier se trouve dans sa portée de transmission. A la fin de cette phase, tous les nœuds du réseau sont partitionnés en couches et possédants leur propre tables de routage. Et pour configurer le réseau en couches deux mécanismes sont possibles: (i) en se basant sur le nombre de sauts qui sépare les nœuds à la station de base, ou bien (ii) en se basant sur leur distance. Nous les avons testé tous les deux.

i. En se basant sur le nombre de sauts

Le mécanisme et l'algorithme relatifs à cette configuration du réseau sont détaillés ci-dessous.

a). Mécanisme

Initialement la station de base diffuse un message d'exploration Hello pour découvrir ses voisins à un seul saut tout en utilisant dans ce paquet une variable nommée : num_couche mise à zéro. Seulement les nœuds à la portée de la station de base peuvent recevoir ce message diffusé. Par la suite, ces noeuds vont agir en tant que station de base locale pour les autres nœuds voisins. Ils explorent le réseau en diffusant le message Hello aux nœuds situés dans leurs portée tout en remplaçant leurs informations dans le message.

Pour chaque message reçu, le nœud récepteur exécute les étapes suivantes :

- Si le nœud émetteur n'existe pas dans sa table de voisinage, il ajoute une entrée dans sa table de voisinage comportant les informations de ce noeud émetteur. Si ce dernier figure déjà dans la table de voisinage il effectue une simple mise à jour des informations correspondantes.
- Un nœud peut recevoir plusieurs messages de diffusion avec un nombre de sauts différent. Le message contenant un nombre de sauts minimum (le moins distant de la station de base) est toujours favorisé. Pour cela, le numéro de couche correspondant au nombre de sauts assigné au nœud récepteur est toujours calculé de la façon suivante :

*Si $mon.num_couche > voisin.num_couche + 1$ Alors
 $mon.num_couche \leftarrow voisin.num_couche + 1$; (mon est le noeud en cours)*

- Un numéro de séquence est utilisé dans le message d'exploration du réseau. Ce numéro permet d'éviter la rediffusion d'un même message d'exploration Hello plus d'une fois. En gardant localement le numéro de séquence du dernier message d'exploration reçu au niveau de chaque nœud, le message ne va être rediffuser que si le numéro de séquence reçu est différent de celui stocké. Et donc, plusieurs couches sont construites selon le nombre de sauts qui sépare la station de base aux différents nœuds capteurs. Cette méthode permet de minimiser le nombre de messages de contrôle échangés pour structurer la topologie.

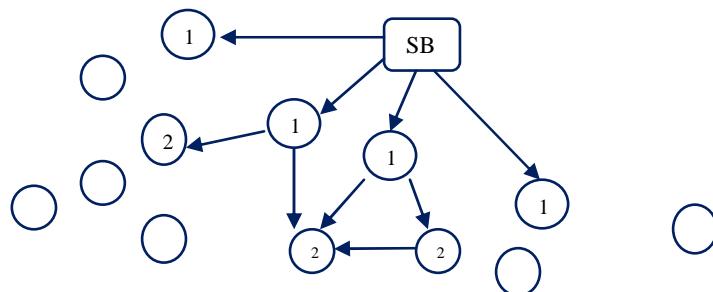


Fig. 4.1 Configuration du réseau en se basant sur le nombre de sauts.

A la fin de cette phase, chaque nœud connaît sa propre distance jusqu'à la station de base en termes de nombre de sauts. De plus, il a une table de voisinage avec une entrée pour chaque voisin contenant : son identifiant ainsi que son numéro de couche reflétant sa distance à la station de base en fonction du nombre de sauts qui les sépare.

b). Algorithme

La configuration du réseau selon le nombre de sauts peut être formulée de la façon suivante :

Soit les nœuds capteurs i et j .
 Soit V_i la table des voisins du nœud i .
 Soit seq_i , seq_j le numéro de séquence du dernier message Hello au niveau du nœud i et nœud j resp.

Lors de la réception d'un message Hello par i depuis le nœud j

- Si* ($j \notin V_i$) *Alors*
 - Ajouter une entrée pour j dans V_i
- Sinon*
 - Mettre à jour les informations de j ;
- Fin Si*

Si ($seq_j > seq_i$) *Alors*

- Rediffuser le paquet Hello avec les informations de i ;
- Mettre à jour la valeur de seq_i par celle de seq_j ;

Sinon

- Rejeter le paquet reçu ;

Fin Si

Fin lors

ii. En se basant sur la distance

Cette étape de configuration du réseau en fonction de la distance est détaillée dans le paragraphe suivant.

a). Mécanisme

Avec cette méthode, des messages d'exploration sont diffusés par la station de base dans tous le réseau. A la réception d'un message d'exploration, chaque nœud calcule sa distance par rapport à la station de base selon la puissance du signal reçu. Cette distance est utilisée par la suite dans l'attribution d'un numéro de couche pour chaque nœud. En comparant sa distance à la

station de base avec une autre distance D donnée par : $D = K R_i$, le nœud s'assigne un numéro de couche de la façon suivante:

```

Tant que  $d(i, SB) > D$  faire
     $k \leftarrow k + 1;$ 
fin Tant que
     $mon.num\_couche \leftarrow mon.num\_couche + k;$ 

```

Où :

$D = k R_i$: R_i représente la portée du nœud i qui est identique pour tous les nœuds du réseau et $k = 0..n$.

$d(i, SB)$: Distance euclidienne entre le nœud i et la station de base.

$mon.num_couche$: variable utilisée pour stocker le numéro de couche pour chaque nœud, initialisée à un.

Par exemple, un nœud est dit de couche 1 si la station de base se trouve dans sa portée.

Une fois que chaque nœud s'est affecté un numéro de couche, il diffuse un message de notification contenant cette information et son identifiant à ses voisins.

A la réception de ce message de notification, un nœud intermédiaire ajoute le nœud émetteur à sa table de voisinage avec son numéro de couche associé. Si ce dernier figure déjà dans la table de voisinage, il ne fait qu'une mise à jour à son numéro.

A la fin de cette phase chaque nœud possède une vue locale sur son voisinage. Les identités et les numéros de couches de tous ses voisins sont sauvegardés dans sa table de voisinage.

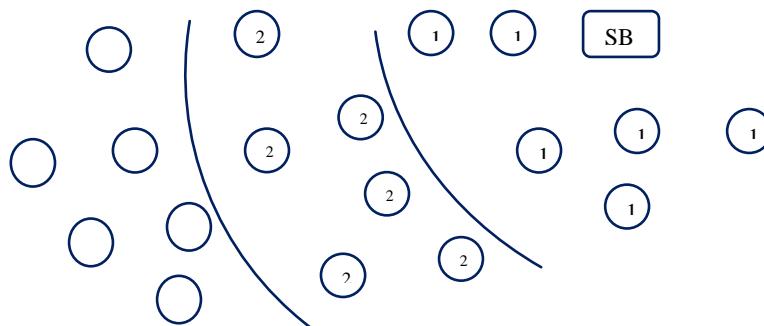


Fig. 4.2 Configuration du réseau En se basant sur la distance

b). Algorithme

L'algorithme qui résume la phase de configuration du réseau en fonction de la distance est donné comme suit :

Soit les nœuds i , j et SB la station de base ; Soit V_j la table des voisins du nœud i ; Soit num_couche : variable pour stocker le numéro de couche assigné à chaque nœud ; Soit R_i : la portée du nœud i ; Soit K un compteur ; $k \leftarrow 0;$ Tant que (la distance entre (i , SB) $> K * R_i$) Faire {incrémenter la valeur du compteur k } $k \leftarrow k + 1;$ fin tant que ; {mise à jour sur la valeur du num_couche } $num_couche \leftarrow num_couche + k;$ {le nœud i informe ses voisins en diffusant un paquet Hello contenant ses informations} Diffuser le paquet Hello;
--

Lors de la réception d'un paquet Hello par j depuis le nœud i

Si (i \notin Vj) *Alors*

Ajouter une entrée pour i dans Vj

Sinon

Mettre à jour les informations de i ;

Fin Si

Fin lors

4.3.2 Formation des clusters et communication de données

Cette phase permet la formation des clusters et la transmission des données à la station de base. Elle se caractérise par un fonctionnement cyclique où chaque cycle comprend trois sous phases : élection de cluster-heads (CHs), formation de clusters et communication de données.

1. Election des cluster-heads

La question la plus pertinente qu'un processus de clustering essaye de la satisfaire est de trouver le nombre optimal de nœuds à choisir comme CHs dans le réseau. Si le nombre de CHs est très élevé, donc un nombre important de nœuds (CHs) qui se consacrent aux tâches très coûteuses en ressources énergétiques, la dissipation d'énergie dans le réseau est considérable. D'un autre côté, si le nombre de CHs est très petit, ces derniers vont gérer des groupes de grande taille et donc épuiseront rapidement leur énergie à cause du travail important qui leur est demandé.

De ce fait, il faut trouver un compromis entre le nombre total de nœuds dans le réseau et le nombre de noeuds choisis comme CHs. Pour ce faire, notre protocole inclut le numéro de couche dans la fonction de calcul de poids assigné à chaque nœud et de sorte que la taille des clusters est réduite en s'approchant de la station de base.

a) . Mécanisme

Cette phase est amorcée par l'annonce d'un nouveau cycle (round) en exécutant au niveau de chaque capteur les étapes suivantes :

- Un nœud s'auto-élise pour rôle de CH en évaluant une fonction poids Pi. La fonction de calcul de poids utilisée doit permettre de choisir les nœuds ayant la plus grande capacité énergétique, possédant le grand nombre de voisins et aussi privilégier les nœuds jouant le rôle de CH pour un nombre de fois limité. Le degré d'implication de chaque paramètre varie en fonction du numéro de couche de chaque nœud. Pour déterminer la fonction de calcul de poids on combine :

1. *Le niveau d'énergie résiduelle du nœud* : cette métrique est inversement proportionnelle à la distance vers la station de base. Son degré d'implication pour les couches hautes (proches de la station de base) est plus grand par rapport aux autres paramètres. Un nœud disposant de plus d'énergie résiduelle aura une forte probabilité d'être choisi CH.

2. *Le degré* : présente le nombre de voisins pour un nœud donné. Ce paramètre permet la sélection du nœud disposant d'un plus grand nombre de liens de communication avec ses nœuds voisins. Egalement, il offre la possibilité d'effectuer un contrôle sur la distribution des CHs dans le réseau: plus on s'éloigne de la station de base plus est important ce paramètre. Ceci permet d'assurer une distribution uniforme des clusters dans le réseau afin d'équilibrer la charge des CHs choisis.
3. *Le nombre de fois qu'un nœud a été déjà sélectionné CH* : cette métrique permet de garantir une rotation du rôle de CH et donc offrir plus de robustesse pour le fonctionnement.

Le poids P_i d'un nœud i est donnée par :

$$P_i = \left(\frac{1}{\alpha - num_{couche}(i)} * \frac{deg_i}{N} \right) + \left(\frac{1}{\beta + num_{couche}(i)} * \frac{E_{res}}{E_{int}} \right) + \left(\gamma * (1 - \frac{d(i, SB)}{d_{max}}) \right) - \left(\mu * 1 - \frac{1}{1 + nbr_{CH}} \right)$$

Tel que :

$$\frac{1}{\alpha - num_{couche}(i)} + \frac{1}{\beta + num_{couche}(i)} + \gamma + \mu = 1 \quad \text{et} \quad \gamma = 0, \quad \alpha, \beta, \gamma, \mu \quad \text{paramètres fixés par le système.}$$

Où :

$num_{couche}(i)$: numéro de couche du nœud i , N : nombre totale de nœuds dans le système.

deg_i : le nombre de voisins du nœud i . nbr_{CH} : nombre de fois déjà choisi comme nœud CH.

E_{res} : correspond à l'énergie résiduelle estimée dans les nœud i , E_{int} : Énergie initiale maximum de référence qui correspond à une batterie entièrement chargée qui est identique pour tous les nœud.

$d(i, SB)$: distance entre la station de base et le nœud i . d_{max} : diamètre du réseau.

-Si la valeur du poids P_i est inférieure à un seuil T_{ci} , ce nœud se déclare CH pour le cycle courant. Les valeurs du seuil T_{ci} sont inversement proportionnels au numéro de la couche c du nœud i afin de mieux s'adapter aux différentes couches et donc d'avoir un nombre de CHs qui augmente en se rapprochant à la station de base.

Il est à noter, qu'au niveau des nœuds de la première couche situés à la portée de la station de base, la valeur de γ est différente de zéro. Ceci permet d'introduire la distance par rapport à la station de base comme paramètre dans la fonction de calcul de poids afin de favoriser les nœuds les moins distant à cette dernière et donc consomme un minimum d'énergie pour l'atteindre.

b).Algorithme

Soit i un noeud dont le numéro de couche égale à C_i ;

Soit T_{ci} seuil de la couche C_i ;

Soit $isCH$: variable booléenne utilisée pour indiquer le statut du nœud ;

Soit P_i : poids du noeud i ;

```

Fonction décider _CH : { pour un nœud i }

  Si ( Ci = 1 ) Alors
     $P_i \leftarrow \left( \frac{1}{\alpha - num_{couche(i)}} * \frac{deg_i}{N} \right) + \left( \frac{1}{\beta + num_{couche(i)}} * \frac{E_{res}}{E_{int}} \right) + \left( \gamma * \left( 1 - \frac{d(i, SB)}{d_{max}} \right) \right) - \left( \mu * 1 - \frac{1}{1 + nbr_{CH}} \right);$ 
  Sinon
     $P_i \leftarrow \left( \frac{1}{\alpha - num_{couche(i)}} * \frac{deg_i}{N} \right) + \left( \frac{1}{\beta + num_{couche(i)}} * \frac{E_{res}}{E_{tot}} \right) - \left( \mu * 1 - \frac{1}{1 + nbr_{CH}} \right);$ 
  Fin si

  Si ( Pi < Tci ) Alors
    isCH  $\leftarrow$  vraie
  sinon
    isCH  $\leftarrow$  faux // nœud ordinaire ;
  Fin si

Fin

```

2. Formation de clusters

Une fois qu'un nœud s'auto-élu CH, il doit informer les autres nœuds de son nouveau rôle dans le cycle courant. Pour cela, un message d'annonce contenant l'identifiant et le poids du CH est diffusé à tous les nœuds de son voisinage en utilisant le protocole CSMA de la couche MAC. La diffusion permet de s'assurer que tous les nœuds voisins ordinaires ont reçu le message d'annonce. Par ailleurs, elle garantie aussi que les autre nœuds CHs adjacents reçoivent à leur tour ce message.

