

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**  
**MINISTERE DE L'ENSEIGNEMENT SUPERIEURE**  
**ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE LARBI BEN M'HIDI OUM EL BOUAGHI**  
**FACULTE DE SCIENCES EXACTES, DE NATURE ET DE LA VIE**  
**DEPARTEMENT DE MATHÉMATIQUE ET D'INFORMATIQUE**  
Option : Ingénierie des systèmes distribués

**THESE**

Pour l'Obtention du diplôme de:

**DOCTORAT EN INFORMATIQUE**

**THEME :**

**Algorithme de routage pour les réseaux de  
capteurs avec prise en charge de la  
consommation d'énergie**

**Présenté par :**

Mlle. Gherbi Chirihane

**Rapporteur :** Pr. Mohamed Benmohammed (Université de Constantine)

**Co-encadreur :** Dr Zibouda Aliouat (Université de Sétif)

**Président de Jury :** Dr Nini Brahim (Université de Oum el-Bouaghi)

**Examineurs :** Pr. Bilami Azeddine (Université de Batna)

Pr. Chikhi Salim (Université de Constantine)

Dr Boutekkouk Fateh (Université de Oum el-Bouaghi)

*Année Universitaire 2016 – 2017*

## Dédicace

*A mon très cher père*

*A ma très chère mère*

*A mon très cher frère*

*A toute ma famille en Syrie*

*et en Algérie*

*Je dédie ce travail à mes parents. Aucun mot n'est assez fort pour leur exprimer la reconnaissance sincère que je leur porte pour la richesse de leurs enseignements.*

{ وَأَخْفِضْ لَهُمَا جَنَاحَ الدَّلِّ مِنَ الرَّحْمَةِ وَقُلْ رَبِّ اَرْحَمُهُمَا كَمَا رَبَّيْتَنِي صَغِيرًا }

صدق الله العظيم- الإسراء (24)

*Merci du fond du cœur*

# Remerciement

☺ *J'exprime ma profonde gratitude à mon directeur de thèse Pr. Mohammed Benmohammed pour avoir dirigé mes travaux de thèse. Il a pris le temps de me guider et de me donner des conseils de mes travaux et pour sa disponibilité, son écoute, son expérience, sa sympathie et ses encouragements qui m'ont permis de mener à bien cette thèse, et je tiens à remercier aussi les membres de l'équipe de laboratoire « LIRE » à Constantine.*

☺ *Je suis très reconnaissante envers Dr Aliouat Zibouda, qui n'a pas compté les heures pour me guider, je tiens à exprimer mes plus vifs remerciements d'une part, pour avoir accepté de diriger ce travail de thèse, et d'autre part pour les connaissances qu'elle m'a transmises, ses conseils, ses remarques, ses encouragements et son soutien (professionnel et personnel) ont été d'une importance cruciale dans l'aboutissement de ce projet. Merci infiniment Madame pour ta patience, et pour tout le temps que tu m'as consacré durant ces cinq années.*

☺ *Un grand merci à mon professeur Pr. Aliouat Makhlouf d'avoir accepté de faire une relecture de mon article scientifique, merci également pour ses remarques intéressantes et pour son soutien qui m'a permis de mener à terme mon travail de recherche.*

☺ *J'adresse également mes très sincères remerciements à Dr Nini Brahim, d'avoir accepté de présider mon jury.*

☺ *J'exprime ma plus profonde gratitude à Pr. Bilami Azeddine, Dr Boutekkouk Fateh et Pr Chikhi Salim de m'avoir fait l'honneur de rapporter cette thèse, soient assurés de ma reconnaissance pour l'intérêt qu'ils ont apporté à ce travail.*

☺ *Un merci particulier à Pr. Mokhati Farid pour ses conseils judicieux et son aide précieuse.*

☺ *Merci à mes amis, mes collègues et toute personne qui a contribué à la réalisation de ce travail.*

☺ *Enfin, sur un plan plus personnel, je veux remercier les personnes sans qui ce travail n'aurait jamais vu le jour. Je ne trouverais sans doute les mots pour remercier assez les personnes qui me sont les plus chères : mon père et ma mère pour leurs sacrifices, leur patience et tout ce qu'ils ont fait pour m'apporter le bonheur. Qu'ils sachent à travers ces quelques mots combien je leur suis reconnaissant et combien je sais tout ce que je leur dois, je vous serai reconnaissante à vie. Une mention très particulière à toute ma famille en Syrie qui m'a toujours soutenu, je vous aime infiniment.*

---

---

## *Abstract*

---

---

Clustering is a well-known approach to cope with large nodes density and efficiently conserving energy in Wireless Sensor Networks (WSN). Load balancing is an effective approach for optimizing resources like channel bandwidth, the main objective of this paper is to combine these two valuable approaches in order to significantly improve the main WSN service such as information routing. So, our proposal is a routing protocol in which load traffic is shared among cluster members in order to reduce the dropping probability due to queue overflow at some nodes. To this end, a novel hierarchical approach, called Hierarchical Energy-Balancing Multipath routing protocol for Wireless Sensor Networks (HEBM) is proposed. The HEBM approach aims to fulfill the following purposes: decreasing the overall network energy consumption, balancing the energy dissipation among the sensor nodes and as direct consequence: extending the lifetime of the network. In fact, the cluster-heads are optimally determined and suitably distributed over the area of interest allowing the member nodes reaching them with adequate energy dissipation and appropriate load balancing utilization. In addition, nodes radio are turned off for fixed time duration according to sleeping control rules optimizing so their energy consumption. The performance evaluation of the proposed protocol is carried out through the well-known NS2 simulator and the exhibited results are convincing. Like this, the residual energy of sensor nodes was measured every 20 seconds throughout the duration of simulation, in order to calculate the total number of alive nodes. Based on the simulation results, we concluded that our proposed HEBM protocol increases the profit of energy, and prolongs the network lifetime duration from 32% to 40% compared to DEEAC reference protocol and from 25% to 28% compared to FEMCHRP protocol. The authors also note that the proposed protocol is 41.7% better than DEEAC with respect to FND (First node die), and 25.5% better than FEMCHRP with respect to LND (last node die) while maintaining the average data transmission delay. We found also that HEBM achieved 66.5% and 40.6% more rounds than DEEAC and FEMCHRP respectively.

**Keywords:** Energy Efficiency, Life Time Duration, Cluster-heads Distribution, Load Balancing, Hierarchical clustering, Wireless Sensor Networks.

---

---

## Résumé

---

---

La distribution des capteurs sans fil dans une zone géographique a permis d'optimiser les applications de surveillance. La conception de ce type d'architecture est soumise à certaines contraintes. Nous nous sommes intéressés aux paramètres suivants : l'énergie, la localisation, le facteur d'échelle, les ressources limitées, la bande passante limitée, l'équilibrage de la charge et la topologie dynamique. Nous avons proposé un protocole de routage (HEBM : Hierarchical Energy balancing multipath routing protocol). Notre intérêt s'est porté sur un mécanisme d'équilibrage de charge grâce à la réalisation d'une bonne distribution des cluster-heads dans le modèle de réseaux et pour améliorer les performances en termes de durée par la réduction du signal de transmission et les messages de contrôle. En fait, dans le cas où un ensemble de capteurs se situe dans la même zone géographique et doit capter la même information, tous ces capteurs vont par la suite communiquer cette information à la station de base. De ce fait, une consommation inutile de l'énergie aura lieu (malgré l'utilisation des mécanismes d'agrégation dans le but d'éliminer les données redondantes) et qui par conséquent, dégrade les performances du réseau et diminue la durée de vie des capteurs et du réseau tout entier. Dans ce cas, un mécanisme d'équilibrage de la charge et d'énergie devient indispensable afin d'équilibrer la consommation de l'énergie.

L'approche proposée vise à remplir les objectifs suivants : Diminution de la consommation globale d'énergie en équilibrant la dissipation d'énergie entre les nœuds. Cela a eu comme conséquence directe l'extension de la durée de vie du réseau. L'équilibrage de la charge des clusters doit être bien fait par l'utilisation de deux contributions : Le cluster-head temporaire et le cluster-head final. Les Cluster-heads sont parfaitement déterminés par une formule mathématique proposée, et sont convenablement répartis sur la zone d'intérêt. Le Clustering proposé prend en compte la complexité du message et le temps. Les cluster-heads et les nœuds correspondants doivent prendre en compte les métriques suivantes : diminuer la latence, la fiabilité des données reçues à la station de base. L'évaluation de performances du protocole proposé montre qu'il assure une réduction importante du nombre de clusters solitaires et un gain énergétique considérable.

**Mots clés:** Réseaux de capteurs, Clustering, Equilibrage de charge, Optimisation de ressources (énergie, mémoire et processus), routage hiérarchique.

## الملخص

تعتبر شبكات الاستشعار اللاسلكية تكنولوجيا ناشئة تهدف إلى توفير القدرات الابتكارية. وتشهد هذه التكنولوجيا تطورا سريعا بالإضافة إلى استخدامها في العديد من المجالات العلمية، اللوجستية، العسكرية الصحة. غير أن الموارد المحدودة لهذا النوع من الشبكات تشكل عقبة رئيسية، خاصة في مجال الأمن و الطاقة. ان التوزيع الأمثل لأجهزة الاستشعار اللاسلكية في منطقة جغرافية لها اهمية بالغة فيما يخص تطبيقات المراقبة، رغم أن هذا التصميم في هذا النوع من الشبكات يخضع لقيود معينة. ونحن مهتمون في تحسين شبكات الاستشعار باستخدام العوامل التالية: الطاقة، والموقع، وعامل حجم الشبكات، ومحدودية الموارد، موازنة تكثف العقد. لذلك اقترحنا بروتوكول توجيه HEBM الهرمي ويتركز اهتمامنا في هذه المساهمة على آلية موازنة تكثف العقد من خلال تحقيق توزيع جيد للعقد الرئيسية لمجموعات التكتل في نموذج الشبكة وتحسين الأداء من حيث المدة عن طريق الحد من انتقال إشارة و رسائل التحكم، في غالب الأحيان تقع مجموعة من أجهزة الاستشعار في نفس المنطقة الجغرافية، وتحتاج إلى التقاط نفس المعلومات فإن جميع أجهزة الاستشعار تنقل هذه المعلومات إلى المحطة الأساسية. ولذلك يحدث استهلاك غير مجدي للطاقة (على الرغم من استخدام آليات تجميع من أجل القضاء على البيانات المتكررة) وبالتالي في هذه الحالة يتحلل أداء الشبكة ويقلل من حياة أجهزة الاستشعار والشبكة بالكامل. وبالتالي آلية التوازن تصبح ضرورية من أجل تحقيق التوازن في استهلاك الطاقة.