Lors de la réception des messages d'annonces, les nœuds ordinaires construisent des listes pour maintenir ces CHs. La décision d'appartenance d'un nœud à un CH est prise suite à une comparaison de leurs poids. Le CH doté du plus grand poids prévaut. Le nœud ordinaire lui envoie par la suite, un message de rattachement pour l'informer de sa décision d'appartenance à son cluster.

Cette décision contrôlée permet d'équilibrer la charge des CHs en produisant des clusters ayant des tailles différentes. La taille d'un cluster diminue en fonction du numéro de la couche de son CH: les CHs plus loin, ont des clusters de plus grand éventail que ceux plus proches de la station de base. L'objectif est d'assurer un nombre de nœuds membres plus petit et une taille plus petite à gérer pour les CHs de couches hautes proches de la station de base.

D'autre part, au niveau de chaque CH une liste de ses CHs adjacents est maintenue. Les éléments de cette liste sont utilisés dans la sélection du CH relais : prochain saut utilisé dans la phase de communication de données inter-clusters.

a). Mécanisme

Après l'auto-élection des CHs dans la phase précédente, chaque CH diffuse un message dans son voisinage annonçant sa décision pour rôle de CH dans le cycle courant. Ce message contient son identifiant ainsi qu'un poids (PCH) définit de telle sorte à prendre en considération l'énergie résiduelle au niveau de ce CH ainsi que son degré et son numéro de couche. Ce poids est donné par la formule :

$$PCH = \frac{E_{res}}{deg_i} * num_{couche(i)}$$

Lors de la réception d'un message d'annonce par un nœud ordinaire, ce nœud ajoute le CH à sa liste de choix CHs, puis procède à une comparaison de leur poids tout en choisissant le max (le CH possédant une plus grande puissance énergétique et ayant le minimum de capteurs à gérer). En cas d'égalité le CH le plus distant à la station de base est privilégié afin de ne pas surcharger les autres CHs qui sont proches de cette dernière.

Il est important de noter que les nœuds à la portée de la station de base (constituant la première couche de l'hiérarchie du réseau) prennent leurs décisions de rattachement à un CH en fonction de leur distance par rapport à ce CH et à la station de base. Ceci se justifie par le fait que le coût de communication d'un nœud avec un CH moins distant est plus petit que le coût de celui à un CH plus éloigné. De ce fait, les décisions de rattachement sont prises selon le poids PCH qui devient :

$$PCH = \left(\alpha \frac{d(CH, i)}{d_{max}} \right) + \left((1 - \alpha) \frac{d(i, SB)}{d_{max}} \right)$$

Où

α : paramètre fixé par le système prenant ses valeurs dans $[0, 1]$.

$d(CH, i)$: la distance qui sépare un nœud i d'un nœud CH.

$d(i, SB)$: la distance entre la station de base et le nœud i .

d_{max} : diamètre du réseau.

En cas d'égalité, les nœuds choisissent le CH possédant le plus grand poids: ayant la plus grande quantité énergétique et le moins chargé comme déjà détaillé précédemment.

Une fois les nœuds CHs sont choisis, chaque nœud membre envoie un message informant son CH sélectionné de sa décision de rattachement à son cluster.

Les messages d'annonces diffusés par les CHs sont exploités aussi dans la construction des listes de CHs adjacents au niveau des nœuds CHs, dans le cas où le message diffusé par un CH est reçu par un autre nœud CH voisin. Ceci est dans le but de choisir le meilleur nœud relais comme prochain saut requérant le minimum d'énergie pour la communication et ayant la plus petite charge afin d'équilibrer la consommation d'énergie lors de la communication multi-sauts de données entre les CHs des différentes couches et la station de base.

Le CH de couche i (couche supérieure) le moins distant ainsi que celui possédant le plus grand poids (PCH) est élu comme prochain nœud relais par le CH de couche $i+1$. Son identité et sa position sont ajoutées dans la table de routage comme étant le prochain saut.

Si la liste de choix de CHs adjacents de couche supérieure est vide, un des voisins non membre appartenant à la couche i est sélectionné d'une manière aléatoire comme prochain nœud relais par le CH de couche $i+1$.

Autrement, si aucune de ces deux possibilités n'est satisfaite, dans ce cas le CH choisi comme prochain saut le CH adjacent le moins distant possédant le plus grand poids et ayant le même numéro de couche que lui.

Après avoir effectué le choix, chaque CH informe le nœud relais sélectionné de son nouveau rôle dans le cycle courant. De plus, à travers ce mécanisme de routage multi-sauts introduit par les CHs moins d'énergie est consommée pour atteindre la station de base ce qui offre de meilleures performances pour le réseau.

b).Algorithme

```

Soit les nœuds i, j dont Ci, Cj leurs numéros de couche respectifs ;
Soit PCH : poids du nœud i ;
Soit les fonctions :
    Trouver_meilleur_CH ();{retourne l'identifiant du nœud CH ayant une valeur de métrique maximale } ;
    Envoyer_Joig_msg ();{utilisée pour envoyer un paquet de jointure au CH choisi par Trouver_meilleur_CH } ;
    Trouver_interCH (); { retourne l'identifiant du nœud relais ayant une valeur de métrique maximale } ;
    Envoyer_informe_interCH ();{utilisée pour envoyer un paquet au nœud choisi comme relais par Trouver_interCH} ;
Soit les listes :
    meilleur_CH ;{ détient les CHs entendus } ;
    sup_inter_CH ;{ stocke les CHs de couches supérieures } ;
    same_inter_CH ;{ pour maintenir les CHs ayant le même numéro de couche } ;
Soit isCH : variable utilisée pour indiquer le statut du nœud ;{ faux pour nœud ordinaire, vrai pour nœud CH }
```

Fonction Envoyer_ADV_CH :{envoie un paquet ADV_CH par un nœud i déclarant son nouveau statut de CH }

{ calculer le poids PCH }

$$PCH \leftarrow \frac{E_{res}}{\deg_i} * \text{num}_{\text{couche}(i)}$$

{créer le paquet ADV_CH contenant son identifiant i, son poids et son numéro de couche puis l'envoyer à ses nœuds voisins}

Diffuser ADV_CH () ;

Fin

Fonction Recevoir_ADV_CH :{ lors de la réception d'un message ADV_CH par le nœud j depuis le nœud i }

Si (isCH = faux) Alors

 Insérer les informations du nœud j dans la liste meilleur_CH ;

 Trouver_meilleur_CH () ;

 Envoyer_Joig_msg () ;{ paquet envoyé au CH choisi }

Sinon

 Si (Ci < Cj) Alors

 Insérer les informations de i dans la liste sup_inter_CH ;

 Sinon

 Si (Ci = Cj) Alors

 Insérer les informations de i dans la liste same_inter_CH ;

 Fin si

 Fin si

 Trouver_interCH () ;

 Envoyer_informe_interCH () ;{ paquet envoyé au nœud relais choisi }

Fin si

Fin

3. Communication de données

Cette phase assure la collecte des données captées et leur transmission à la station de base. Elle est plus longue que les deux sous phases précédentes.

En utilisant l'ordonnanceur TDMA (Time Division Multiple Access), les membres émettent leurs données captées pendant des slots de temps affectés par leur CH. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leur énergie. Ces données sont ensuite agrégées par le CH puis envoyées au prochain nœud relais. Les données transitent de nœud relais en nœud relais jusqu'à ce qu'elles arrivent à la station de base.

Pour que plusieurs nœuds lors de la transmission intra-cluster et inter-clusters peuvent envoyer simultanément leurs données, le protocole d'accès multiple par répartition en code CDMA (Code Division Multiple Access) est utilisé. Ce protocole ne divise ni la plage de fréquences ni l'intervalle de temps. Ainsi, des nœuds peuvent émettre leur données continuellement et selon une large plage de fréquence via des techniques d'étalement de spectre (DSSS) afin d'éviter les collisions entre les différentes transmissions simultanées des nœuds.

a) . Mécanisme

Les détails de fonctionnement de cette phase de communication de données sont présentés ci-dessous.

Etablissement de l'ordonnancement et communication de données intra-cluster

Une fois les clusters sont formés, chaque CH agit comme étant un centre de commande local pour coordonner les transmissions des données au sein de son cluster. Il crée un ordonnanceur (*schedule*) TDMA et assigne à chaque nœud membre un slot de temps durant lequel il peut émettre ses données. L'ensemble des slots assignés aux nœuds d'un cluster est appelé frame. La durée de chaque frame diffère selon le nombre de membres du cluster. Par ailleurs, afin de minimiser les interférences entre les transmissions des clusters adjacents, chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA. Il le communique par la suite à ses nœuds membres afin de l'utiliser dans leur transmission.

Les données reçues à partir d'un noeud CH de couche $i+1$ par un nœud membre de couche i considéré comme son nœud relais sont agrégées avec ses propres données puis envoyées à son CH durant son intervalle de temps affecté.

Communication de données inter-clusters

Afin d'accroître les performances de LMEER, le multi-saut est adopté comme un mécanisme de routage pour transmettre les données collectées à la station de base. Cette transmission est faite de relais en relais jusqu'à ce que le paquet de données arrive à la station de base. De cette façon, moins d'énergie est consommée pour acheminer les différents paquets de données dans le réseau. De plus, la quantité de messages de communication générée dans le réseau est moins élevée, car les noeuds relais sont dotés d'une fonction d'agrégation réalisée sur les données issues des autres nœuds relais en combinant ces données avec leurs propres données. La fonction d'agrégation de données utilisée n'applique aucune opération de transformation sur la valeur des données ce qui permet de préserver leurs valeurs tout en utilisant un seul message qui détient toutes les données concaténées.

Avant de transmettre ses données au nœud relais, chaque CH vérifie sa liste de nœuds membres. Si cette dernière est vide il envoie ses données vers le nœud désigné comme prochain saut lors de la phase de formation de clusters. Dans le cas contraire, le CH collecte les données de ses noeuds membres, les agrège puis les communique à son noeud relais. Les données reçues par le nœud relais, seront également combinées puis envoyées aux autres nœuds relais de niveau supérieur et ainsi de suite jusqu'au niveau le plus supérieur de la hiérarchie des clusters. Une fois arrivée, les CHs de haut niveau transmettront les données reçues directement à la station de base.

b).Algorithmme

```

Soit relais : variable indiquant le statut du nœud courant ; { vrai si le nœud est un relais } ;
Soit mes_ membres : liste détenant les nœuds membres qui ont envoyé un message de jointure Join_msg ;
Soit les fonctions :
    Create schedule () ; { fonction qui permet la création des frames par les CHs } ;
    Envoyer_slot_TDMA () ; { fonction qui envoie le paquet contenant le slot de temps au nœud membre propriétaire }
    Envoyer_données () ; { fonction utilisée pour passer les données au nœud approprié ; CH ou relais } ;
    Actif () ; { fonction indiquant l'état du nœud : le nœud est en mode actif } ;
    Sommeil () ; { fonction permettant au nœud de basculer en mode sommeil } ;

Fonction Créer_schedule : { créer un plan de communication TDMA puis assigner à chaque nœud membre son slot de temps }
Lors de la réception d'un message Join par un nœud i depuis un nœud j :
    Si ( mes_ membres =  $\Phi$  ) Alors
        Envoyer_données () ;
    Sinon
        Create schedule () ;
        Envoyer_slot_TDMA () ; { envoyer le slot de temps au nœud j }
    Fin si
Fin
Fonction Recv_slot_TDMA:{ traitement du message contenant le slot , reçu par le nœud j depuis le nœud i}

{ le nœud j récupère son slot de temps assigné par son CH ( le nœud i ) }
    Envoyer_données () ;
    Si ( relais ) Alors
        Actif () ;
    Sinon
        Sommeil () ;
    Fin si
Fin

```

4.3.3 La maintenance des clusters

Dans le but de s'adapter aux changements de la topologie du réseau, des messages de reconfiguration pour le réseau sont lancés périodiquement par la station de base afin de reconnaître les nouveaux noeuds ajoutés ou ceux disparus (les nœuds qui ont consommé leur énergie). Cela va permettre aux différents nœuds de mettre à jour les informations de leurs voisins stockées dans leurs tables de voisinage.

D'autre part, les CHs reçoivent plus de paquets et consomment plus d'énergie pour les transmettre aux destinataires, ils sont donc ceux dont l'énergie sera épuisée rapidement s'ils sont élus pour une longue période. De même, si un nœud joue le rôle de CH le plus longtemps possible, même s'il ne possède pas par exemple le poids maximum dans son propre cluster, il perd son rôle une fois sa batterie est épuisée. Pour cela, le fonctionnement de LMEER est dynamiquement segmenté en cycles en relançant le processus de clustering périodiquement. Ainsi, une répartition efficace de la consommation énergétique et une utilisation équitable de la bande passante sont assurées grâce à une réélection périodique du CH. Ce mécanisme permet de prolonger la durée de vie du nœud et celle du réseau en entier.

4.4 Simulations et résultats

Dans cette partie, nous présentons les outils utilisés pour l'évaluation des performances du protocole LMEER puis nous discutons par la suite les résultats obtenus. Ces résultats sont comparés avec ceux obtenus par le protocole LEACH. Ce dernier est choisi comme protocole de comparaison. La raison de ce choix s'explique par le fait que LEACH est considéré comme protocole de référence dans le domaine du routage hiérarchique et un standard de comparaison pour les protocoles de clustering publiés dans la littérature. Son efficacité à remplir les objectifs espérés d'un protocole de clustering a motivé son utilisation dans la comparaison des performances de la majorité des protocoles de clustering proposés dans la littérature. Mais avant d'aborder l'environnement de simulation choisi et traiter ses résultats, nous allons passer en revue les différents outils disponibles dans la littérature pour simuler les réseaux de capteurs sans fil.

4.4.1 Environnements de simulation

La simulation est la méthode d'évaluation de performances la plus prédominante dans le domaine des réseaux de capteurs. Elle est largement utilisée pour évaluer les nouvelles architectures et protocoles de communication, car elle permet de tester à moindre coût ces nouveaux protocoles et d'anticiper les problèmes qui pourront surgir durant leur implémentation réelle. Pour le faire, elle construit un modèle du système réel en représentant toutes ses entités, leur comportement et leur interaction pour mener en suite des expériences sur ce modèle avec une simple modification des paramètres de simulation dont les résultats seront facilement analysables et interprétables.

Pour simuler les réseaux de capteurs, deux types de simulateurs sont utilisés: les simulateurs dédiés et les simulateurs de réseaux généraux. Les simulateurs dédiés sont conçus particulièrement pour les réseaux de capteurs. Or que, les plus généraux ont été conçus pour modéliser et simuler les réseaux classiques : les réseaux filaires et les réseaux ad hoc. Pour ces réseaux la consommation d'énergie n'était pas encore la préoccupation majeure alors qu'elle est la contrainte la plus cruciale dans les réseaux de capteurs. Pour cela, les simulateurs sont adaptés pour pouvoir prendre en charge les particularités des réseaux de capteurs.

Les simulateurs les plus utilisés sont NS-2, TOSSIM , OMNET ++, GlomoSim et J-Sim.

NS-2: Network Simulator2 [11, 84] est le simulateur à évènements discrets¹⁰ le plus répandu dans le domaine des réseaux. Il était initialement conçu pour les réseaux filaires, le support des réseaux sans fil a été ajouté ultérieurement. C'est un outil open source avec une bibliothèque de protocoles très riche où chacun peut ajouter sa contribution. NS-2, repose sur deux langages : OTCL(Object Tool Command Language) qui représente l'extension objet au langage de commande Tcl, utilisé pour écrire des scripts décrivant la topologie du réseau et les communications entre les noeuds, et pour l'implémentation des différents protocoles et modules le C++ est adopté comme langage de programmation.

La simulation doit d'abord être saisie sous forme de script (.tcl) que NS-2 l'utilisera pour produire un fichier contenant les résultats. Deux types de sorties sont possibles sous ns-2 : fichier trace avec l'extensions (.tr) et un autre (.nam). Le premier fichier s'utilise dans l'évaluation des performances des protocoles simulés afin d'analyser les résultats recueillis, et cela par des scripts écrits en AWK ou en PERL pour extraire les données du fichier trace(.tr) et un autre outil pour mettre en forme ces données sous forme de graphes, gnuplot et excel par exemple. Alors que le deuxième fichier est exploité par l'outil NAM "Network Animator" interface graphique très simple, permettant la visualisation du comportement des protocoles et la perception de l'état du réseau.

Les composants et les modules qui constituent un réseau ne sont pas toujours bien séparés dans nS2, ce qui rend l'ajout de nouveaux modèles très difficile à cause des dépendances entre ces modules. De plus, la représentation des résultats dans un seul fichier retraçant l'ensemble des envois, réceptions et suppressions de paquets rend l'analyse des résultats peu aisée et nécessite la maîtrise de beaucoup d'outils.

TOSSIM: TinyOS SIMulator [85] est un simulateur dédié à TinyOS, système d'exploitation destiné aux réseaux de capteurs (voir §1.1.2.2) basé sur la programmation par événements. TOSSIM essaye de tirer parti du mode d'exécution de TinyOS afin de proposer un simulateur efficace, fiable et fidèle. Le mode d'exécution de TinyOS est dirigé par les événements, il se calque bien sur un simulateur à événement discret. TOSSIM contient un modèle abstrait de chaque composant du matériel d'un noeud. Pour une simulation TOSSIM, on utilise le même code que celui destiné au noeud cible mais cette fois-ci TOSSIM émule le comportement du matériel, en utilisant les modèles des composants. Simuler exactement le code qui tournera sur les noeuds permet aux utilisateurs de déboguer, tester et analyser l'implémentation finale des algorithmes, en fournissant ainsi des résultats plus réalistes. Avec un module graphique de visualisation détaillée (TinyViz), les résultats pourraient ensuite être facilement compréhensibles.

Malgré que cette notion d'abstraction du matériel est tout à fait intéressante, TOSSIM dans sa première version ne permet pas d'estimer l'énergie consommée. Pour pallier à cette lacune, une extension de TOSSIM : PowerTossim contenant un modèle de consommation d'énergie est proposée. Selon ce modèle, pour pouvoir estimer la consommation d'énergie il faut connaître les consommations des différents composants d'un noeud suivant leurs états. En d'autre terme, on doit connaître l'état de chaque composant d'un noeud pendant la simulation. Chose compliquée à réaliser et qui ne permet pas de varier la précision de ce modèle de consommation. Toutefois, ces deux simulateurs ne conviennent que pour des applications écrites en TinyOS.

¹⁰ désigne la modélisation d'un système réel tel qu'il évolue dans le temps, par une représentation dans laquelle les grandeurs caractérisant le système (variables) ne changent qu'en un nombre fini ou dénombrable de points isolés dans le temps. Ces points sont les instants où se passent les événements.

OMNET ++ : Objective Modular Network Test-bet in C++ [86], a été conçu en vue de fournir aux chercheurs un meilleur environnement de simulation à événements discrets. Son domaine d'application principal est la simulation des réseaux de communication, mais en raison de son architecture générique et souple, il est utilisé avec succès dans d'autres domaines comme la simulation de systèmes complexes, réseaux de files d'attente, la modélisation des systèmes multiprocesseurs et pas mal d'autres systèmes distribués.

OMNET ++ permet la conception des modèles de simulation modulaire, qui peuvent être combinés et réutilisés avec souplesse. De plus, l'approche orientée objet de ce simulateur permet l'extension flexible des classes de base fournies dans le noyau du simulateur via la technique d'héritage. Une bibliothèque de simulation complète qui inclut un support pour les entrées/sorties, statistiques, collecte de données, la présentation graphique des données de simulation, des générateurs de nombres aléatoires et structures de données est proposée dans ce simulateur.

Le noyau de simulation OMNET ++ utilise le langage C ++ qui facilite son intégration avec d'autres applications et environnements de développement. Et pour construire les modèles de simulation : les modules composés constitués de modules simples développés en C++, on fait recours au langage de haut niveau Ned sans l'utilisation de scripts, ce qui rend sa configuration pour les différentes simulations très facile.

Le fait qu' OMNET ++ est non prévu particulièrement pour les réseaux de capteurs sans fil, oblige ses utilisateurs de l'enrichir avec des modèles implémentant le comportement spécifique de ce type de réseaux ainsi que ses composants.

GloMoSim :Global Mobile Simulator[87] est un environnement de simulation pour les réseaux mobiles sans fil qui a été développé à UCLA (University of California, Los Angeles). Ce simulateur a été conçu en exploitant la capacité de la simulation parallèle fournie par PARSEC (PARallel Simulation Environment for Complex systems) une extension du langage C. Il fournit la possibilité d'exécuter un modèle de simulation avec plusieurs protocoles dans différentes architectures parallèles.

La plupart des réseaux actuels sont construits suivant une approche multicouches, similaire aux sept couches du modèle OSI. De ce fait, GloMoSim se présente ainsi sous la forme de plusieurs fichiers sources organisés en couches, et fournissant plusieurs fonctions permettant les communications entre ces différentes couches tout en définissant des APIs communes entre les couches voisines. Chaque API spécifie les services et les paramètres échangés entre les couches voisines. Cette façon de faire permet aux utilisateurs une intégration rapide des modules développés dans des couches différentes par des groupes de travail différents.

De plus, plusieurs choix de conception sont offerts pour les modèles de propagation du signal radio, protocoles MAC, les implémentations d'UDP et TCP et les protocoles de routage sans fil. Certes, GloMoSim est un bon simulateur de réseaux mobiles IP. Toutefois, plusieurs problèmes sont rencontrés sur les formats de paquets, les modèles d'énergie et les protocoles MAC représentatifs de ceux utilisés dans les réseaux de capteurs. En outre, GloMoSim ne fournit aucun support pour les capteurs, les phénomènes physiques ou les conditions environnementales, et la précision de ses résultats d'évaluation de performance des protocoles de réseaux de capteurs sont discutables.

J-Sim: Java-Simultaor [83,88] est un simulateur open source calqué sur ns-2 et fonctionnant avec la plateforme JAVA, développé par Ohio State University et UIUC: l'Université de l'Illinois à Urbana-Champaign.

Ce simulateur utilise quasi-indifféremment deux langages: Java et Jacl. Le langage Java sert à implémenter les différentes classes des packages J-sim. Alors que le Jacl (Tcl pour Java) est utilisé pour la mise en place des scénarios de simulation.

J-Sim est basé sur la notion de composants autonomes ACA: autonomous component architecture; chaque entité est appelée composant. Un composant est une entité indépendante représentant un objet physique (une batterie, un module radio, etc.), ou logique (un protocole de routage, un modèle de mobilité, etc.). Chaque composant possède plusieurs ports qui sont utilisés pour communiquer avec d'autres composants par l'envoi/ réception de données sur ces ports.

Cette répartition en composants, rend J-Sim une plate-forme véritablement extensible et un environnement réutilisable. Il offre également une analyse des résultats plus aisés que le simulateur NS-2 par exemple. Plus important encore, il permet de simuler des réseaux de l'ordre de 1000 noeuds via l'architecture suffisamment bien structurée de ses composants. En outre, J-Sim gère correctement l'aspect consommation d'énergie grâce au module batterie et les modèles de consommation d'énergie offerts, sans oublier le modèle de mobilité, canaux de communication sans fil et les supports physiques disponibles aussi dans ce simulateur.

Toutefois J-Sim souffre peut être de sa jeunesse car il est moins connu dans la communauté de recherche que ns2 ou OMNET ++ (le support des réseaux de capteur n'est fixé qu'en 2006). De plus, ce simulateur est très peu documenté ce qui rend son utilisation un peu complexe.

4.4.2 Choix de l'environnement de simulation

Pour évaluer les performances du protocole LMEER, le simulateur J-sim a été utilisé comme outil d'expérimentation et validation. Généralement le choix d'un simulateur dépend de ses performances d'exécution et ses réponses aux besoins de l'utilisateur. Pour cela, notre choix de J-sim se justifie par le fait que ce simulateur libre et ouvert est très puissant dans la gestion de la consommation d'énergie le point le plus crucial dans les réseaux de capteurs sans fil. Et comme déjà mentionné précédemment, J-sim offre des modèles de batteries permettant de prendre en charge différents modèles de consommation d'énergie (voir en bas le détail de l'un des modèle de consommation d'énergie offert par J-sim). Ainsi, son architecture basée composant, lui permet d'être modulaire, hiérarchique, facilement flexible et extensible pour supporter des nouveaux modèles, scénarios et protocoles. De plus, ce simulateur est développé entièrement en Java, langage orienté objet satisfaisant aux besoins de l'immense majorité des développeurs, caractérisé par sa fiabilité, réutilisabilité, architecture neutre et indépendante, performances élevées et multithread. Le langage Jacl est utilisé pour intégrer les classes java dans les scripts des scénarios de simulation. Il est possible aussi d'utiliser d'autres langages de script tels que Perl, Tcl ou Python. En outre, J-sim est très scalable, il peut simuler jusqu'à 1000 noeuds avec moins deux ordres de grandeur inférieure de mémoire allouée pour effectuer la simulation comparé à NS-2. Cela fournit une très grande performance pour effectuer des simulations à grande échelle.

La suite de ce paragraphe apporte des détails sur le modèle de consommation d'énergie utilisé dans l'environnement de simulation J-sim.

Le modèle de consommation d'énergie

Le modèle de consommation d'énergie radio utilisé dans LMEER est celui développé par Henzelmen et al pour le protocole LEACH [33]. Ce modèle de consommation d'énergie (voir Figure 4.3) est résumé comme suit :

L'énergie consommée pour émettre un message de k bits avec une distance d est donnée par la formule suivante : $E_{Tx}(k, d) = E_{Tx}(l) + E_{Tx_amp}(k, d)$

Pour recevoir un message de k bits le récepteur consomme : $E_{Rx}(k) = E_{Rx_elec}(k) = k \cdot E_{elec}$

Où :

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

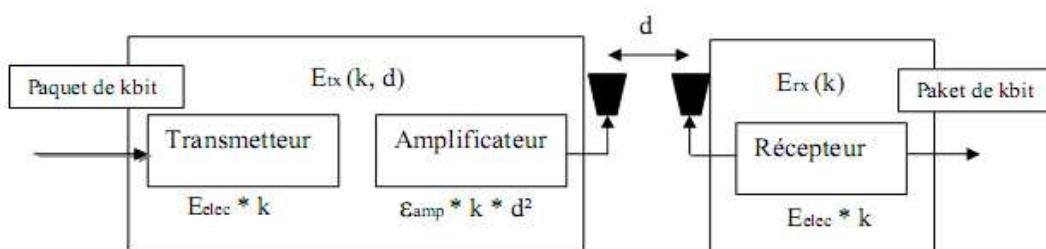


Fig. 4.3 modèle de consommation d'énergie.