ويهدف المنهج المقترح لتحقيق الأهداف التالية: الحد من الاستهلاك الكلي للطاقة في الشبكة، وتحقيق التوازن في تبديد الطاقة بين عقد الاستشعار وكننتيجة مباشرة لتمديد عمر الشبكة، موازنة عدد عقد المجموعات ويجب أن يتم أيضا عن طريق استخدام مساهمتين: العقد الرئيسية المؤقتة والعقد الرئيسية النهائية، يتم تحديد العقد الرئيسية بدقة وفق برهان رياضي مقترح ودقيق، يبقى فعال رغم تعقيد المعلومات والوقت. يتم تحديد واختيار العقد الرئيسية النهائية الخاصة لمجموعات التكتل وفق 4 مقاييس مقترحة. يتم بعدها اثبات فعالية البروتوكول المقترح بمجموعة من القياسات والنتائج ضمن برنامج محاكاة NS2.

**كلمات البحث:** شبكات الاستشعار، وتحسين الموارد (الطاقة، الذاكرة والعمليات)، والتوجيه الهرمي.

# Table des matières

## I. Etat de l'art

### Chapitre 1

#### Concepts Généraux Sur Les Réseaux De Capteurs Sans Fils

1. Introduction .....	1
2. Les réseaux sans fils .....	1
2.1. Les réseaux ad hoc .....	2
2.2. Les réseaux de capteurs sans fils (RCSFs) .....	2
2.2.1 Les capteurs .....	3
2.2.2 Classification des capteurs .....	5
2.2.3. Caractéristiques d'un capteur .....	6
2.2.4. Architecture d'un nœud capteur .....	6
i. Sous-système d'acquisition .....	7
ii. Sous-système de traitement .....	8
iii. Sous-système de transmission .....	8
iv. Sous-système d'alimentation .....	9
2.2.5 Les plates-formes existantes sans fils .....	10
3. Qu'est-ce qu'un réseau de capteurs ? .....	12
a) Le capteur (sensor) .....	12
b) L'agrégateur (aggregator) .....	13
c) Le puits (sink) .....	13
d) La passerelle (gateway) .....	14
3.1. Le réseau des capteurs mobiles .....	14
3.2. Les états opérationnels d'un nœud capteur .....	14
3.3. Les principales caractéristiques et contraintes des RCSFs .....	15
4. Les types de communication .....	17
5. Rôle de la pile de communication .....	17
6. Scénarios d'utilisation des réseaux de capteurs sans fils .....	20
• L'industrie .....	24

# Table des matières

• Contrôle et maintenances dans les équipements industriels.....	24
• La sécurité industrielle .....	24
• Les moyens de transport.....	24
• Maison bureau et vie civile .....	24
• Gestion et contrôle de l'inventaire .....	25
• Surveillance et dépistage des véhicules .....	25
7. La gestion des RCSFs.....	25
8. Critères de conception de systèmes de gestion de RCSFs.....	26
8.1 Les fonctions principales de gestion des RCSFs.....	27
9. Conclusion.....	29

## Chapitre 2

### *Les Techniques De Conservation D'énergie Dans Les Réseaux De Capteurs Sans Fil*

1. Introduction .....	30
2. Facteurs intervenants dans la consommation d'énergie.....	30
3. Les méthodes de conservation d'énergie dans les réseaux de capteurs sans fils.....	34
3.1. Les méthodes proposées pour optimiser l'énergie de capture .....	35
1) Echantillonnage adaptatif .....	35
2) Echantillonnage hiérarchique .....	36
3) l'échantillonnage actif.....	36
3.2. La méthode proposée pour optimiser l'énergie de calcul.....	36
a).1 Techniques d'adaptation dynamique de la vitesse et de la tension du processeur .....	37
a).2 Techniques de partitionnement du système .....	38
3.3. Méthodes proposées pour minimiser l'énergie de communication .....	39
a) Agrégation de données .....	39
b) Implosion .....	39
c) Chevauchement (Overlap).....	39
d) L'ignorance des ressources (Resource blindness) .....	39
e) Le contrôle de topologie .....	40
f) Le contrôle de puissance.....	40
g) Les méthodes du groupement (clustering).....	40

# Table des matières

h) L'élection du cluster-head .....	40
i) Communication intra-cluster et inter-cluster .....	40
j) La maintenance des clusters .....	41
k) Les régimes de rendez-vous programmés (Scheduled rendez-vous).....	41
• S-MAC Le protocole S-MAC (Sensor-MAC).....	41
• T-MAC Dans le protocole T-MAC (Timeout-MAC).....	42
• D-MAC D-MAC (Data gathering MAC) .....	42
L) Les régimes asynchrones Un protocole sleep/wakeup asynchrone .....	43
4) Taxonomie des protocoles de routage.....	43
4.1 Classification selon les paradigmes de communication .....	43
4.1.1. Centré-nœuds .....	43
4.1.2. Centré-données .....	44
4.1.3. Basé-localisation .....	44
4.2. Classification selon la topologie du réseau .....	45
4.2.1. Topologie plate.....	45
4.2.2. Topologie hiérarchique.....	46
a) Définitions .....	47
Qu'est-ce que le clustering?.....	47
b) Communication des clusters .....	48
1) Communications intra-cluster.....	48
- Communications multi-sauts .....	48
2) Communications inter-cluster.....	48
- Transmissions à un seul saut.....	48
- Routage multi-sauts .....	48
4.3 Classification selon la méthode d'établissement de routes .....	49
4.3.1 Protocoles proactifs .....	49
4.3.2 Protocoles réactifs .....	49
4.3.3 Protocoles hybrides .....	50
4.4 Classification selon le mode de fonctionnement du protocole.....	50
4.4.1. Routage multi-chemins (Multipath) .....	50

# Table des matières

4.4.2.Routage basé sur les requêtes .....	50
4.4.3 Routage basé sur la négociation .....	50
4.4.4.Routage basé-QoS .....	51
5.Conclusion.....	51

## Chapitre 3

### *Les réseaux hiérarchiques et clustering « travaux antérieurs »*

1. Introduction .....	52
2. Facteurs de conception de protocoles de routage .....	52
2.1.Le déploiement des nœuds .....	53
2.2 Le modèle de livraison de données .....	53
2.3 La tolérance aux pannes .....	53
2.4 L'hétérogénéité des nœuds.....	53
2.5 La consommation d'énergie .....	54
2.6 La scalabilité .....	54
2.7 La connectivité .....	54
2.8 La mobilité .....	55
2.9 L'agrégation des données.....	55
2.10 La qualité de service.....	55
3.Classification des algorithmes de clustering .....	55
3.1.Modèle de réseau.....	55
3.1.1.Dynamique d'un réseau .....	56
3.1.2 Traitement des données dans le réseau.....	56
3.1.3.Déploiement et capacités des nœuds.....	58
3.2.Objectifs du clustering.....	58
3.2.1.Equilibrage de la charge .....	58
3.2.2. Tolérance aux fautes.....	58
3.2.3.Connectivité et délai réduit.....	59
3.2.4.Nombre minimal de cluster .....	59
3.2.5.Longévité maximal du réseau.....	59
3.3.Taxonomie des attributs de clustering .....	60
3.3.1.Propriétés de cluster .....	60

# Table des matières

3.3.2.Possibilités de cluster-head .....	60
3.3.3.Processus de clustering.....	61
4.Les protocoles de routage hiérarchiques proposés pour minimiser l'énergie.....	63
4.1.Energy-aware routing in cluster-based sensor networks .....	63
4.2.Algorithme FCM (Fuzzy C-Means) .....	64
4.3.HEER: A Delay-aware and Energy Efficient Routing Protocol for WSN.....	67
4.4 ICP: Instantaneous clustering protocol for wireless sensor networks .....	68
4.5Clustering algorithms for maximizing the lifetime of wsn with energy-harvesting .....	69
4.6 An Energy Aware Distributed Clustering Algorithm Using Fuzzy Logic for WSN with Non-uniform Node Distribution .....	70
4.7 FEMCHRP : Fuzzy Based Energy Efficient Multiple Cluster Head Selection Routing Protocol for Wireless Sensor Networks .....	71
4.8.Adaptive energy aware cluster-based routing protocol for WSN.....	72
4.9 SHaRP: A Hierarchical and Role Based Secure Routing Protocol for MWSN .....	73
4.10 GASONeC:A Genetic Algorithm-Based,Dynamic Clustering Method Towards Improved WSN Longevity.....	74
4.11 MRRCE: Multi-hop Routing Algorithm Using Steiner Points for Reducing Energy Consumption in Wireless Sensor Networks .....	75
4.12 Effect of multi-path fading model on T-ANT clustering protocol for WSN.....	76
4.13 MECPC: Mobility- and Energy-Conscious Clustering Protocol for wsn .....	76
4.14 GFTCRA :A grid based clustering and routing algorithm for solving hot spot problem in wireless sensor networks .....	77
4.15 An Energy Efficient Cluster Head Selection Technique Using Network Trust and Swarm Intelligence.....	78
4.16 H-CERP: Energy Efficient Scheme for Industrial WSN Applications .....	79
4.17 DDUCA: An Energy-Efficient Unequal Clustering Algorithm Using 'Sierpinski Triangle' for WSNs.....	80
4.18 EE-CDRDG: An Eneyg Efficient Clustering with Delay Reduction in Data Gathering Using Mobile Sensor Node.....	81
4.19 Energy Efficient Architecture for Intra and Inter Cluster Communication for Underwater Wireless Sensor Networks.....	83
5.Avantages et Inconvénients des protocoles proposés.....	84
6.Conclusion.....	85

# Table des matières

## II. La contribution

### Chapitre 4:

### *Proposition du Protocole HEBM*

1. Introduction .....	87
2. Motivation .....	88
3. Présentation de HEBM .....	88
3.1. Principe de HEBM .....	88
3.2. Le modèle de réseau du protocole HEBM .....	89
3.3 Le modèle radio .....	90
4. Phases de HEBM .....	91
4.1. La phase d'initialisation du HEBM .....	92
4.2 Phase de découverte des voisins .....	94
4.3 La phase d'élection de CH temporaire .....	97
4.4 La phase d'élection de cluster-head final .....	100
4.5 La phase de formation des clusters .....	102
4.6 Phase d'ordonnancement .....	105
5. phase de transmission .....	106
5.1 Les communications inter-cluster .....	106
5.2 Communications intra-cluster .....	108
5.3 Le Duty-Cycle synchrone .....	109
5.3.1 Exemple de Synchronisation .....	110
6. Discussions sur l'efficacité du protocole HEBM .....	112
7. Conclusion .....	113