La consommation d'énergie dépend de la distance entre l'émetteur et le récepteur et des caractéristiques radios de E_{elec} et E_{Tx_amp} choisis lors de la phase de conception matérielle.

4.4.3 Les paramètres de simulation

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

Paramètre	Valeur
Nombre de Noeuds	De 50 à 400 nœuds
Durée de simulation (s)	500
Surface de Simulation (mXm)	100x100
Placement des noeuds	Aléatoire
Nombre de Stations de Base	1
Nombre de Nœuds Cibles	3
Portée de transmission (m)	30
Paquets de données générés par les sources (packets/s)	Un paquet de donnée tout les 0.2 sec.
Energie initiale (Joules)	2 J
Puissance de transmission radio (watt)	0.028
Puissance de réception radio (watt)	0.036

Tab. 4.1 – Paramètres de simulation

Les capteurs utilisés dans la simulation sont considérés comme homogènes : possédant la même quantité d'énergie initiale, capacités de calcul et mémoire, portée de transmission et équipés par les mêmes interfaces de communication IEEE 802.11. De plus, l'énergie de la station de base est considérée comme illimitée. Ces capteurs sont déployés à l'intérieur d'un carré de taille 100 par 100 créant ainsi des topologies contenant 50 à 400 capteurs. La position des nœuds dans les topologies générées est réalisée d'une manière aléatoire.

4.4.4 Les métriques d'évaluation de performances

Pour quantifier les performances des deux protocoles LMEER et LEACH, quatre métriques sont sélectionnées: taux moyen de livraison de paquets, moyenne d'énergie consommée, le délai moyen de bout en bout et la durée de vie du réseau. Ces métriques sont affectées par des facteurs comme le nombre de nœuds utilisés dans les scénarios de simulation, le temps de simulation, la présence de défaillance...etc.

Les valeurs moyennes des taux de livraison de paquets, énergie consommée et le délai de bout en bout sont obtenues par l'exécution de 30 simulations indépendantes pour chaque scénario.

Moyenne d'énergie consommée (MEC)

L'un des principaux critères de performances pour un capteur est la durée d'utilisation efficace de son énergie embarquée avant de l'épuiser. Parce qu'une fois qu'elle sera épuisée, il n'y a pas d'autre moyen de recharger à nouveau sa batterie, si il été déployé à des endroits inaccessibles et n'utilisant pas de cellules solaires.

Ainsi, l'Énergie Consommée (MEC) mesure en moyenne la quantité d'énergie consommée par un capteur. Elle est obtenue par la somme de joules de tous les nœuds au nombre de nœuds du réseau. Mathématiquement, on peut l'exprimer ainsi :

$$EC = \frac{\sum_{i=1}^n E_i}{n}$$

Où :

E_i : l'énergie consommée pour un capteur i. Elle présente la différence entre l'énergie initiale et son énergie résiduelle. Et n : est le nombre de capteurs dans le réseau.

Et pour obtenir la moyenne d'énergie consommée MEC, pour chaque scénario 30 simulations seront exécutées :

$$MEC = \frac{\sum_{i=1}^{30} EC_i}{30}$$

Taux Moyen de Livraison de Paquets de données (TMLP)

Le taux de livraison de paquets (TLP) est le rapport entre le nombre de paquets délivrés avec succès à la station de base à celui émis par les noeuds sources. Cette métrique nous permet d'avoir une idée sur les performances du protocole en termes de routage et son degré de succès pour fournir les paquets de données captées à la station de base. Mathématiquement TLP est donné par :

$$TLP = \frac{\text{Le nombre de paquets reçus par la station de base}}{\text{Le nombre de paquets émis par les noeuds sources}}$$

Et pour calculer le taux moyen de livraison de paquet TLMP, on divise la somme des taux de livraisons totales sur le nombre d'exécutions d'un scénario. Comme déjà mentionné 30 simulations seront relâchées pour chaque test d'un scénario. Et donc TMLP peut être exprimé par :

$$TMLP = \frac{\sum_{i=0}^{30} TLP}{30}$$

Le délai moyen de bout en bout (DBBM)

Le délai de bout en bout (DBB) est exprimé par la moyenne des différences entre le temps de réception d'un paquet de donnée au niveau du nœud destinataire et le temps de son émission par le nœud source, pour tous les paquets de données bien reçus dans le réseau.

$$DBB = \frac{\sum_{i=1}^n Temps\ de\ reception\ du\ paquet_i - Temps\ d'émission\ du\ paquet_i}{Nombre\ total\ de\ paquets\ reçus}$$

DBB est un critère déterminant dans les réseaux de capteurs car il permet à l'usager d'intervenir rapidement à l'occurrence d'un événement. Ce délai inclut tous les retards causés lors de l'établissement des routes, mise en mémoire tampon ainsi que les retards de retransmission au niveau de la couche MAC lors d'une collision.

Alors, le délai moyen de bout en bout est donné par la somme de l'DBB de tous les paquets transitant dans le réseau sur le nombre total d'exécutions du scénario de simulation :

$$DBBM = \frac{\sum_{i=1}^{30} DBB_i}{30}$$

Durée de vie du réseau

Pour mesurer la durée de vie du réseau, nous avons choisi de suivre l'évolution du nombre de noeuds en vie au cours du temps. En effet, cette métrique nous donne un aperçu sur la façon et la fréquence de la mort des noeuds. Elle peut être utilisée pour déterminer la durée de vie du réseau selon le temps qui s'écoule avant que le dernier nœud en vie dans le réseau épouse son niveau d'énergie.

Pour l'évaluation des performances des deux protocoles LEACH et LMERR nous avons suivi l'évolution de 100 noeuds.

Il est à noter que l'énergie initiale utilisée dans les scénarios de simulation de la durée de vie du réseau est de 0,5 Joules pour chaque nœud capteur. Ceci est pour optimiser le temps de simulation et le processus de convergence car avec 2 Joules comme énergie initiale le temps de simulation s'étend à l'ordre de jours.

4.4.5 Discussion des résultats

Dans ce qui suit, nous allons présenter et analyser les résultats de simulation obtenus suivant les métriques de performances discutées précédemment.

Les simulations sont effectuées sur des capteurs déployés de manière aléatoire et pour différents pourcentages de défaillance dans le réseau avec une sélection aléatoire des nœuds défaillants.

Dans tous les scénarios de simulations générés le protocole LEACH est comparé avec les deux configurations proposées pour LMEER afin de différencier leurs performances. On note LMEER-D pour le fonctionnement de LMEER avec une configuration selon la distance à la station de base et LMEER-S pour spécifier la configuration du réseau avec le protocole LMEER en fonction du nombre de sauts par rapport à la station de base.

1. Effet de la densité du réseau sur les performances des protocoles LMEER et LEACH

Afin d'examiner l'influence de la densité (le nombre moyen de voisins par nœud) sur les performances des protocoles LMEER (LMEER-D, LMEER-S) et LEACH, nous avons mené des simulations en variant le nombre de nœuds déployés dans le réseau (de 50 à 400 nœuds par un pas de 50 noeuds) augmentant ainsi la densité du réseau.

Moyenne d'énergie consommée (MEC)

Sur la figure 4.4, on peut voir que le protocole LMEER (LMEER-S et LMEER-D) surperforme le protocole LEACH, en marquant jusqu'à moins de 10% de moyenne d'énergie consommée (pour 400 noeuds par exemple) par rapport au protocole LEACH. Ce gain en énergie est grâce à l'équilibrage dans la distribution de la charge qu'a présenté LMEER-S et LMEER-D entre les CHs des différentes couches, ainsi qu'aux différentes techniques utilisées dans la formation des clusters et le choix des nœuds CHs. Ces techniques prennent en compte les paramètres influant sur la bonne gestion de la ressource énergétique à savoir le degré d'un nœud et son énergie résiduelle. Alors que le protocole LEACH n'exerce aucun contrôle sur le choix des nœuds CHs et la sélection est réalisée d'une manière complètement aléatoire.

En outre, les techniques de transmission utilisées entre les noeuds pour router les données à la station de base dans les deux protocoles LMEER-S et LMEER-D contrôlent efficacement l'énergie de transmission, en adoptant un routage multi-sauts avec des mécanismes pour choisir les meilleurs noeuds relais pour réaliser ce routage. Or, le routage de données dans le protocole LEACH est effectué via un seul saut ce qui consomme beaucoup d'énergie pour atteindre la station de base.

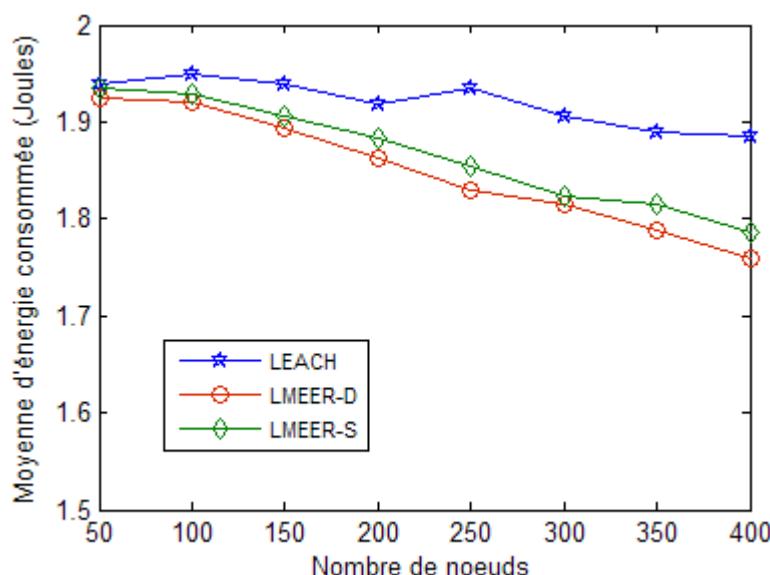


Fig. 4.4 Moyenne d'énergie consommée : LMEER-D vs LMEER-S vs LEACH

Egalement, on peut remarquer que l'accroissement du nombre de nœuds déployés dans le réseau induit une augmentation dans la consommation d'énergie dans les trois protocoles. Ceci est attendu car le fonctionnement de ces trois protocoles est strictement basé sur des interactions locales entre les nœuds voisins et ne nécessite en aucun cas des informations globales sur l'ensemble des nœuds du réseau. Par conséquent, l'augmentation du nombre de capteurs dans le réseau augmente le nombre de voisins pour chaque noeud, ce qui provoque plus de messages échangés entre ces noeuds et consomme ainsi plus d'énergie dans la communication.

Taux Moyen de Livraison de Paquets de données (TMLP)

Comme l'illustre la figure 4.5, le taux moyen de livraison de paquets pour les trois protocoles LEACH, LMEER-D et LMEER-S varie entre 20% et 21 % avec un léger avantage marqué par le protocole LMEER-D par rapport aux protocoles LEACH et LMEER-S. On peut remarquer aussi que le TMLP le plus bas et celui présenté par LMEER-S : ne dépassant pas 20.36% lorsque le nombre de noeuds est inférieur à 200 noeuds. Mais au delà de ce nombre, le TMLP de LMEER-S est légèrement meilleur que celui du protocole LEACH et ses performances en terme d'acheminement de paquets s'améliorent légèrement avec l'augmentation du nombre de noeuds. Ces résultats sont très raisonnables et traduisent bien l'efficacité de la politique de communication adoptée dans les deux protocoles LMEER-D et LMEER-S, en acheminant les paquets de données par plusieurs sauts. En effet, la densification du réseau augmente le nombre de voisins par nœud ce qui offre une variété dans le choix des nœuds relais utilisés dans la transmission des paquets de données à la station de base et augmente ainsi le taux de succès pour l'atteindre. Par ailleurs, nous pouvons bien constater que le protocole LEACH est influencé par la densité du réseau et son taux de paquets livrés avec succès pour la station de base diminue en augmentant le nombre de noeuds déployés dans le réseau. Ceci est dû principalement au type de communication utilisé entre les CHs et la station de base qui est établie via un seul saut, ce qui engendre plus de collisions entre les paquets de données générés dans le réseau avec comme résultat un TMLP plus bas .

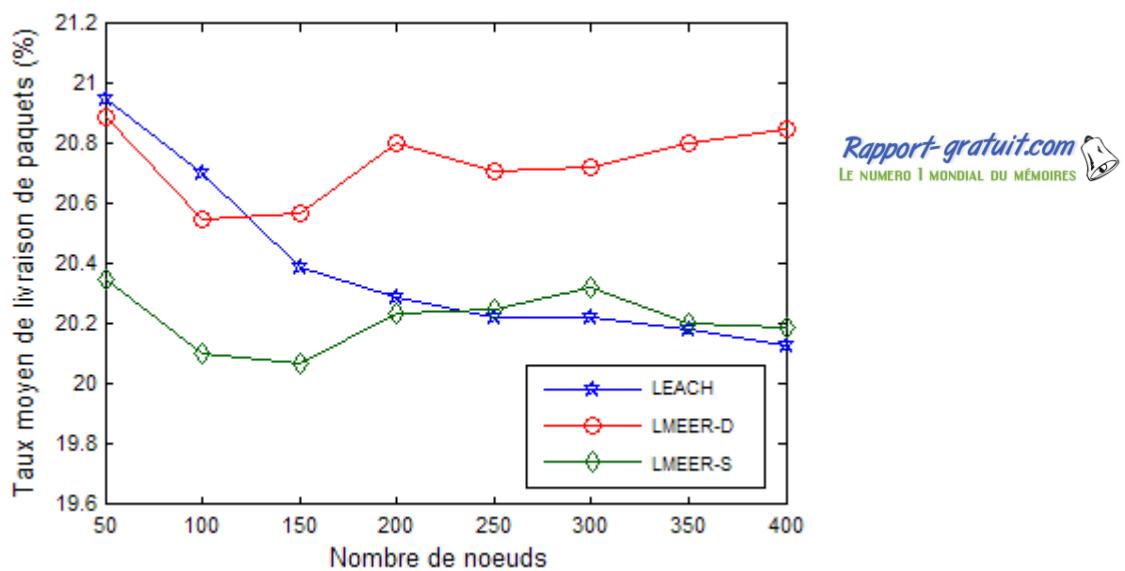


Fig.4.5 Taux moyen de livraison de paquets LMEER-D vs LMEER-S vs LEACH

Le délai moyen de bout en bout (DBBM)

Comme le montre la figure 4.6, les deux protocoles LMEER-S et LMEER-D expriment en moyenne le même délai de transmission de bout en bout pour acheminer les paquets de données à la station de base avec un léger avantage pour le protocole LMEER-S par rapport à LMEER-D, quand le nombre de noeuds déployés dans le réseau dépasse 300 noeuds. Cet avantage marqué par le protocole LMEER-S revient au nombre élevé de paquets reçus par la station de base. Et comme le DBBM n'est que le rapport entre les retards générés dans le réseau et le nombre de paquets reçus avec succès par la station de base, alors ces résultats sont bien attendus et confirment l'efficacité du mécanisme utilisé dans la livraison des paquets et ainsi les résultats obtenus dans la figure 4.5.