# Table des matières

## Chapitre 5

### Simulation et performance de protocole HEBM

1. Introduction .....	114
2. Métriques d'évaluation de notre contribution .....	114
3. Environnement de simulation.....	115
3.1 Objectifs de simulation.....	115
3.2 Choix de simulateurs.....	115
3.3 Présentation du simulateur NS2 .....	116
3.4 Comment peut-on simuler le protocole HEBM dans NS2 ? .....	116
3.5 Avantages, limites et difficultés de NS2 .....	117
4. Evaluation des Performances .....	117
4.1 Energie résiduelle .....	118
4.2 L'énergie consommée .....	119
4.3 Durée de vie de HEBM .....	119
- Pourcentage des noeuds vivants pour chaque protocole. ....	120
- Instant de déconnexion du 1 <sup>er</sup> et du dernier nœud.....	122
- La relation entre la position de la BS et la durée de vie de réseau.....	123
- Le nombre des clusters par rapport au nombre des de noeuds.....	124
- Le nombre de cluster par rapport la portée de transmission .....	125
- La distribution des cluster-head de protocole HEBM.....	126
- La latence moyenne des paquets par rapport au nombre des noeuds.....	126
- Le Nombre des paquets transmis avec différentes tailles au fil de temps .....	127
- Le nombre de paquets délivrés à la BS .....	128
Conclusion.....	130
Conclusion générale .....	131
Nos contributions .....	133
Références bibliographiques .....	135
Annexe 1 : Méthode d'intégration d'un protocole sous NS2	
Annexe 2 : Code d'implémentation de 07 phases du HEBM	

# Liste des figures

## Chapitre 1

<b>Figure 1.1</b> Réseau Ad hoc .....	2
<b>Figure 1.2</b> Le fonctionnement d'un capteur.....	4
<b>Figure 1.3</b> Principe de l'étalonnage .....	4
<b>Figure 1.4</b> Schéma de principe d'un capteur.....	5
<b>Figure 1.5</b> Architecture des différents types de noeuds: capteur, robot, puits, passerelle.....	7
<b>Figure 1.6</b> Quelques modèles de capteurs sans fil.....	10
<b>Figure 1.7</b> Quelques cartes De captage. ....	11
<b>Figure 1.8</b> L'architecture habituelle des réseaux de capteurs sans fils.....	12
<b>Figure 1.9</b> Les états possibles d'un nœud capteur.....	15
<b>Figure 1.10</b> Pile de communication d'un capteur.....	19
<b>Figure 1.11</b> Scénarios d'utilisation des RCSFs .....	20

## Chapitre 2

<b>Figure 2.1</b> L'écoute abusive dans une transmission.....	33
<b>Figure 2.2</b> Les méthodes de conservation d'énergie dans les réseaux de capteurs sans fil....	35
<b>Figure 2.3</b> Taxonomie des protocoles de routage selon les paradigmes de communication..	44
<b>Figure 2.4</b> Classification selon la topologie du réseau .....	45
<b>Figure 2.5</b> Routage plat .....	46
<b>Figure 2.6</b> Routage hiérarchique .....	46
<b>Figure 2.7</b> Classification selon la méthode d'établissement de routes.....	49
<b>Figure 2.8</b> Classification selon le mode de fonctionnement du protocole .....	50

# Liste des figures

## Chapitre 3

<b>Figure 3.1</b> Topologie hiérarchiques .....	58
<b>Figure 3.2</b> Taxonomie des attributs de clustering .....	62
<b>Figure 3.3</b> Cluster typique dans un réseau de capteurs .....	64
<b>Figure 3.4</b> Les phases de protocole HEER .....	68
<b>Figure 3.5</b> Un exemple de résultat d' ICP .....	69
<b>Figure 3.6</b> Le model de réseau d'algorithmme de clustering .....	69
<b>Figure 3.7</b> La structure de Fuzzy System .....	70
<b>Figure 3.8</b> Le modèle de réseau du protocole FEMCHRP .....	71
<b>Figure 3.9</b> Organisation en cluster avec chaines .....	72
<b>Figure 3.10</b> La méthode de clustering dans SHaRP .....	73
<b>Figure 3.11</b> Le modèle de protocole GASONeC .....	74
<b>Figure 3.12</b> Processus de sélection des points Steiner .....	75
<b>Figure 3.13</b> Construction des grids GFTCRA .....	78
<b>Figure 3.14</b> Structure de Clustering avec gateway .....	79
<b>Figure 3.15</b> Le modèle de réseau de protocole H-CERP .....	80
<b>Figure 3.16</b> La phase finale de protocole DDUCA .....	81
<b>Figure 3.17</b> Schéma de structure du protocole EE-CDRDG .....	82
<b>Figure 3.18</b> Architecture de communication de cluster intra et inter pour UWSN .....	83

## Chapitre 4

<b>Figure 4.1</b> Le modèle de réseau .....	90
<b>Figure 4.2</b> Les phases de HEBM .....	92
<b>Figure 4.3</b> La distance $D(N_i, B_s)$ .....	93
<b>Figure 4.4</b> Dérivation de la distance à partir de l'instant d'arrivée du signal .....	93
<b>Figure 4.5</b> Pseudo Algorithmme de la phase d'initialisation du protocole HEBM .....	95
<b>Figure 4.6</b> Découverte du voisin (Node Weight) .....	96

## Liste des figures

<b>Figure 4.7</b>	La création de table des voisins et le calcul de ‘node-weight’ .....	96
<b>Figure 4.8</b>	Election du clusterhead temporaire $Ch_{temp}$ .....	99
<b>Figure 4.9</b>	Election du cluster-head final .....	100
<b>Figure 4.10</b>	La différence entre Cluster-head tempraire et cluster head final .....	101
<b>Figure 4.11</b>	Pseudo algorithme de la phase de décision.....	102
<b>Figure 4.12</b>	Pseudo algorithme de sélection d’un CH temporaire et finale selon la condition de distance .....	102
<b>Figure 4.13</b>	Phase de formation des clusters .....	103
<b>Figure 4.14</b>	Re-clustering de HEBM.....	104
<b>Figure 4.15</b>	TDMA et les différents rounds de HEBM.....	105
<b>Figure 4.16</b>	Accès multiple par répartition de temps du HEBM.....	105
<b>Figure 4.17</b>	La portée optimale de communication intra et inter cluster .....	106
<b>Figure 4.18</b>	Phase de transmission Inter-cluster.....	107
<b>Figure 4.19</b>	Communication multisauts avec le meilleur chemin .....	107
<b>Figure 4.20</b>	La synchronisation par la technique duty-cycle .....	108
<b>Figure 4.21</b>	Procédure Duty Cycle des CH.....	109
<b>Figure 4.22</b>	Exemple illustratif de la phase de synchronisation de notre proposition Duty Cycle.....	111

## Chapitre 5

<b>Figure 5.1</b>	Les étapes de simulation d’un protocole sous NS2 .....	116
<b>Figure 5.2</b>	Energie moyenne résiduelle des capteurs en fonction du temps.....	118
<b>Figure 5.3</b>	L’énergie consommée au fil de temps .....	119
<b>Figure 5.4</b>	La durée de vie de réseau pour 100 nœuds .....	121
<b>Figure 5.5</b>	La durée de vie de réseau pour 200 nœuds .....	121
<b>Figure 5.6</b>	Pourcentage des noeuds vivants pour chaque protocole. ....	122
<b>Figure 5.7</b>	Le 1 er et le dernier noeud meurt au fil de temps de simulation .....	123
<b>Figure 5.8</b>	La relation entre la position de la BS et la durée de vie .....	123
<b>Figure 5.9</b>	Le nombre de cluster par rapport au nombre de nœuds .....	124

## *Liste des figures*

<b>Figure 5.10</b>	Le nombre de cluster par rapport la portée de transmission .....	125
<b>Figure 5.11</b>	La distribution des cluster-heads de protocole HEBM .....	126
<b>Figure 5.12</b>	La moyenne de latence des paquets par rapport au nombre des nœuds .....	126
<b>Figure 5.13</b>	Le nombre des paquets transmis avec différentes tailles au fil de temps .....	128
<b>Figure 5.14</b>	Le nombre de paquets délivrés à la BS .....	129

# *Liste des Tableaux*

**Tableau 1.1** Cartes d'acquisitions sensorielles..... 11

**Tableau 5.1** Les paramètres de simulation..... 117

**Tableau 5.2** Etude comparative des trois protocoles (FEMCHRP, DEEAC and HEBM)..... 129

# *Introduction Générale*

## **I. Contexte général du travail et problématique**

L'évolution technologique récemment connue dans le domaine de l'électronique, mécanique et aussi le domaine de la communication a donné naissance à une nouvelle génération de réseaux, les réseaux de capteurs sans fil. Cette nouvelle ère de l'informatique embarquée promet de révolutionner notre façon de vivre, de travailler et d'interagir avec l'environnement physique qui nous entoure.

De par leur petite taille, les capteurs sont extrêmement contraints. Ils ne peuvent pas se rivaliser avec les ordinateurs et les terminaux utilisés pour les réseaux Ad hoc classiques en termes de traitement des données, de capacité de stockage et de communication. Etant donné les applications prometteuses ainsi que les problématiques liées aux réseaux de capteurs, l'objectif de cette thèse consiste à étudier et à proposer des algorithmes distribués permettant d'apporter une solution aux problèmes liés à la conservation d'énergie pour les réseaux de capteurs sans fil. Nous nous sommes également attachés au problème du maintien de connectivité et aussi la latence.

Nous avons proposé un algorithme distribué (HEBM : Hierarchical Energy balancing multiph routing protocol) qui peut être adapté facilement aux réseaux avec le passage à l'échelle et qui a pour objectif la minimisation de l'énergie consommée sans dégrader la connectivité entre les nœuds. L'algorithme est basé sur la répartition des cluster-heads et sur le contrôle de puissance. Périodiquement chaque nœud adapte sa puissance de transmission en se basant sur les informations de son voisinage à un et à deux sauts.

Notre contribution a introduit le mécanisme d'équilibrage de la charge grâce à la réalisation d'une bonne distribution des cluster-heads dans le modèle de réseaux et pour améliorer les performances en termes de durée par la réduction du signal de transmission et les messages de contrôle. Le protocole proposé est un protocole efficace en consommation d'énergie et garantit une meilleure distribution des chefs dans le réseau. En fait, dans le cas où un ensemble de capteurs se situe dans la même zone géographique et doit capter la même information, tous ces capteurs vont par la suite communiquer cette information à la station de base. De ce fait, une consommation inutile de l'énergie aura lieu (malgré l'utilisation des mécanismes d'agrégation dans le but d'éliminer les données redondantes) et qui par

# *Introduction générale*

conséquent, dégrade les performances du réseau et diminue la durée de vie des capteurs et du réseau tout entier. Dans ce cas, un mécanisme d'équilibrage de charge et d'énergie devient indispensable afin d'équilibrer la consommation de l'énergie.

Nous avons proposé cet algorithme distribué pour RCSFs qui peut être adapté facilement au passage à l'échelle et qui a pour objectif la minimisation de l'énergie consommée sans dégrader la connectivité entre les nœuds. L'algorithme est basé sur le contrôle de puissance. Périodiquement chaque nœud adapte sa puissance de transmission en se basant sur les informations de son voisinage à un et à deux sauts.

## **II. Organisation de la thèse**

Ce manuscrit s'articule autour de cinq chapitres

**Chapitre 1** : Les réseaux de capteurs définitions et concepts de base : ce chapitre représente une introduction au domaine des réseaux de capteurs sans fil, nous retraçons les principales notions de base des réseaux de capteurs telles que l'architecture matérielle et logicielle du nœud capteur, voir comment ils sont déployés pour former des réseaux de capteurs sans fil, les domaines d'application et nous terminons le chapitre par une petite comparaison entre les réseaux de capteurs et les réseaux Ad Hoc classiques.

**Chapitre 2** : Les techniques de conservation d'énergie dans les réseaux de capteurs sans fil. Dans ce chapitre, nous avons cité les facteurs qui induisent une consommation inutile de l'énergie (surconsommation). Ces facteurs sont nombreux, ils peuvent être engendrés lors de la détection lorsque celle-ci est mal gérée, ou lors de la communication. En effet, la communication est sujette à plusieurs facteurs qui surconsomment de l'énergie notamment côté MAC. Certains de ces facteurs sont les causes majeures de la perte d'énergie et ils ont été recensés dans ce chapitre.

**Chapitre 3**: Les réseaux hiérarchiques et Clustering « Introduction à HEBM proposé »

Dans ce chapitre nous avons donné une étude critique sur les protocoles de routage proposés dans le cadre des réseaux de capteurs sans fil. Il est organisé de la façon suivante, nous commencerons par voir les facteurs qui influencent la conception des protocoles de routage par la suite nous verrons plusieurs classifications des protocoles de routage dans les réseaux de capteurs sans fil. Et nous avons cité les améliorations de notre contribution (HEBM) par rapport aux autres nouveaux protocoles proposés.

# *Introduction générale*

## **Chapitre 4 :** La conception détaillée du protocole HEBM proposé

Ce chapitre a pour objectif d'étudier une proposition de protocole de routage distribué, équilibré, efficace en énergie avec une agrégation hiérarchique de données, une auto-organisation des nœuds, et une évaluation de passage à l'échelle (HEBM : Hierarchical Energy Balancing Multipath routing protocol). Nous avons étudié une conception détaillée du 'HEBM' et les 07 phases proposées avec une description bien organisée de toutes les différentes sous phases.

## **Chapitre 5 :** Implémentation du protocole HEBM et les résultats de simulation

Une implémentation correcte de protocole de routage et d'applications pour les réseaux de capteurs nécessite de passer par une phase très importante c'est la phase de la simulation. Dans ce chapitre nous avons implémenté le protocole HEBM, et après nous avons évalué les performances de nos propositions par un simulateur NS2. Nous schématisons les résultats obtenus à l'aide de graphes que nous exposons et interprétons. Nous comparons les performances entre les différents protocoles, nous prenons en compte plusieurs métriques la consommation d'énergie, et la durée de vie de notre réseau, la fiabilité de données, la QoS, la connectivité des nœuds, les paquets envoyés et reçus, le premier nœud et le dernier nœud qui épuise leur énergie, l'énergie moyenne, la latence de transmission et de recevoir. ....etc.

# Chapitre 1

## Concepts Généraux Sur Les Réseaux De Capteurs Sans Fil

### 1. Introduction

Les réseaux de capteurs sans fils (RCSFs) sont une technologie émergente notamment grâce aux énormes progrès réalisés dans les domaines de la microélectronique et des communications sans fils. A l'heure actuelle, cette technologie, qui permet la surveillance des environnements hostiles et parfois inaccessibles à l'homme, est exploitée dans de nombreux domaines tels que l'industrie, le bâtiment, l'écologie, le domaine médical ou bien encore le militaire. Toutefois, ce type de réseaux présente certaines contraintes liées à la limitation des capteurs en termes de ressources énergétiques, de stockage et capacité de calcul. Successeur des réseaux Ad hoc, cette nouvelle génération de réseau est motivée par la convergence connue récemment entre les domaines de technologie « MEMS » (*Microelectro mechanical systems*), de communication sans fil et d'électronique numérique.

Un réseau de capteurs répond aux besoins accrus d'observer et de contrôler des phénomènes physiques et biologiques dans différents domaines. [1] Chaque capteur collabore avec ses voisins via des liens sans fil, et si nécessaire doit retransmettre les informations venant de capteurs trop éloignés pour communiquer directement avec la station de base qui à son tour retransmet ces données, après les avoir traités, aux utilisateurs finals du réseau. La conception des capteurs à petite taille et à faible coût qui ont la capacité de traiter, détecter, stocker et communiquer des données, et de les intégrer dans ce qu'on appelle réseaux de capteurs sans fil ouvre la possibilité de les intégrer dans différents domaines d'application allant de la surveillance.

### 2. Les réseaux sans fils

De nos jours les technologies sans fils font parties de notre quotidien. Que ce soit des simples télécommandes infrarouges à l'internet sans fil en passant par les périphériques Bluetooth comme les téléphones portables, les souris, etc. aucune maison moderne n'échappe à la règle. On constate que de plus en plus de matériel électronique devient communicant, le plus souvent

sans fils, et on ne voit pas ce qui pourrait freiner cette évolution.[4] Cependant, la majeure partie de ces technologies sans fils, à commencer par le Wifi, est basée sur des infrastructures fixes, limitant la mobilité des utilisateurs. [17] Pour faciliter cette mobilité, il existe un autre type de réseau, de plus en plus courant, qui consiste à permettre aux nœuds du réseau de communiquer directement entre eux sans nécessiter d'infrastructure : ce sont les réseaux ad hoc. On distingue donc deux principales classes de réseaux sans fils, les classiques structurés et les non structurés comme les réseaux ad hoc. Les réseaux ad hoc offrent la possibilité de connecter différents dispositifs sans avoir à préinstaller une infrastructure fixe comme dans les réseaux traditionnels.

### 2.1. Les réseaux ad hoc

Dans les réseaux ad hoc, l'ensemble des nœuds communiquent directement entre eux. C'est-à-dire que cette fois-ci, si l'ordinateur portable veut envoyer un message au Tablet PC, il ne va pas devoir contacter systématiquement un point d'accès, mais il va envoyer les paquets directement au destinataire. Pour cela, grâce à un protocole de routage, il va déterminer une route valide à un moment donné, puis envoyer ses paquets en suivant cette route.



Figure 1.1 Réseau Ad hoc

Dans notre cas une route valide serait : ordinateur portable, mobile 1, Tablet PC. Bien sûr, dans ce cas, toute la difficulté va résider dans le fait de trouver une route valide. (voir figure 1.1). [9] C'est dans ce but que de nombreux protocoles de routage ont été proposés. L'étude de ces protocoles reste d'ailleurs encore aujourd'hui un domaine de recherche très actif.[18] Le but d'un tel protocole va être logiquement de découvrir une route valide et de la maintenir,

mais aussi, de s'assurer de ne pas surcharger un nœud par rapport à un autre, ce qui causerait sa défaillance prématurée. La maintenance va inclure l'optimisation de la route en cas de mobilité des nœuds, la découverte d'une nouvelle route en cas de défaillance d'un nœud intermédiaire, etc. Différentes solutions peuvent être adoptées pour la construction d'une route. Tout d'abord elle peut être construite à la volée, c'est-à-dire au moment d'envoyer un paquet [1], ou bien de manière proactive, c'est-à-dire en prévision d'un envoi possible. La première solution est plus basée sur une approche de découverte de nœuds tandis que la deuxième solution sera elle basée sur des tables recensant les nœuds du réseau.[19] Les routes peuvent ensuite soit être établies à chaque paquet, c'est-à-dire que deux paquets d'une même communication peuvent emprunter deux routes totalement différentes, soit être la même pour toute la communication. D'autres solutions basées sur les coordonnées GPS (Global Positioning System) ont aussi été proposées.

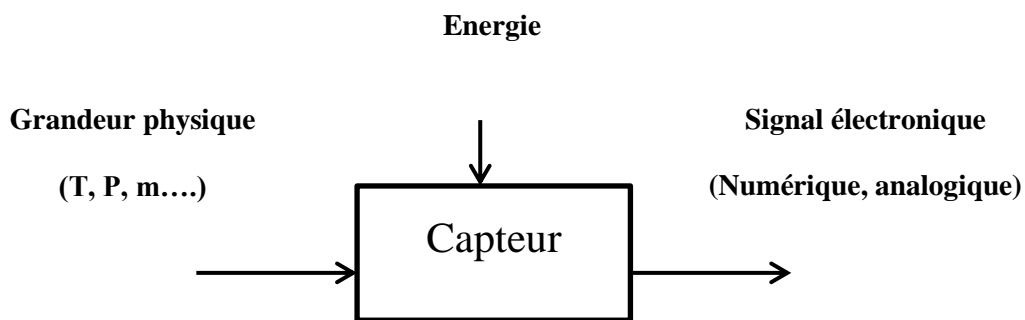
### **2.2. Les réseaux de capteurs sans fils (RCSFs)**

Les réseaux de capteurs sont une application des réseaux ad hoc. Comme dit précédemment, de plus en plus de matériels électroniques sont maintenant communicants, et plus globalement on tend à tout automatiser, à commencer par les tâches répétitives, ingrates, dangereuses ou plus simplement ce que l'on a tendance à oublier. Ainsi on peut imaginer des stylos capables d'alerter un niveau d'encre faible et planifier automatiquement l'achat de nouvelles cartouches d'encre ; on peut imaginer automatiser le ramassage des déchets, leur tri et leur recyclage ; ou encore informatiser la surveillance des frontières. [20] Dans tous ces cas on a besoin de nombreux capteurs, y compris pour l'automatisation des tâches, afin de s'assurer que tout se déroule correctement. Cependant l'ajout et le câblage d'un trop grand nombre de capteurs peut coûter très cher et donc rendre le projet irréalisable. Les réseaux de capteurs sans fils pallient ces difficultés grâce à l'utilisation de réseaux ad hoc.