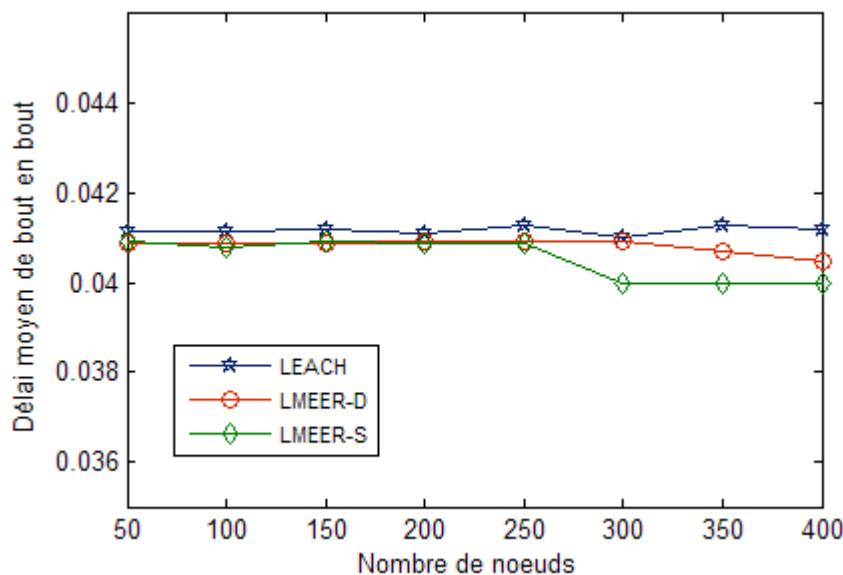


Fig.4.6 Délai moyen de bout en bout de LMEER-D vs LMEER-S vs LEACH

De plus, les deux protocoles LMEER-S et LMEER-D surperforment légèrement le DBBM présenté par le protocole LEACH en offrant un DBBM plus bas, et ce quelque soit le nombre de noeuds déployés dans le réseau. Cela peut être expliquer par le nombre élevé de retransmissions nécessaires dans LEACH pour réussir l'envoie d'un paquet de données, après les éventuelles collisions causés par les interférences entre les noeuds qui sont non uniformément distribués dans le réseau, car LEACH n'assure aucun contrôle sur la distribution des CHs et leur noeuds membres (la répartition des CHs est réalisée d'une manière aléatoire). Par conséquent, ces retransmissions induisent une augmentation dans les délais d'attente pour accéder au canal de communication et comme effet, accroître le délai de transmission de bout en bout.

Durée de vie du réseau

Pour pouvoir étudié la durée de vie du réseau en fonction du nombre de noeuds déployés nous avons suivi l'évolution de 50, 100 et 200 noeuds dans le temps.

La figure 4.7 (a, b et c) présente l'effet de la densité sur la durée de vie du réseau afin d'examiner l'efficacité des protocoles LEACH et LMEER (LMEER-S, LMEER-D) à maximiser la durée de vie des noeuds capteurs et, par conséquent, celle du réseau en entier.

Il est clair, sur ces trois figures (a, b et c), que la durée de vie du réseau offerte par le protocole LMEER (LMEER-S, LMEER-D) superforme celle du protocole LEACH, et ce dans

les trois configurations testées. L'explication sera la même que celle donnée pour commenter les résultats du test de la moyenne d'énergie consommée. En effet, la prise en compte de la contrainte énergétique en adoptant la métrique d'énergie résiduelle lors de la sélection des nœuds CHs, la formation des clusters et lors du choix des nœuds relais utilisés dans la transmission des données à la station de base, a apporté un équilibrage dans la consommation d'énergie des nœuds sélectionnés CHs et a augmenté la durée de vie du réseau en entier.

Ce que l'on peut également constater c'est que la durée de vie du réseau (liée à l'épuisement de l'énergie du dernier nœud capteur en vie) diminue proportionnellement avec la densification. Ces résultats s'expliquent aisément par l'augmentation des interférences entre les nœuds voisins avec l'augmentation du nombre de nœuds déployés dans le réseau. Par conséquent, la contention sur le canal de transmission sans fil devient de plus en plus forte dans les trois protocoles ce qui cause plus de consommation d'énergie dans la communication et l'échange des messages, avec comme résultat une dégradation dans la durée de vie du réseau. Mais malgré cela, les résultats montrent clairement que notre approche apporte une amélioration importante à la durée de vie du réseau par rapport à LEACH et ce dans les trois topologies testées.

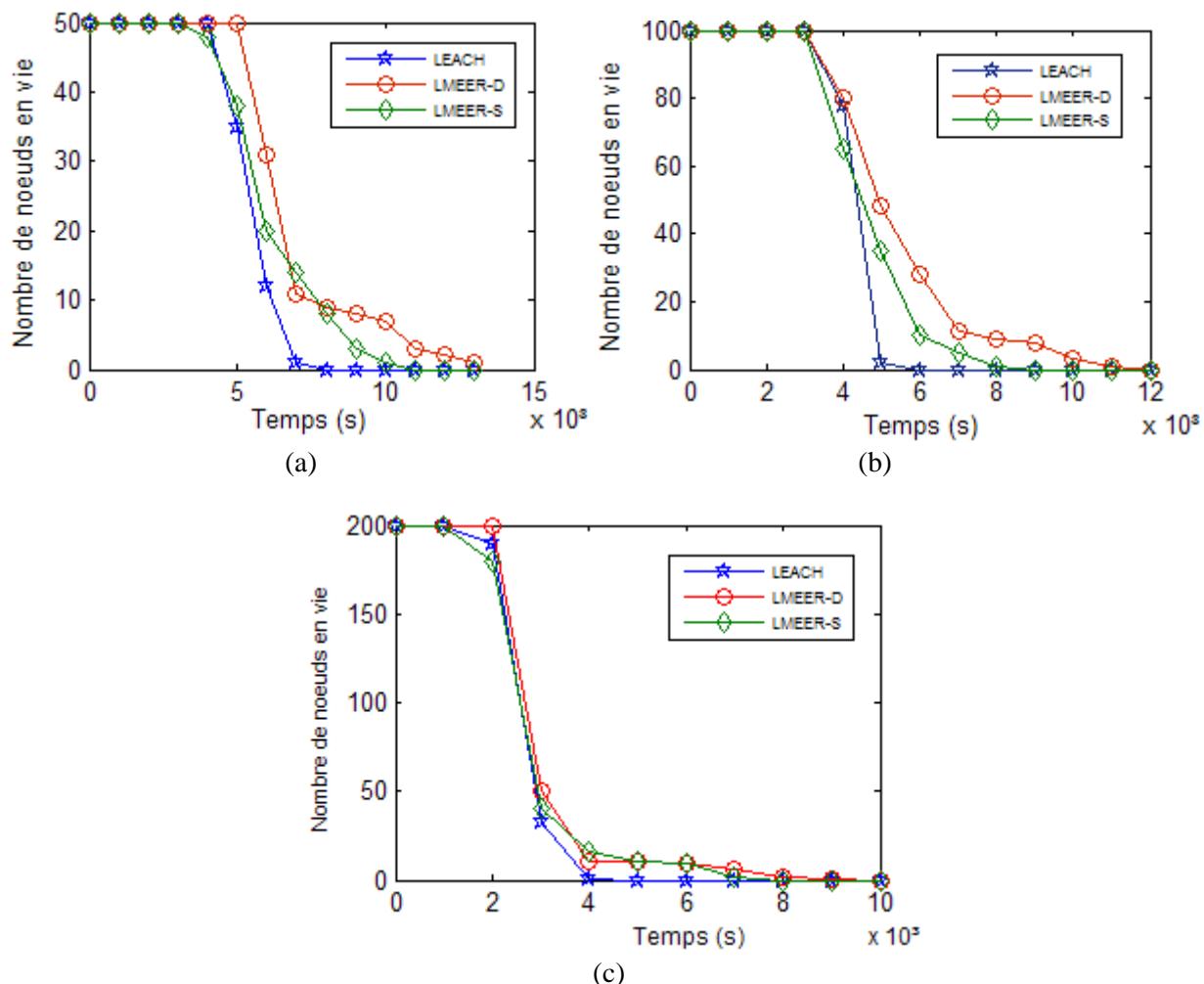


Fig.4.7 Taux moyen de livraison de paquets LMEER-D vs LMEER-S vs LEACH :
(a) Pour 50 nœuds , (b) pour 100 nœuds (c) pour 200 noeuds.

2. Etude de l'effet de la défaillance des capteurs sur les protocoles LMEER et LEACH

Dans cette deuxième série d'expérimentation, nous nous intéressons à l'étude de l'influence de la défaillance des nœuds sur les performances des protocoles LMEER-D, LMEER-S et LEACH. Dans cette perspective, nous avons varié le taux de défaillance des nœuds dans le réseau entre 10%, 30% et jusqu'à 50%.

Moyenne d'énergie consommée (MEC)

La figure 4.7 à savoir (a), (b) et (c) représente la moyenne d'énergie consommée dans les trois protocoles (LMEER-S, LMEER-D et LEACH) en fonction du nombre de nœuds déployés dans le réseau et ce, avec la présence d'un pourcentage variant de nœuds défaillants.

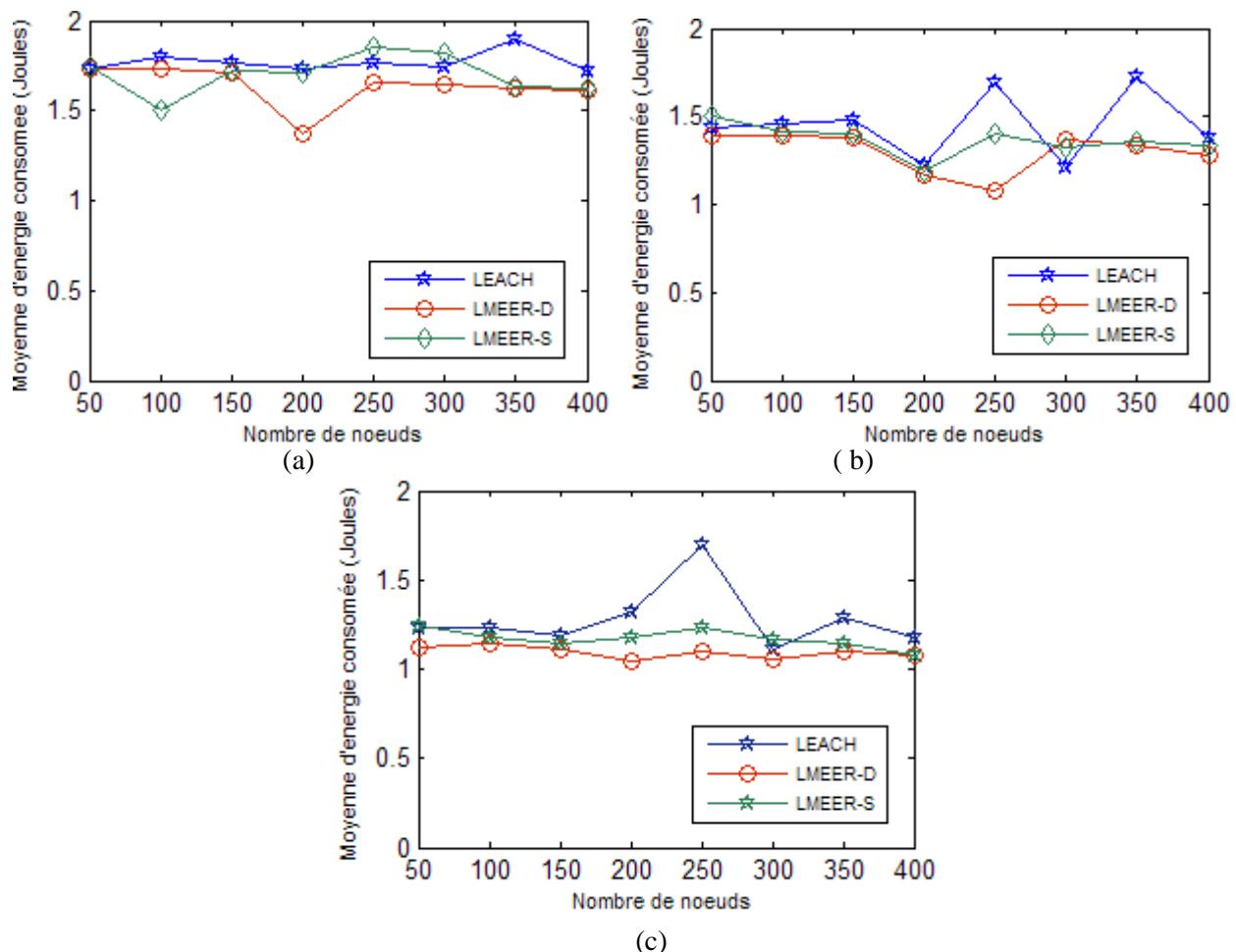


Fig.4.8 Moyenne d'énergie consommée avec la présence d'un pourcentage de :
(a) 10% de nœuds défaillants (b) 30% de nœuds défaillants
(c) 50% de taux de nœuds défaillants.

L'augmentation du taux de défaillance dans le réseau a causé une décroissance dans la moyenne d'énergie consommée (MEC) dans les trois protocoles LMEER-S, LMEER-D et LEACH. Par exemple, dans le cas de 50 nœuds la MEC dans une topologie de réseau exempt de défaillance était 1.95J (voir la figure 4.5) alors qu'elle devient respectivement 1.73J, 1.50J et 1.20J pour les taux de défaillance 10%, 30% et 50 %. Cela s'explique aisément par le fait que

l'augmentation du taux de défaillance a pour effet de diminuer le nombre de nœuds fonctionnels dans le réseau et par conséquent ceux choisis comme nœuds CHs, ce qui va accroître la charge des autres nœuds CHs fonctionnels et consomme plus d'énergie pour la gestion des clusters. Le protocole LEACH est le plus sensible à la variation des taux de défaillance car aucune technique d'équilibrage de charge n'est appliquée pour sa répartition ni sur les tailles des clusters formés. De même la défaillance des nœuds situés dans les couches proches de la station de base pour les deux protocoles LMEER-S et LMEER-D a provoqué une charge supplémentaire pour le reste des nœuds CHs fonctionnels dans ces couches avec comme résultat, plus de consommation d'énergie. Néanmoins, la MEC des deux protocoles LMEER-S et LMEER-D était presque toujours inférieure que celle du protocole LEACH, ce qui confirme l'efficacité de la méthode d'équilibrage de charge utilisée dans ces deux protocoles en se rattachant au CHs dotés de plus d'énergie et ayant moins de nœuds membres, et apportant ainsi une répartition en tailles inégales pour les clusters avec plus de contrôle sur la charge de chaque nœud CH.

Taux Moyen de Livraison de Paquets (TMLP)

Sur les figures suivantes (voir figure 4.9, a, b et c), nous expérimentons l'impact de la variations du taux de défaillance sur le taux moyen de livraison de paquets (TLMP) par rapport au nombre de nœuds déployés dans le réseau

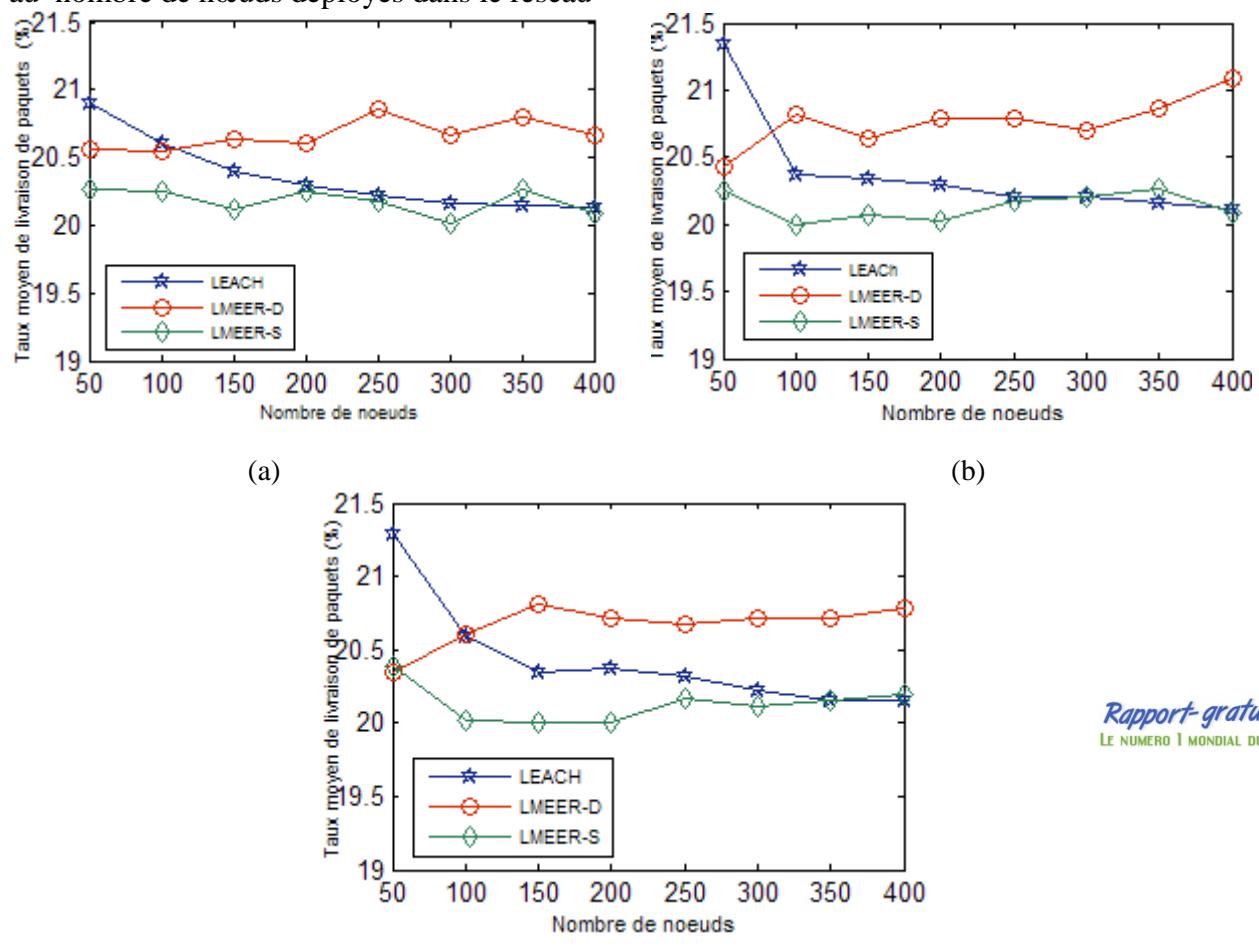


Fig.4.9 Taux moyen de livraison de paquets avec la présence d'un pourcentage de :
(a) 10% de nœuds défaillants (b) 30% de nœuds défaillants
(c) 50% de taux de nœuds défaillants.

La présence d'un pourcentage variant de défaillance dans le réseau n'a pas dégradé les performances du protocoles LMEER-D en termes de taux de livraison de paquets (TMLP). Ceci peut être remarqué sur (a), (b) et (c) de la figure 4.9 où le taux moyen de livraison de paquets marqué par LMEER-D superforme celui du protocole LEACH ainsi que celui du LMEER-S et ce, malgré l'accroissement du pourcentage des nœuds défaillants dans le réseau. De plus, les TMLPs marqués par ces trois protocoles LEACH, LMEER-D et LMEER-S sont légèrement différents de leurs TMLPs marqués dans une topologie sans défaillance (voir figure 4.6). Ceci est aisément justifiable vu la mise à jour dynamique réalisée sur les clusters et les nœuds choisis CHs, en exécutant périodiquement les phases d'élections de CHs et formation de clusters ce qui leur permet de s'adapter aux changements de la topologie, et donc offrir des chemins plus adaptatifs pour atteindre la station de base et acheminer les paquets de données.

Le délai moyen de bout en bout (DBBM)

Le délai moyen de bout en bout marqué par les trois protocoles LMEER-S, LMEER-D et LEACH en présence d'un pourcentage variant de défaillance et en fonction du nombre de nœuds déployés dans le réseau est présenté sur la figure 4.10 (a, b et c).

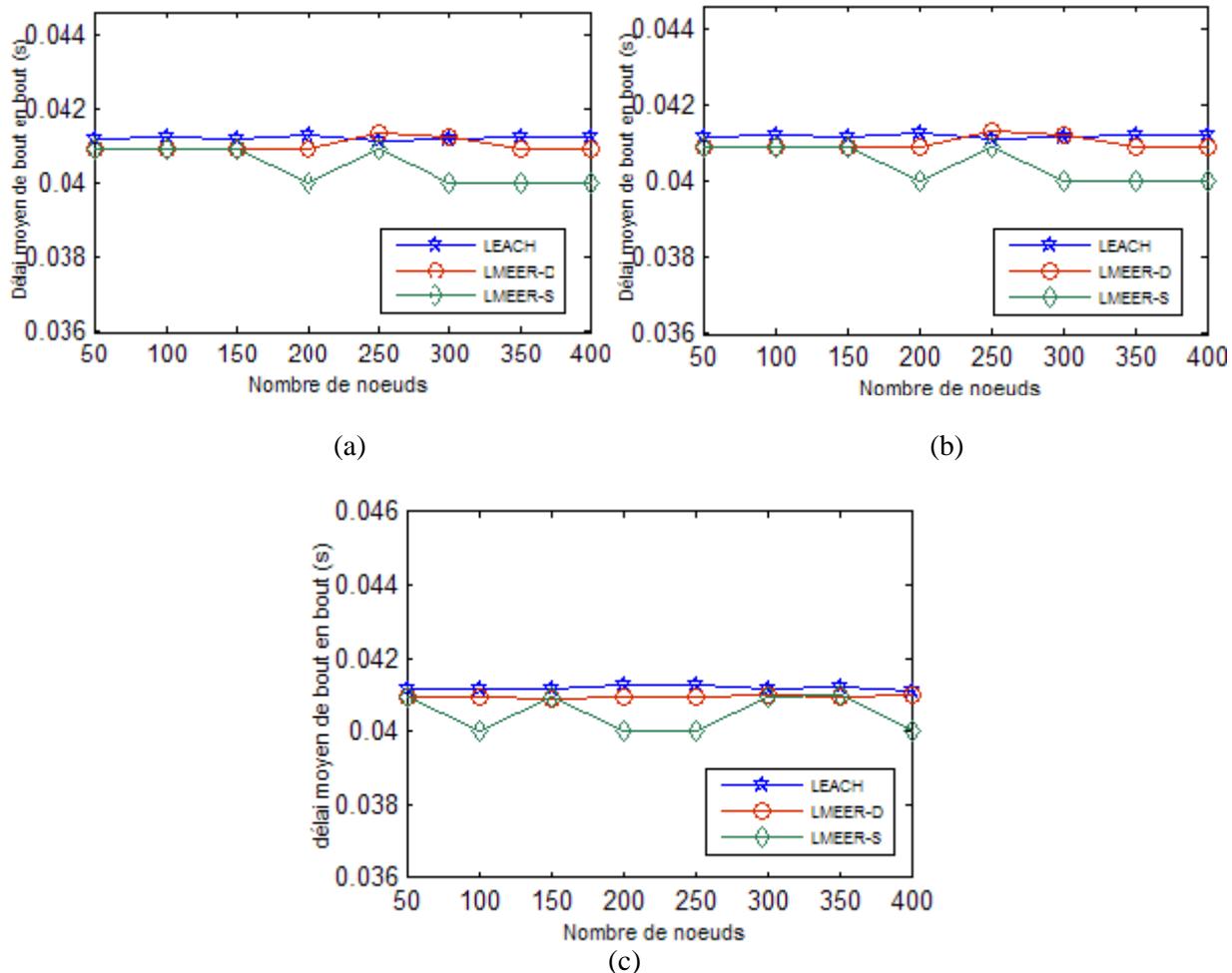


Fig.4.10 Délai moyen de bout en bout avec la présence d'un pourcentage de :
(a) 10% de nœuds défaillants (b) 30% de nœuds défaillants
(c) 50% de taux de nœuds défaillants.

Malgré que le pourcentage des nœuds défaillants dans le réseau a augmenté de 10% à 50%, on peut constater que cette augmentation n'a pas causé des changements considérables sur le délai moyen des trois protocoles LMEER-D, LMEER-S et LEACH, car ce délai varie toujours entre 0.04s et 0.041s comme dans la configuration sans défaillance (voir figure 4.7). Ces résultats sont très attendus et confirme ceux obtenus dans la figure 4.9, où la défaillance n'a pas influencé sur le taux moyen de livraison de paquets. Et comme le délai de bout en bout est inversement proportionnel au nombre de paquets reçus par la station de base alors l'avantage marqué par le protocole LMEER-S est très logique. En effet, dans le protocole LMEER-S le nombre de paquets reçus par la station de base est légèrement inférieur que celui du protocole LMEER-D et du protocole LEACH, et par conséquent son délai de bout en bout était presque le plus bas.

Durée de vie du réseau

Afin d'évaluer l'impact de la défaillance des nœuds sur la durée de vie du réseau nous avons exposé aléatoirement tout les 150 secondes, les 100 nœuds utilisés dans les scénarios de simulation à un pourcentage variant de défaillance, après 500 secondes d'exécution. Les figures suivantes montrent l'évolution de la durée de vie des 100 nœuds déployés dans le temps.