#### **2.2.1. Les capteurs**

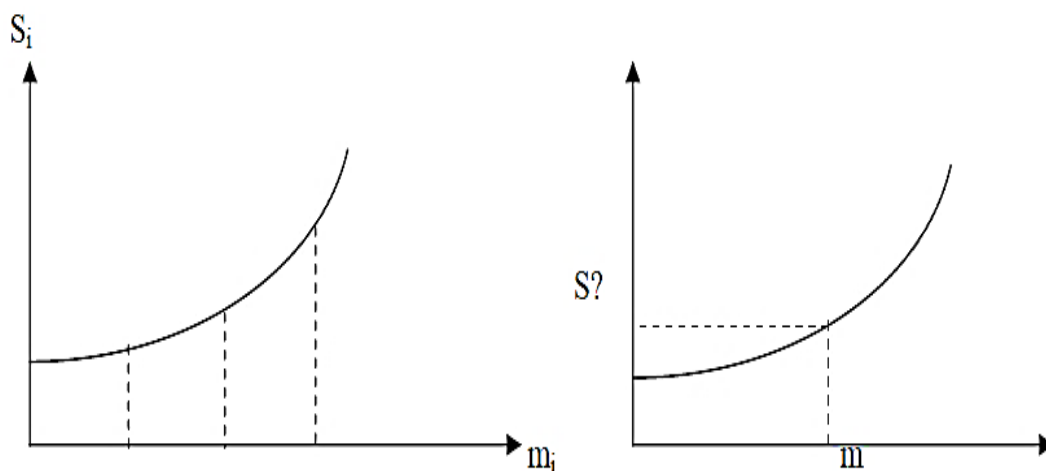
Les capteurs sont de simples petits modules électroniques à faible coût qui, grâce au principe d'auto-organisation des réseaux ad hoc, peuvent être déposés, voir même largués depuis un avion, sans avoir à se soucier de leur connexion et sans avoir à déployer une infrastructure lourde. [5][21] Le principal problème de ces capteurs réside dans le fait que pour être produits et utilisés en grande quantité, ils doivent avoir un coût de production faible. Cela implique donc forcément des capacités réduites : capacité de calcul, capacité de stockage, portée radio. Mais aussi, au-delà du problème du coût, le fait que les capteurs fonctionnent sur la batterie, implique non seulement les mêmes restrictions sur la capacité, mais impose aussi de tout faire

pour maximiser la durée de vie de la pile. Cela peut inclure de mettre en veille le capteur la majeure partie du temps, de limiter le trafic qui circule sur le réseau. (Figure 1.2)



**Figure 1.2** Le fonctionnement d'un capteur

La surveillance des paramètres physiques, chimiques ou biologiques dans une chaîne de mesure requiert souvent l'utilisation des capteurs. Par définition, un capteur est un dispositif qui génère une image sous forme en général électrique, d'une grandeur physique. De manière plus précise, un capteur peut être vu comme une boîte noire qui accepte en entrée une mesurande c'est à dire la quantité que l'on cherche à mesurer ( $m$ ) et produit en sortie une grandeur électrique notée  $S$  pouvant être une charge électrique, un courant, un différentiel de potentiel ou une impédance. [22] Autrement dit, le rôle d'un capteur est de calculer  $S = F(m)$ . Le principe de base repose sur l'opération d'étalonnage. Sachant différentes valeurs  $m_1, m_2, \dots, m_i$ , on relève les signaux  $S_1, S_2, \dots, S_i$ . Le rôle du capteur consiste donc à lire  $S$  lorsque  $m$  est inconnu [3]. Un capteur transforme une grandeur physique en une grandeur normée, (voir figures 1.3 ; 1.4), généralement électrique, qui peut être interprétée par un dispositif de contrôle commande.



**Figure 1.3** Principe de l'étalonnage

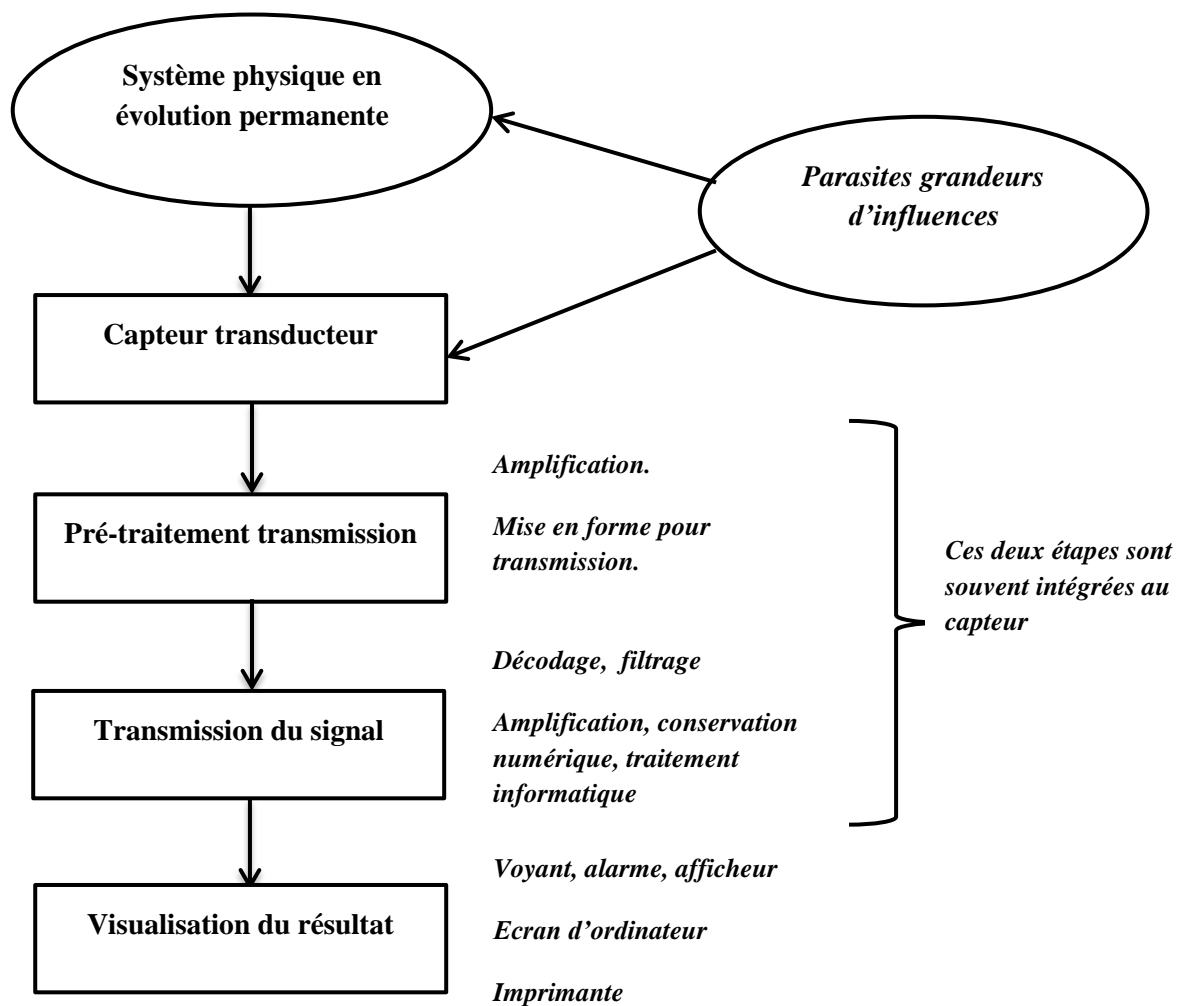


Figure 1.4 Schéma de principe d'un capteur

### 2.2.2. Classification des capteurs

Les capteurs ont plusieurs modes de classification :

- **Apport énergétique**

- ✓ *Capteurs passifs*

Ils n'ont pas besoin d'apport d'énergie extérieure pour fonctionner (exemple : thermistance, potentiomètre, thermomètre à mercure...). [8] Ce sont des capteurs modélisables par une impédance. Une variation du phénomène physique étudié (mesuré) engendre une variation de l'impédance.

- ✓ *Capteurs actifs*

Ils sont constitués d'un ou d'un ensemble de transducteurs alimentés (exemple : chronomètre mécanique, jauge d'extensomètre appelée aussi jauge de contrainte, gyromètre...). [23] Ce sont des capteurs que l'on pourrait modéliser par des générateurs

comme les systèmes photovoltaïques et électromagnétiques. Ainsi ils génèrent soit un courant, soit une tension en fonction de l'intensité du phénomène physique mesuré.

### **2.2.3. Caractéristiques d'un capteur**

- Etendue de mesure : valeurs extrêmes pouvant être mesurée par le capteur.
- Résolution : plus petite variation de grandeur mesurable par le capteur.
- Sensibilité : variation du signal de sortie par rapport à la variation du signal d'entrée.
- Précision : aptitude du capteur à donner une mesure proche de la valeur vraie.
- Rapidité : temps de réaction du capteur.
- La rapidité est liée à la bande passante.
- Linéarité : représente l'écart de sensibilité sur l'étendue de mesure.

La structure de base d'une chaîne de mesure comprend au minimum quatre étages [8] : Un capteur sensible aux variations d'une grandeur physique et qui, à partir de ces variations, délivre un signal électrique. Un conditionneur de signal dont le rôle principal est l'amplification du signal délivré par le capteur pour lui donner un niveau compatible avec l'unité de numérisation; cet étage peut parfois intégrer un filtre qui réduit les perturbations présentes sur le signal. Une unité de numérisation qui va échantillonner le signal à intervalles réguliers et affecter un nombre (image de la tension) à chaque point d'échantillonnage. L'unité de traitement informatique peut exploiter les mesures qui sont maintenant une suite de nombres (enregistrement, affichage de courbes, traitements Mathématiques, transmissions des données ...). [24][27]De nos jours, compte tenu des possibilités offertes par l'électronique et l'informatique, les capteurs délivrent un signal électrique et la quasi-totalité des chaînes de mesure sont des chaînes électroniques et informatiques.

### **2.2.4. Architecture d'un nœud capteur**

Les nœuds capteurs sont l'élément central d'un RCSF. La détection, le traitement et la communication ont lieu grâce à un nœud. Il stocke et exécute les protocoles de communication et les algorithmes de traitement des données. La qualité, la taille et la fréquence des données détectées qui peuvent être extraites à partir du réseau sont influencées par les ressources physiques disponibles pour le nœud. Par conséquent, la conception et la mise en œuvre d'un nœud capteur sans fil sont des étapes cruciales. Un nœud est constitué d'un sous-système de captage, un sous-système de traitement, un sous-système de communication et un sous-système d'alimentation (voir figure 1.5). Selon la spécificité de l'application, les nœuds capteurs peuvent également inclure des composants supplémentaires comme un système de recherche d'emplacement pour déterminer leur position (par exemple

GPS), un mobilisateur pour changer leur emplacement ou leur configuration (par exemple, l'orientation de l'antenne), et ainsi de suite. Le concepteur a une pléthore d'options pour décider comment construire et mettre l'ensemble de ces sous-systèmes dans un nœud programmable unique. [25] Le sous-système de traitement est l'élément central du nœud et le choix d'un processeur détermine le compromis entre la flexibilité et l'efficacité - en termes d'énergie et de rendement. Il y a plusieurs processeurs en option : microcontrôleurs, DSP (Digital Signal Processor), les circuits intégrés spécifiques à l'application (Application-Specific Integrated Circuits) et le circuit intégré comportant une matrice de portes (Field Programmable Gate Arrays).