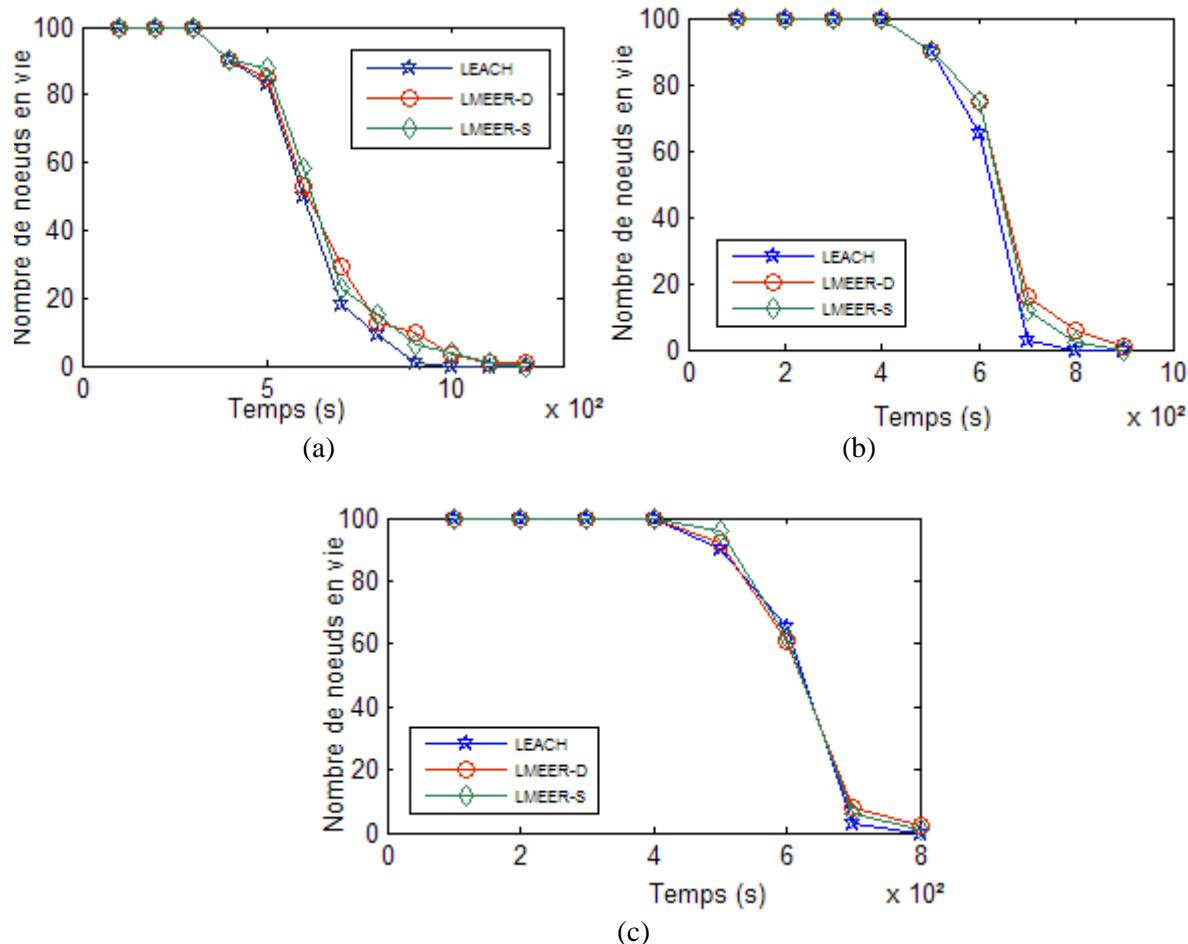


Fig.4.11 La durée de vie du réseau avec la présence de :
(a) 10% de taux de défaillance (b) 30% de taux de défaillance
(c) 50% taux de défaillance.

La première remarque qu'on peut tirer des figures (a), (b) et (c), c'est que l'augmentation du taux de défaillances dans le réseau a causé une dégradation dans sa durée de vie et ce, dans les trois protocoles LMEER-D, LMEER-S et LEACH. Cela est très attendu car l'augmentation du taux des nœuds défaillants a pour effet de diminuer le nombre de nœuds fonctionnels dans le réseau et par conséquent la durée de vie du réseau en entier. Néanmoins, la durée de vie du réseau avec les deux protocoles LMEER-S et LMEER-D était toujours meilleure que celle du protocole LEACH. Ceci est grâce à l'efficacité de la méthode d'équilibrage de charge utilisée dans ces deux protocoles, en permettant un contrôle sur la charge de chaque nœud CH lors du processus de rattachement à son cluster ; les nœuds non CH se rattachent au CH doté de plus d'énergie et ayant moins de nœuds membres ce qui augmente la durée de fonctionnement de ces nœuds CHs et par conséquent, prolonge la durée de vie du réseau en entier.

4.5 Conclusion

Lors de ce chapitre nous avons présenté une description détaillée du protocole LMEER , en exposant son principe de fonctionnement et ses mécanismes pour structurer le réseau en clusters et communiquer les données captées à la station de base. Dans notre protocole des techniques pour assurer une exploitation plus équitable de l'énergie des nœuds ont été adoptées. Cela est réalisé à travers une distribution de la charge des nœuds sélectionnés pour rôle de CHs, leur nombre de capteurs membres à gérer et paquets de données à router, tout en prenant en compte les facteurs influant sur la bonne gestion de la ressource énergétique à savoir leur niveau d'énergie résiduelle, leur degré et leur distance par rapport aux nœuds voisins et à la station de base.

L'étude comparative menée par simulation entre les performances du protocole LEACH et notre protocole LMEER à était réalisée sur deux tests distincts. Le premier de ces tests portait sur l'effet de la densité du réseau sur ses performances. Le second test analyse les performances du réseau dans la présence d'un pourcentage variant des nœuds défaillants. Les performances du réseau dans les deux tests de simulation ont été évaluées en fonction de la durée de vie, taux moyen de livraison de paquets, moyenne d'énergie consommée et le délai moyen de bout en bout.

Les résultats du premier test montrent que notre protocole améliore la dissipation d'énergie de 10% comparé à LEACH augmentant ainsi le gain d'énergie, et prolongeant la durée de vie du réseau. De même, les performances de notre protocole en termes de taux de livraison de paquets ne se dégradent pas avec la densification du réseau, ce qui est l'inverse du protocole LEACH.

Reste à noter que notre protocole utilisant la distance par rapport à la station de base pour configurer le réseau offre de meilleurs résultats que celui se basant sur le nombre de sauts car avec la première technique il y a moins de messages échangés lors de la phase de configuration du réseau et donc moins d'énergie consommée, ce qui explique ses résultats par rapport à l'autre variante.

Dans le deuxième test de simulation, notre protocole a prouvé son efficacité dans la conservation d'énergie car ses performances en termes de durée de vie et d'énergie consommée ne se dégradent pas en présence de la défaillance des nœuds. Ainsi, les résultats de simulation obtenus en termes de délai de bout en bout et de taux de livraison de paquets, montrent une tolérance aux défaillances qu'a présenté notre protocole comparé au protocole LEACH, ce qui le rend utilisable dans les environnements où la communication n'est pas fiable.

Conclusion Générale et Perspectives

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

Dans cette optique, le routage hiérarchique s'est présenté comme étant une solution prometteuse pour conserver l'énergie des nœuds, et faciliter la transmission des données capturées dans le réseau vers la station de base. Selon ce type de routage, les nœuds du réseau sont organisés en clusters gérés par un seul nœud (cluster-head). Ce cluster-head est non seulement responsable de la gestion des nœuds de son cluster, mais aussi l'assurance de la communication des données collectées localement à partir de ses noeuds membres vers la station de base. Cette structure hiérarchique permet d'alléger le trafic du réseau en effectuant des traitements au sein de chaque cluster, avant de faire parvenir les données à la station de base. L'obtention de meilleures performances pour les réseaux de capteurs avec l'approche de clustering résulte d'une organisation efficace de ses capteurs en clusters, tout en essayant de produire un nombre stable et équilibré.

Guidés par un seul objectif, réduire la consommation énergétique des noeuds capteurs pour augmenter la durée de vie du réseau tout en prenant en compte les limites rencontrés dans les solutions étudiées, nous avons conçu un nouveau protocole de routage selon l'approche du clustering. Ce protocole est nommé LMEER (*Layered Multi-hop Energy Efficient clustering Routing Protocol*), il prend en considération les contraintes imposées par les capteurs, basse capacité de stockage et durée de vie limitée, en permettant une gestion plus efficace de la ressource énergétique lors de la communication des données dans le réseau, à travers l'introduction d'une structure en couches pour la topologie du réseau. Cette configuration en couches offre une souplesse dans la communisation des données captées vers la station de base, en adoptant un routage multi-sauts réalisé entre les nœuds relais des différentes couches, ce qui permet de consommer moins d'énergie pour atteindre la station de base. Par ailleurs, notre protocole assure un équilibrage dans la charge des cluster-heads en termes du nombre de capteurs à gérer et nombre de paquets à router dans le réseau. La prise en compte des capacités énergétiques des nœuds cluster-heads pour grouper les noeuds capteurs en clusters lors du processus de rattachement permet d'optimiser leurs ressources énergétiques, et d'apporter un partitionnement de tailles inégales en nombre de capteurs par cluster.

L'évaluation des performances de notre protocole a été réalisée par simulation, en utilisant le simulateur open source J-Sim. Ce dernier contient un module spécifique pour les réseaux de capteurs sans fil. La série de simulation menée sur notre protocole nous a permis de prouver son efficacité dans la réduction de l'énergie consommée par les nœuds capteurs, conduisant ainsi à une prolongation dans leur durée de fonctionnement et étendant la durée de vie du réseau. Cette série de simulation a aussi montré l'efficacité de notre protocole dans les réseaux à forte densité et face à la présence de défaillance, en présentant des résultats très intéressant en terme de délai de bout en bout et taux de livraison de paquets à la station de base, mais aussi en termes de préservation de l'énergie des nœuds capteurs et la durée de vie du réseau.

Comme perspectives de notre travail, nous envisageons la modification du mécanisme utilisé lors de l'élection des noeuds relais assurant le routage multi-sauts des paquets de données à la station de base. Comme amélioration possible, on maintient plusieurs nœuds pour rôle du prochain nœud relais au niveau de chaque cluster-head avec l'application d'un choix probabiliste parmi l'ensemble maintenu, afin de permettre un fonctionnement plus robuste pour le réseau. La deuxième voie consiste à apporter une amélioration sur le mécanisme d'élection des nœuds cluster-heads, en permettant une sélection de nœuds backups (cluster-heads remplaçants ou secondaire). Ces derniers prendront le relais lors de l'épuisement de l'énergie des nœuds choisis comme cluster-heads primaires. Cette technique permettra de contrôler le fonctionnement cyclique en diminuant le déclenchement périodique du processus de clustering pour l'élection des cluster-heads et la formation des clusters.

Après avoir testé le réseau dans un environnement dense et homogènes, nous examinerons les résultats du protocole proposé, dans le cas d'un réseau avec des nœuds capteurs hétérogènes. Enfin une dernière perspective consiste à étudier notre protocole dans un environnement mobile. En effet, pour pouvoir prendre en charge la mobilité de la station de base ou des nœuds capteurs nous devrons modifier le fonctionnement de notre protocole de façon à l'adapter à cet environnement mobile.

Bibliographie

- [1] J.N. Al-Karaki et A.E. Kamal, “*Routing Techniques in Wireless Sensor Networks: a Survey*”, In IEEE Wireless Communications, Volume 11, Issue 6, On pages 6-28, ISSN: 1536-284, December 2004.
- [2] V. Raghunathan, C. Schurges, S. Park, and M..B. Srivastava, “*Energy-Aware Wireless Microsensor Networks*”, IEEE Signal Processing Magazine , 40-50, 2002.
- [3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci “*A survey on sensor networks:*”, IEEE Communications Magazine, Vol. 40, No. 8, pp. 102-116, August 2002.
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. I. Cayirci. “*Wireless sensor networks : A survey*”. IEEE Computer Networks, Vol. 38, No. 4, pp. 393-422, March 2002.
- [5] Holger Karl, Andreas Willig , “*A short survey of wireless sensor networks* “, Berlin, October 2003.
- [6] Pierre Parrend , “*Localisation dans les Réseaux de Capteurs* ” , INSA Lyon, janvier 2005.
- [7] Bhaskar Krishnamachari, ”*Networking Wireless Sensors*”, Cambridge University Press, 2005.
- [8] Ananthram Swami, Qing Zhao ,Yao-Win Hong ,Tsing Hua Lang Tong, ”*Wireless Sensor Networks Signal Processing and Communications Perspectives* ”, Willy edition, 2007.
- [9] Edgar Chavez , Nathalie Mitton, and Hector Tejeda ,“*Routing in Wireless Networks with Position Trees*”, ADHOC-NOW France, 2007.
- [10] E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, and V.Z. Groza,” *Sensor-based information Appliances*”, IEEE Instrumentation Measurement Magazine, vol.3, no.4, pp.31-35, December 2000.
- [11] Ludovic SAMPER, “*Modélisations et Analyses de Réseaux de Capteurs* ” ,Thèse de doctorat, institut national polytechnique, Grenoble, France, 7 avril 2008.
- [12] M. A. M .Vieira and al, “*Survey on wireless sensor network devices* ”, In 9th IEEE International Conference on Emerging Technologies and Factory Automation(ETFA’03), Lisbon, Portugal, septembre 2003.
- [13] K. Baumgartner, “*Réseaux de capteurs sans fil* ”, IBCOM, Décembre 2005.
- [14] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, Kristofer Pister ,“*System Architecture Directions for Networked* ”, ACM 0-89791-88-6/97/05, 2000.
- [15] Philip Levis, David Gay, and David Culler, Bridging the Gap, “*Programming Sensor Networks with Application Specific Virtual Machines* ”, OSDI , 2004.
- [16] R. Meraihi, "Gestion de la qualité de service et contrôle de topologie dans les réseaux ad hoc". Thèse de doctorat, Ecole nationale supérieure des télécommunications, TELECOM Paris, 2003.
- [17] François Ingelrest, ‘*Protocoles localisés de diffusion et économie d'énergie dans les réseaux ad hoc et de capteurs* ’, Thèse de doctorat (spécialité Informatique), le 30 juin 2006.
- [18] Guillermo BARRENETXA, ”*Distributed Routing Algorithms For Sensor Networks* “, école polytechnique fédérale de Lausanne, 2006.