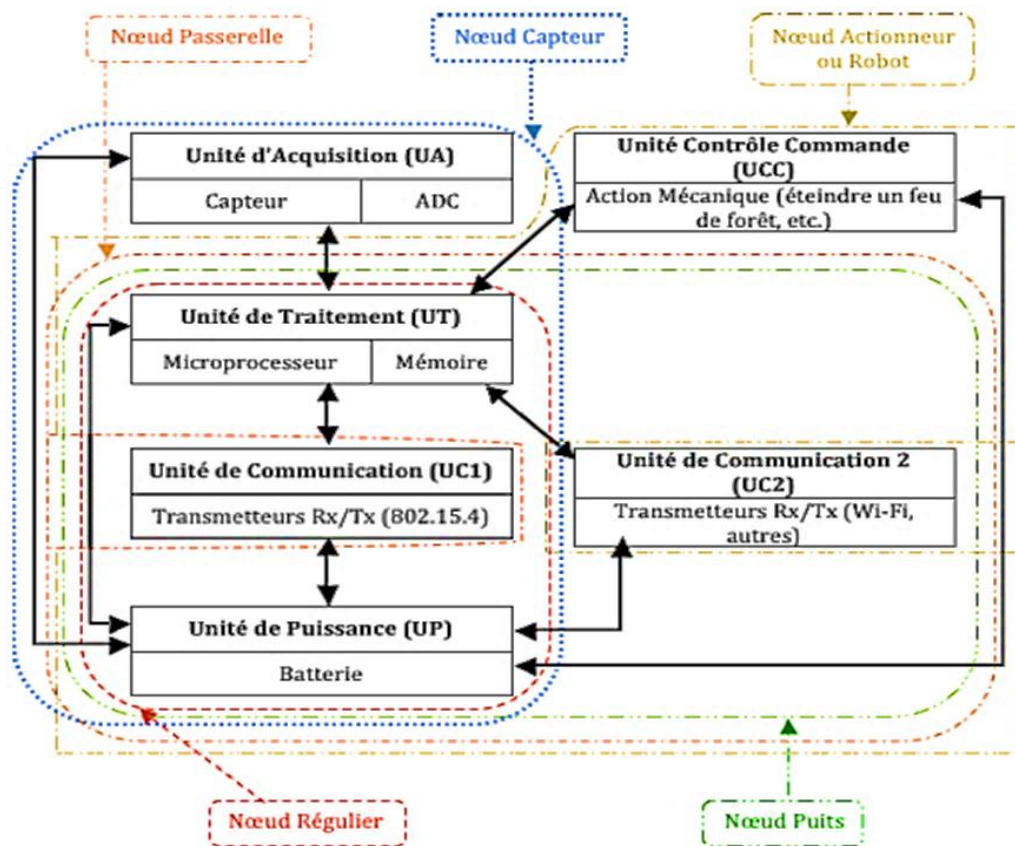


Figure 1.5 Architecture des différents types de nœuds : capteur, robot, puits, passerelle

### i. Sous-système d'acquisition

Le sous-système d'acquisition intègre un ou plusieurs capteurs physiques et fournit un ou plusieurs convertisseurs analogiques numériques ainsi que le mécanisme de multiplexage pour les partager.[26] Les capteurs interfacent le monde virtuel avec le monde physique. La détection des phénomènes physiques n'est pas quelque chose de nouveau. L'avènement de systèmes micro-électromécaniques (MEMS) a réalisé la détection d'un processus

omniprésent. Aujourd'hui, il existe une multitude de capteurs qui mesurent et quantifient les caractéristiques physiques à un prix bon marché. Un capteur physique contient un transducteur, un dispositif qui convertit une forme d'énergie en une autre, généralement en une énergie électrique (tension). La sortie de ce transducteur est un signal analogique ayant une amplitude constante en fonction du temps.[10] Par conséquent, un convertisseur analogique-numérique est nécessaire pour interfacer un sous-système de détection avec un processeur numérique. Le convertisseur analogique-numérique (ADC) convertit la sortie d'un capteur - qui est, un signal analogique continu - en un signal numérique. Ce processus requiert deux étapes :

1. Le signal analogique doit être quantifié (c.-à-d. converti à partir d'un signal à valeur continue en un signal à valeur discrète à la fois dans le temps et en fréquence). La décision la plus importante à ce stade est de déterminer le nombre de valeurs discrètes admissibles. Cette décision est à son tour influencée par deux facteurs : (a) la fréquence et l'amplitude du signal, et (b) la disponibilité des ressources de traitement et de stockage.
2. La fréquence d'échantillonnage. Dans l'ingénierie de la communication et de traitement numérique du signal, cette fréquence est choisie par le taux de Nyquist. Cependant, dans les RCSFs, le taux de *Nyquist* ne suffit pas. Un sur-échantillonnage est nécessaire à cause du bruit.

### **ii. Sous-système de traitement**

Le sous-système de traitement rassemble tous les autres sous-systèmes ainsi que les périphériques supplémentaires. Son but principal est de traiter (exécuter) des instructions relatives à la détection, la collecte, la communication et l'auto-organisation. Il se compose d'un puce processeur, une mémoire non volatile (généralement une mémoire flash interne) pour le stockage des instructions de programme, une mémoire active pour stocker temporairement les données détectées et une horloge interne. Ce sous-système est composé de deux interfaces, une interface avec le sous-système de capture et une interface avec le sous-système de transmission. [28] Il acquiert les informations en provenance du sous-système de capture et les envoie au sous-système de transmission.

### **iii. Sous-système de transmission**

Le sous-système de transmission est responsable de toutes les émissions et réceptions de données via un support de communication radio. Comme le choix du bon type de processeur est vital pour les performances, ainsi que la consommation d'énergie d'un nœud capteur sans fil, la manière dont les sous-composants sont interconnectés avec le sous-système de

traitement est également essentielle. Un transfert rapide et économe en énergie de données entre les sous-systèmes d'un nœud capteurs est essentiel à l'efficacité globale du réseau mis en place. [30] Cependant, la taille pratique du nœud met une restriction sur les bus du système. Puisque la communication via un bus parallèle est plus rapide qu'un bus série, un bus parallèle a besoin de plus d'espace. De plus, il nécessite une ligne dédiée pour chaque bit qui devrait être transmis simultanément tandis que le bus série nécessite une ligne de données uniques. En raison de la taille du nœud, les bus parallèles ne sont jamais utilisés dans la conception du nœud. Donc, le choix est souvent fait entre les interfaces série telles que SPI (Serial Peripheral Interface), GPIO (General Purpose Input/Output), SDIO (Secure Data Input/Output), I2C (Inter-Integrated Circuit) et USB (Universal Serial Bus). [29] Parmi ceux-ci, les bus les plus couramment utilisés sont SPI et I2C.

### **iv. Sous-système d'alimentation**

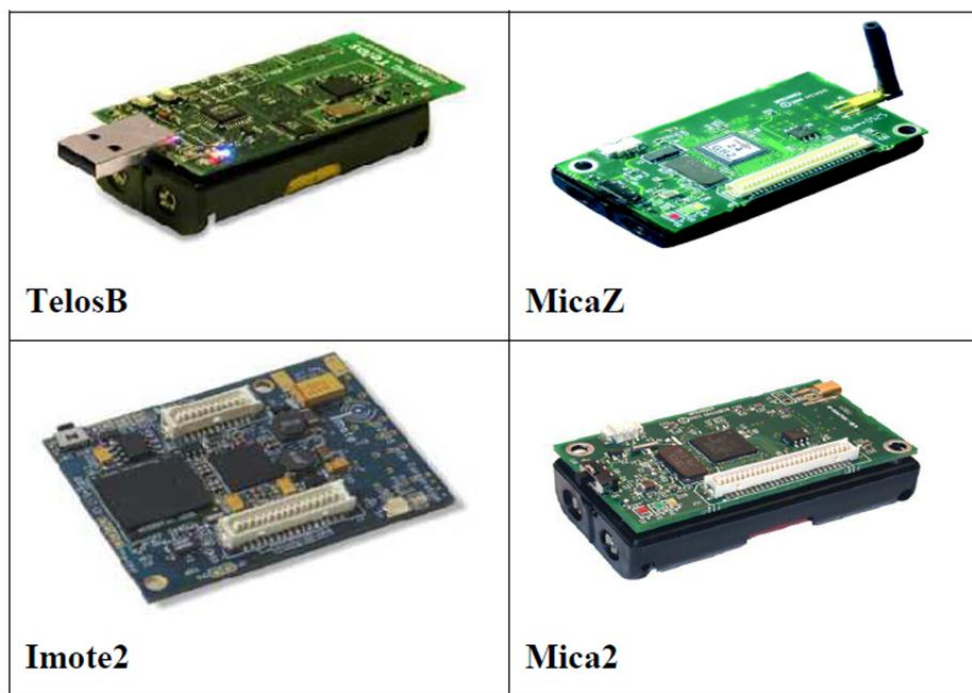
La gestion de l'énergie est un point primordial dans les réseaux de capteurs. Tout capteur est muni d'une ressource énergétique à durée de vie limitée, et ce, à cause de la taille réduite de la batterie. [11] Il est à noter que la gestion de l'énergie est un problème réel pour les réseaux de capteurs, et divers travaux tentent d'y remédier.

De ce fait, l'unité de contrôle d'énergie est sans doute le composant le plus important du capteur. Elle s'occupe de la répartition de l'énergie au sein du capteur entre les divers modules, elle permet également quelques fois de réduire la consommation d'énergie en agissant sur les modules inactifs. [31] Le niveau de batterie est un critère primordial pour les décisions de participation au routage et de transmission ; un nœud ayant un niveau de batterie critique ne participe pas au routage. On peut aussi conditionner la transmission lorsque le niveau de batterie devient critique, c'est-à-dire une fois le niveau critique atteint, le nœud ne transmet que si l'information captée est elle aussi critique. Un autre remède au problème de gestion de l'énergie, consiste à profiter de la densité des nœuds : dans une région dense, certains capteurs doivent se sacrifier au captage et d'autres au routage et un troisième groupe au remplacement des nœuds ayant consommés la totalité de leurs réserves d'énergie. La consommation de l'énergie dans un réseau de capteurs est définie par l'ordre décroissant suivant :

- La communication (réception, émission).
- Le traitement des données.
- La capture des données.

### 2.2.5. Les plates-formes existantes

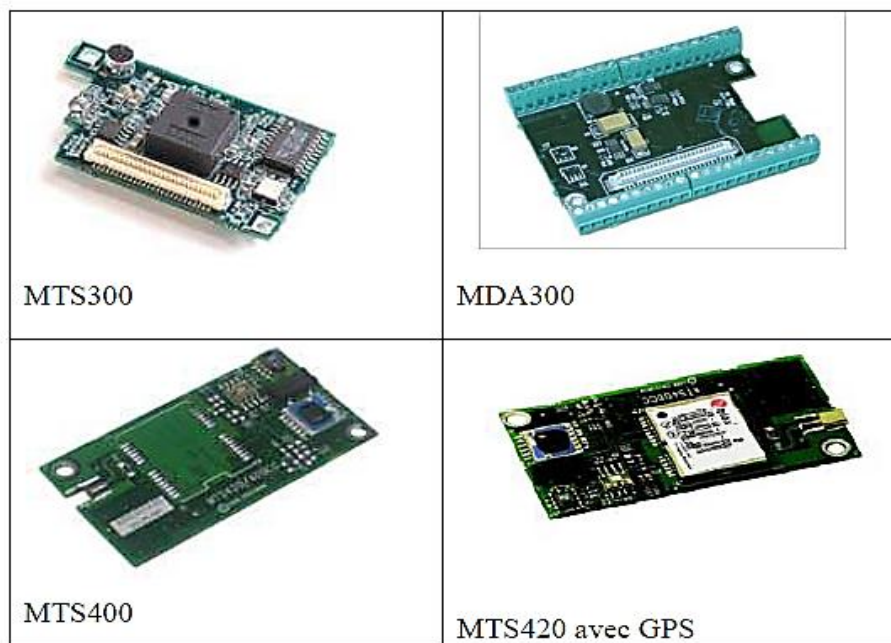
Comme un certain nombre de technologies connues à ce jour, les noeuds de capteurs sans fil doivent être nés d'un projet militaire, ce qui entrave la mise en place d'une chronographie précise de leur développement. Cependant, le titre du premier prototype de noeuds de capteurs sans fil identifiable dans la bibliographie correspond sans aucun doute au module LWIM (*Low-power Wireless Integrated Microsensors*) développé dans le milieu des années 90 par l'agence pour les projets de recherche avancée de défense (DARPA) des Etats-Unis et l'UCLA. [32] Il s'agissait d'un géophone équipé d'un capteur de transmission radio fréquences et d'un contrôleur PIC. Depuis un peu plus de 10 ans, la technologie des capteurs sans fil a beaucoup évolué. Les modules deviennent de plus en plus petits et les durées de vie prévues augmentent. Aujourd'hui, le marché de noeuds a été ouvert à l'industrie. Le fournisseur le plus connu est *Crossbow Inc.*, avec son offre de capteurs Mica2 et MicaZ (voir Figure 1.6).



**Figure 1.6** Quelques modèles de capteurs sans fil

De même, nous pouvons combiner le nœud Mica2 avec une carte MTS420 pour le doter d'un capteur d'humidité et d'un capteur de pression barométrique, et même d'un GPS pour le positionnement géographique. Une autre possibilité pour la même unité est l'ajout d'une carte d'acquisition MDA320. [33] La Figure 1.7 et le tableau 1.1 montrent les cartes intégrées peuvent être combinées avec un mote. La plupart des fabricants adoptent des émetteurs RF à basse fréquence. Certains ont choisi de mettre en œuvre un protocole d'origine récente conçu

pour les modules sans fil industriels et spécifié dans la norme IEEE 802.15.4. Ce protocole de transmission opère dans la bande de fréquences des 2.4GHz. Les microcontrôleurs choisis sont généralement d'une faible vitesse et de très faible consommation d'énergie. De même, la mémoire disponible pour les programmes et les données est très réduite en comparaison avec celle des équipements informatiques d'aujourd'hui.



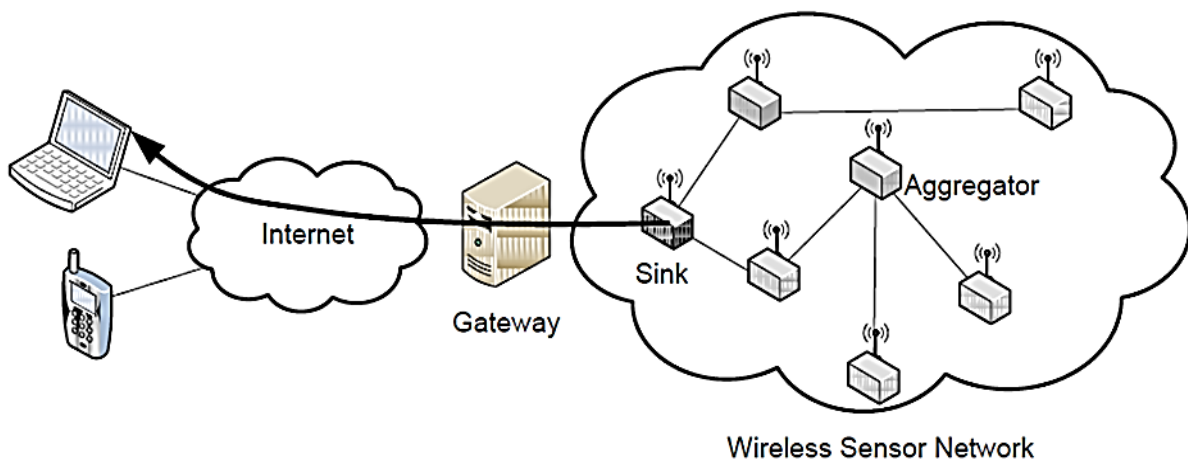
**Figure 1.7** Quelques cartes De captage.

Carte d'acquisition	Type de mesures captées
MTS300	Lumière, température et son.
MTS310	Lumière, température, son, accéléromètre X et Y, magnétomètre X et Y
MTS400	Pression barométrique, température, humidité relative, lumière et un accéléromètre X et Y.
MTS420	Pression barométrique, température, humidité relative, lumière, un accéléromètre X et Y et GPS.
MTS510	Lumière, son et un accéléromètre X et Y.
MDA300	Température et humidité.

**Tableau 1.1** Cartes d'acquisitions sensorielles

### 3. Qu'est-ce qu'un réseau de capteurs ?

Traditionnellement, l'emploi des capteurs était limité au prélèvement de la température, l'humidité, la vitesse, par la suite on a pensé à rajouter des capacités de communication en ayant la possibilité d'accéder à distance aux grandeurs mesurées par les capteurs. L'idée consiste donc à déployer des capteurs dans un champ. Un ou plusieurs nœuds capteurs prennent le rôle de source (entité qui fournit des informations). [34] De plus, on a besoin d'une ou plusieurs stations de base (entité où l'information est requise). (Voir figure 1.8) Ça peut être un PDA ou un Gateway à un autre réseau Internet, satellitaire, wifi, wimax... Dans ce contexte, un ou plusieurs utilisateurs peuvent avoir accès aux informations collectées et acheminées par les capteurs qui assurent la double fonction de collecte des données et celle du routage. [35] En particulier, l'interaction avec l'environnement, le traitement des informations et la communication de manière locale constituent les éléments clés d'un réseau de capteurs.



**Figure 1.8** l'architecture habituelle des réseaux de capteurs sans fils

La Figure 1.8 représente l'architecture habituelle des réseaux de capteurs sans fils. Ils sont construits autour des quatre principales entités suivantes :

#### **a) Le capteur (sensor)**

Comme le dit bien son nom, il est en charge de mesurer une valeur relative à son environnement (température, pression, luminosité, présence, etc.).

On peut parfois rencontrer des capteurs-actionneurs qui non seulement mesureront mais auront aussi pour rôle d'entreprendre une action en fonction de la valeur mesurée. L'intelligence nécessaire à la prise de décision quant à l'action à entreprendre peut alors être déportée sur un autre nœud du réseau.

### **b) L'agrégateur (aggregator)**

Il est en charge d'agréger les messages qu'il reçoit de plusieurs capteurs puis de les envoyer en un seul message au puits (sink). Cette opération a pour principal but de limiter le trafic sur le réseau et donc de prolonger la durée de vie globale du réseau de capteur. Il correspond généralement à la tête d'une grappe (cluster head). [36][37]L'utilisation de grappes offre de nombreux intérêts à tous les niveaux, notamment pour le routage.

Dans les réseaux de capteurs sans fil, le trafic de données va essentiellement dépendre de la raison pour laquelle le réseau a été déployé. Il est possible de distinguer trois types de trafic de données qui peuvent être utilisés seuls ou conjointement :

- **Applications orientées événements** Dans ce cas, les capteurs envoient leurs données seulement si un événement spécifique se produit. On peut citer l'exemple de surveillance des feux dans les forêts où un capteur envoie des alarmes à la station de base dès que la température dépasse un certain seuil. [38]Au départ, cette classe d'application était conçue à des fins militaires, comme la surveillance du déplacement d'objets dans le champ de bataille. Par la suite, cette classe a rapidement trouvé de nouvelles perspectives comme le contrôle industriel, le contrôle médical des patients, la surveillance d'édifices (barrages, ponts, voies de chemins de fer, etc.)
- **Applications orientées requêtes** Dans ce cas, un capteur envoie de l'information uniquement suite à une demande explicite de la station de base. Cette classe d'application est destinée aux applications adaptées à l'utilisateur. [39] Ce dernier peut requérir des informations à partir de certaines régions dans le réseau ou interroger les capteurs pour acquérir des mesures d'intérêts. Dans ce cas, des connaissances sur la topologie du réseau et l'emplacement des capteurs sont nécessaires.
- **Applications hybrides** Ce type d'application met en œuvre les trois modes de fonctionnement décrits précédemment. Par exemple, dans un réseau conçu pour le suivi d'objets, le réseau peut combiner entre un réseau de surveillance (time driven) et un réseau de collecte de données par événements (event driven). [41] Par exemple, pendant les longues périodes d'inactivité des capteurs et lorsque aucun objet n'est présent, le réseau peut assurer une fonction de surveillance.

### **c) Le puits (sink)**

Le puits est le noeud final du réseau. C'est à lui qu'est envoyé l'ensemble des valeurs mesurées par le réseau. Il peut arriver qu'il y ait plusieurs puits sur un même réseau de capteurs.