- [19] Jon S. Wilson, “*Sensor Technology Handbook*”, Elsevier , 2005.
- [20] Eiko Yoneki, Jean Bacon ,”*A survey of Wireless Sensor Network technologies:research trends and middleware’s role* “,UCAM-CL-TR-646 , ISSN 1476-2986 university of Cambridge , September 2005.
- [21] I. KHELLADI, N. BADACHE, “*Les réseaux de capteurs: état de l’art, rapport de recherche* ”, laboratoire LSI USTHB, février 2004.
- [22] kemal Akkaya ,Mohamed Younis, ”*A survey on routing protocols for wireless sensor network* ”, Ad Hoc Networks, Vol. 3, No. 3, pp. 325-349, 2005.
- [23] Fei Hu, Neeraj K.Sharma, ”*Security consideration in ad hoc sensor networks* ”, Elsevier B.V, 26 November 2003.
- [24] Youssef, Mohamed; El-Sheimy, Naser; ”*Wireless Sensor Network: Research vs. Reality Design and Deployment Issues* ”, Communication Networks and Services Research CNSR '07, Mai 2007.
- [25] Holger Karl,Andreas Willig, ”*Protocols And Architectures For Wireless Sensor Networks* ”, John Wiley & Sons , 2005.
- [26] Mohammad Ilyas ,Imad Mahgoub, ”*Handbook Of Sensors Networks :Compact Wireless And Wired Sensing Systems* ” CRC press , 2005.
- [27] [En ligne] <http://french.bluetooth.com/> dernière consultation : le 28 janvier 2009.
- [28] [En ligne] <http://www.zigbee.org/> dernière consultation : le 28 janvier 2009.
- [29] E. Shih et al; Physical Layer; ”*Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks* ”, Proc. ACM MobiCom '01, pp. 272–86, Rome, Italy, July 2001.
- [30] C.Intanagonwiwat; R. Govindan; D. Estrin, Directed Diffusion,”*A Scalable and Robust Communication Paradigm for Sensor Networks* ”, Proc. ACM MobiCom'00, Boston, MA, pp. 56–67, 2000.
- [31] M. Younis and T. Nadeem,”*Energy efficient mac protocols for wireless sensor networks* ”, Technical report, University of Maryland Baltimore County,USA, 2004.
- [32] W. Ye , J. Heidemann, and D. Estrin”*An Energy-Efficient MAC Protocol for Wireless Sensor Networks* ”. In Proceedings of IEEE Infocom, pages 1567–76, New York, NY, 2002.
- [33] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. ”*Energy-efficient communication protocol for wireless sensor networks* ”. Proceedings of the IEEE Hawaii International Conference on System Sciences, pp. 3005-3014, January 2000.
- [34] G. J. Pottie and W. J. Kaiser ,”*Wireless Integrated Network Sensors* ” , Common. ACM, vol. 43, no. 5, pp.551-58, May 2000.
- [35] S. ZHUANG and C-G. CASSANDRAS,” *Optimal dynamic voltage scaling for wireless sensor networkswith real-time constraints* ”. In Proceedings of SPIE Conference on Intelligent Systems in Design and Manufacturing VI, pages 8–16, October 2005.
- [36] Mounir Achir , ”*Technologies basse consommation pour les réseaux Ad Hoc* ”. Thèse pour obtenir le grade de Docteur de l’INPG. Institut National Polytechnique de Grenoble, juillet 2005.

- [37] A. BACHIR ‘‘Allongement de la durée de vie des réseaux sans fil de capteurs par l’optimisation des protocoles de routage et d’accès au canal’’. Thèse de doctorat en informatique, Institut National Polytechnique, Grenoble, France, Janvier 2007.
- [38] K. Sohrabi and G.J. Pottie, ” *Performance of a novel self-organization protocol for wireless ad-hoc sensor networks*”. In Proceeding of the 50th IEEE Vehicular Technology Conference (VTC’99), vol.2, pp.1222-1226, Amsterdam, September 1999.
- [39] W. Ye , J. Heidemann, D. Estrin , ‘‘*An Energy-Efficient MAC Protocol for Wireless Sensor Networks*’’, IEEE Computer and Communications Societies (INFOCOM’02), vol.3, pp.1567-1576, New York, NY, USA, June 2002.
- [40] T. Van Dam and K. Langendoen , “*An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks*”, In proceedings of the 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys’03), pp.171-180, Los Angeles, California, USA, November 2003.
- [41] P. Lin, C. Qiao, and X.Wang, ” *Medium access control with a dynamic duty cycle for sensor networks*”. In IEEE Wireless Communications and Networking Conference (WCNC’04), vol.3, pp.1534-1539, March 2004.
- [42] D. J. Dechene, A. El Jardali, M. Uccini, A. Sauer, ” *A Survey of Clustering Algorithms for Wireless Sensor Networks*”, Project Report, Department of Electrical and Computer Engineering, The University of Western Ontario, Canada, December 2006-2007.
- [43] Ameer Ahmed Abbasi, Mohamed Younis “*A survey on clustering algorithms for wireless sensor networks*”, Computer. Communication Architecture, vol. 30,no. 14–15, pp. 2826–2841, 2007.
- [44] A. Delye, V. Gauthier, M. Marot, and M. Becker. "Etat de l'art sur les réseaux de capteurs", Rapport de recherche INT N-05001RST GET-INT, UMR5157 SAMOVAR, Institut National des Télécommunications, Evry, France, 2005.
- [45] M.younis ,G.Gupla “ *Load Balanced Clustering Of Wireless Sensor Networks* “, IEEE International Conference on Communication (ICC 2003), Anchorage, Alaska, Mai 2003.
- [46] Nauman Israr and Irfan Awan Energy Efficient Intra Cluster Head Communication Protocol (EICHCP) Department of Computing, University of Bradford 2006.
- [47] M.Battacharayya, A.Kumar “ *Design ad Analasys of energy reference metric in cluster based wireless sensor networks*”, 4th European Conference on Circuits and Systems for Communications, ECCSC, September 2008.
- [48] ACQUIRE N. Sadagopan, B. Krishnamachari, and A. Helmy. "The ACQUIRE mechanism for efficient querying in sensor networks". Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska, pp. 149-155, Mai 2003.
- [49] Raluca Marin-Perianu,’’ *Wireless Sensor Networks in Motion : Clustering Algorithms for Service Discovery and Provisioning* ’’ thèse de doctorat , University of Twente, November 2008.
- [50] G.Xin, W.Yangant, B.Degang , “*EEHCA: An Energy Effecient Hierarchical Clustering Algorithm For Wireless Sensor Networks*”,Information Technologie Journal, Asian Network For Scientific Information, 2008.
- [51] B. Liu, P. Brass, and O. Dousse, ” *Mobility Improves Coverage of Sensor Networks*”, MobiHoc’05, 25- 7, Mai 2005.

- [52] B. Krishnamachari, D. Estrin, S. Wicker, "Modeling Data Centric Routing in Wireless Sensor Networks," Proceedings of IEEE INFOCOM, New York, NY, June 2002.
- [53] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. "Energy An Application-Specific Protocol Architecture for Wireless Microsensor Networks ". Proceedings of the IEEE TRANSACTIONS On Wireless Communications, vol. 1, no. 4, October 2002.
- [54] G. Gupta, M. Younis, "Fault-Tolerant Clustering of Wireless Sensor Networks," Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003), Louisiana, Mars 2003.
- [55] E. Ilker Oyman and Cem Ersoy, "Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks", Proceedings of the IEEE International Conference on Communications (ICC 2004), Paris, June 2004.
- [56] DCA S. Basagni. "Distributed clustering for ad hoc networks". Proceedings of Fourth International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN'99), pp.310-315, Perth/Fremantle, WA, Australia, June 1999.
- [57] V. Loscri, G. Morabito, and S. Marano, "A Two-Level Hierarchy for Low-Energy Adaptive Clustering Hierarchy", DEIS Department, University of Calabria. IEEE, 2005.
- [58] Ossama Younis , Sonia Fahmy " HEED: A Hybrid, Energy-Efficient,Distributed Clustering Approach for Ad Hoc Sensor Networks for Networked Sensors", IEEE Transactions on mobile computing, vol. 3, no. 4, October-December 2004.
- [59] M. Ye, C. Li, G. Chen, and J. Wu, "EECS: An Energy Efficient ClusteringScheme in Wireless Sensor Networks", National Laboratory of NovelSoftware Technology,Nanjing University,China 2002.
- [60] C.Li, M.Ye, G.Chen, J.Wu "An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks, IEEE International Conference on Mobile Adhoc and Sensor Systems, December 2005.
- [61] M. Demirbas, A. Arora and V. Mittal, "FLOC: A fast local clustering service for wireless sensor networks", Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'04), Palazzo dei Congressi, Florence, Italy, June 2004.
- [62] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Francisco, California, April 2003.
- [63] K. Xu and M. Gerla, "A Heterogeneous Routing Protocol Based on a New Stable Clustering Scheme", in the Proceeding of IEEE Military Communications Conference (MILCOM 2002), Anaheim, CA, October 2002.
- [64] R. Nagpal and D. Coore, "An algorithm for group formation in an amorphous computer", Proceedings of the 10th International Conference on Parallel and Distributed Systems (PDCS'98), Las Vegas, NV, October 1998..
- [65] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks," Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01), Anchorage, AK, April 2001.
- [66] P. Tsigas, "Project on Moblie Ad Hoc Networking and Clustering". For the Course EDA390 Computer Communcation and Distributed Systems, Manual for University Course.

- [67] D. J. Baker ,A. Epheremides, “*The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm*”, IEEE Transactions on Communications, vol.Com-29, no11, Novembre 1981.
- [68] S. Lindsey and C.S. Raghavendra, “*PEGASIS :Power-Efficient Gathering in Sensor Information Systems*”, IEEE Transaction on Parallel and Distributed Systems, vol.13, no.9,pp.924-935, September 2002.
- [69] Amis, R. Prakash, T. Vuong, and D. Huynh, “*Max-Min D-Cluster Formation in Wireless Ad Hoc Networks,*” IEEE INFOCOM, Mach 2000.
- [70] M. Chatterjee, S. K. Das, and D. Turgut, “*WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks,*” Clustering Computing, vol. 5, pp. 193–204, 2002.
- [71] H. Chan and A. Perrig, “*ACE: An Emergent Algorithm for Highly Uniform Cluster Formation,*” Proceedings of the 1st European Workshop on Sensor Networks (EWSN), Berlin, Germany, January 2004.
- [72] Y. Yao and J. Gehrke. “*The cougar approach to in-network query processing in sensor networks*”. ACM SIGMOD Record, Vol. 31, No. 3, pp. 9-18, September 2002.
- [73] S. Basagni, “*Distributed clustering for ad hoc networks*”, Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN), pages 310–315, USA, 1999.
- [74] B. KRISHNAMACHARI, D. ESTRIN, and S.WICKER.” *Modelling data-centric routing in wireless sensor networks*”, Proceedings of the IEEE INFOCOM, 2002.
- [75] C. INTANAGONWIWAT, R. GOVINDAN, and D.estrin,” *Directed diffusion : a scalable and robust communication paradigm for sensor networks*”, Proceedings of the ACM/IEEE international conference on Mobile computing and networking, pages 55–67, Boston, USA, Août 2000.
- [76] C. D. Braginsky and D. Estrin. “*Rumor routing algorithm for sensor networks*”. Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), pp. 22-31, ACM Press New York, NY, USA, 2002.
- [77] Y. Wang, T .L.X. Yang and D. Zhang “ *PEBECs: An Energy Efficient and Balance Hierarchical Unequal Clustering Algorithm for Large Scale Sensor Networks*”, Information Technology Journal vol. 8. page no.: 28-38, Asian Network For Scientific Information, 2009.
- [78] B. KRISHNAMACHARI, D. ESTRIN, and S.WICKER,” *Modelling data-centric routing in wireless sensor networks*”. Proceedings of the IEEE INFOCOM, 2002
- [79] R. Shah and J. M. Rabaey. “*Energy aware routing for low energy ad-hoc sensor networks*”. Wireless Communications and Networking Conference (WCNC), Vol. 1, 2002.
- [80] SAR K. SOHRABI, J. GAO, V. AILAWADHI, and G-J. POTTIE.” *Protocols for self-organization of a wireless sensor network.*” IEEE Personal Communications, 7(5) :16–27, 2000.
- [81] T. HE, J-A. STANKOVIC, C. LU, and T. ABDELZAHER.” *Speed : A stateless protocol for real time communication in sensor networks*”. In Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS), Mai 2003.
- [82] W-R. HEINZELMAN, J. KULIK, and H. BALAKRISHNAN. “*Negotiation based protocols for disseminating information in wireless sensor networks*”. Wireless Networks, 8(2-3) :169–185, 2002.

- [83] Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Yang Tyan, Honghai Zhang. “*J-Sim: A Simulation and Emulation Environment for Wireless Sensor Networks*”. IEEE Wireless Communications magazine . 2005, Vol. 13, p, 2006.
- [84] The network simulator, NS version 2, [En ligne] : <http://www.isi.edu/nsnam/ns/>.
- [85] Levis, P. et N. Lee. “*TOSSIM: A Simulator for TinyOS Networks*”, 2003.
- [86] The OMNeT++ Discrete Event Simulator System, [En ligne], <http://www.omnetpp.org/>.
- [87] Global Mobile Information Systems Simulation library : Glomosim, [En ligne] : <http://pcl.cs.ucla.edu/projects/glomosim/>.
- [88] J-Sim Home page. [En ligne]: <http://sites.google.com/site/jsimofficial/>.