### **d) La passerelle (gateway)**

La passerelle est un dispositif qui a la particularité d'avoir deux interfaces réseau. Il permet de relier le réseau de capteurs sans fils à un réseau plus traditionnel, typiquement l'internet. En effet, habituellement le réseau de capteurs ne sert qu'à faire remonter les mesures, les applications traitant ces informations étant exécutées sur la machine de l'utilisateur final.[40] Le fonctionnement global de cette architecture consiste donc à ce que les capteurs fassent des mesures qu'ils font remonter au puits via les agrégateurs. L'application finale tournant sur une machine se situant sur un autre réseau a ainsi accès aux valeurs via une passerelle. À noter que les agrégateurs sont facultatifs, et que le puits et la passerelle sont généralement localisés dans un seul dispositif.

### **3.1. Le réseau des capteurs mobiles**

La mobilité est une question clé pour les réseaux de capteurs où chaque nœud peut se déplacer à l'intérieur du site, seul ou avec un groupe. Par exemple, quand les capteurs sont embarqués sur des dispositifs mobiles tels que les véhicules, ou sur des animaux. Lorsque la mobilité est trop fréquente, elle ne peut être considérée comme un problème secondaire. [42][43] Ainsi, la détection des voisins et la reconfiguration du réseau exigent habituellement un nombre important de messages de contrôle de la topologie, donc une dépense importante d'énergie.

### **3.2. Les états opérationnels d'un nœud capteur**

Le diagramme suivant présente les différents états possibles lors de fonctionnement d'un nœud capteur. Nous voyons dans (la figure 1.9) que l'état du capteur dépend de son niveau d'énergie. La radio opère dans quatre modes de fonctionnement : émission, réception, "idle", et sommeil. Une observation importante dans le cas de la plupart des radios est que le mode "idle" induit une consommation d'énergie significative, presque égale à la consommation en mode réception. Ainsi, il est plus judicieux d'éteindre complètement la radio plutôt que de passer en mode "idle" quand l'on a ni à émettre ni à recevoir de données. Un autre facteur déterminant est que, le passage de la radio d'un mode à un autre engendre une dissipation d'énergie importante due à l'activité des circuits électroniques. [44]Par exemple, quand la radio passe du mode sommeil au mode émission pour envoyer un paquet, une importante quantité d'énergie est consommée pour le démarrage de l'émetteur lui-même. Un autre point important est que les données des constructeurs sous-estiment assez régulièrement ces différentes consommations comme ont pu le montrer les auteurs de [40], en particulier concernant la consommation dans le mode "idle".

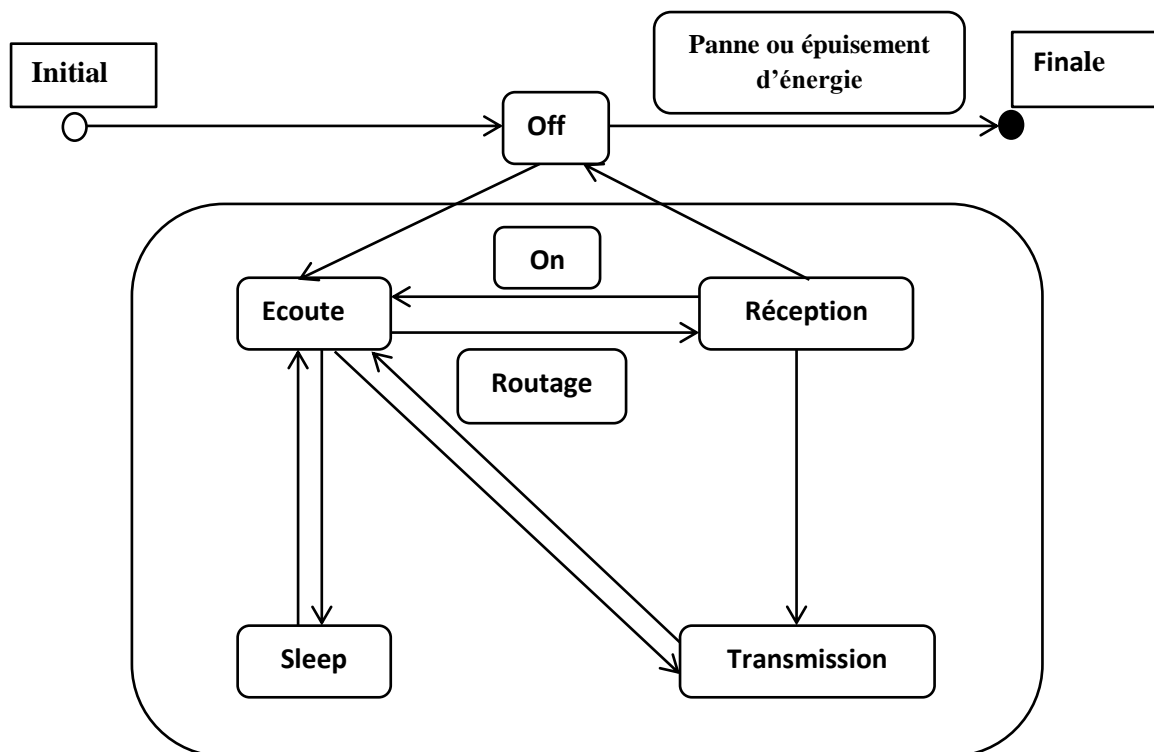


Figure 1.9 Les états possibles d'un nœud capteur [40]

### 3.3. Les principales caractéristiques et contraintes des RCSFs

- **La consommation réduite l'énergie**

Les nœuds capteurs utilisent des batteries de taille minuscule comme ressources en énergie, ce qui limite leur durée de vie. La spécificité des applications des RCSFs fait que la recharge ou le remplacement de ces batteries est une tâche difficile ou presque impossible. [44]Ce qui nous mène à déduire que la durée de vie d'un nœud est essentiellement dépendante de la durée de vie de la batterie. Ainsi, la méthode de gestion de consommation d'énergie constitué une contrainte majeure dans ce type de réseau.

- **L'auto-configuration des nœuds capteurs**

Dans un RCSF, les nœuds sont déployés soit d'une manière (missile, avion,...), soit déplacés nœud par nœud par un humain ou n robot, et ceci à l'intérieur ou autour du phénomène observé (champ de guerre, surface volcanique, patient malade..). [14]Ainsi, un nœud capteur soit avoir des capacités d'une part, pour s'auto-configurer dans le réseau, et d'autre part pour collaborer avec les autres nœuds dans le but de reconfigurer dynamiquement le réseau en cas de changement de topologie du réseau.

Dans un RCSFs, chaque nœud X possède une unité émettrice/réceptrice qui lui permet de communiquer avec les nœuds qui lui sont proches. En échangeant des informations avec ces

derniers, le nœud X pourra alors découvrir ses nœuds voisins et ainsi connaître la méthode de routage qu'il va adopter selon les besoins de l'application. L'auto-configuration apparaît comme une caractéristique nécessaire dans le cas des RCSF étant donné que d'une part, leur déploiement s'effectue d'une manière aléatoire dans la majorité des applications, et d'autre part le nombre des nœuds capteurs est très grand. En revanche, avec une approche aléatoire, les capteurs sont éparpillés.

- **L'évolutivité**

Contrairement aux réseaux sans fil traditionnels (personnel, local ou étendu), un RCSF peut contenir un très grand nombre de nœuds capteurs (des centaines, des milliers..). [45] Un réseau de capteur est évolutive parce qu'il a la faculté d'accepter un très grand nombre de nœuds qui collaborent ensemble afin d'atteindre un objectif commun.

- **La tolérance aux pannes**

La tolérance aux pannes est la capacité de maintenir un réseau de capteurs et d'assurer les fonctionnalités sans aucune interruption due aux échecs des nœuds. Dans le cas de mal fonctionnement d'un nœud (manque d'énergie, interférences avec l'environnement d'observation...) ou aussi en cas d'ajout de nouveaux nœuds capteurs dans le réseau, ce nœud doit continuer à fonctionner normalement sans interruption. Ceci explique le fait qu'un RCSF n'adopte pas topologie fixe mais plutôt dynamique. Un premier défi sera donc d'identifier et modéliser formellement les modes de défaillances des capteurs, puis de repenser les techniques de tolérance aux fautes à mettre en œuvre.

- **Une densité importante des noeuds**

Les RCSF sont caractérisés par leur forte densité. Cette densité peut atteindre, selon le type d'application. La densité peut varier de quelques nœuds à quelques centaines dans une région. Les algorithmes de routage doivent être capables de fonctionner efficacement avec un grand nombre de capteurs. De plus, ces algorithmes doivent traiter un grand nombre d'évènement sans être assurés.

- **La surveillance**

Pour l'augmentation de la fiabilité, de la disponibilité et de la sûreté de fonctionnement des processus, des systèmes de surveillance sont mis en œuvre dont l'objectif est d'être capable à tout instant, de fournir l'état de fonctionnement des différents équipements constitutifs d'un processus technologiques. Tant au niveau de la détection et l'isolation des fautes (FDI) qu'au niveau de la tolérance aux fautes (FTC). L'opérateur de supervision gère deux types

d'information, le premier concerne la détection et l'isolation des défauts survenus, et le deuxième indique les possibilités de laisser fonctionner ou non le processus.

- **La capacité de communication**

Elle peut prendre deux aspects : multi-sauts ou à un seul saut. Parce-que le multi-sauts est moins énergivore, il reste le type de communication le plus sollicité par les applications de RCSFs qui requièrent une faible consommation d'énergie. [22] Les capteurs ont des possibilités de rassembler des données et de les acheminer vers le Sink, par une architecture multi-sauts. Le sink peut communiquer avec l'utilisateur final ou le décideur par l'intermédiaire d'un réseau de transport tel que l'internet ou le satellite.

#### 4. Les types de communication

Il existe différents types de communication utilisée dans les RCSFs [21] :

- **Unicast** : ce type de communication est utilisé pour échanger des informations entre deux nœuds sur le réseau.
- **Broadcast** : Le sink transmet des informations vers tous les nœuds du réseau. Ces informations peuvent être des requêtes de données bien précises. (ex : la température de la région A).
- **Local Gossip** : ce type de communication est utilisé par des nœuds situés dans une région bien déterminée qui collaborent ensemble afin d'avoir une meilleure estimation de l'évènement observé et d'éviter l'émission du même message vers le nœuds « Sink » ce qui contribue à consommer moins d'énergie.
- **Convergecast** : il est utilisé dans les communications entre un groupe de nœuds et un nœud bien spécifique (qui peut être le « sink »). L'avantage de ce type de communication est la diminution de contrôle d'entête des paquets ('Contrôle Overhead') ce qui économise l'énergie au niveau du nœud récepteur.
- **Multicast** : il permet une communication entre un nœud et un groupe de nœuds. Ce type de communication est utilisé dans les protocoles qui incluent le 'clustering' dans lesquels le cluster-head s'intéresse à communiquer avec un groupe de nœuds.

#### 5. Rôle de la pile de communication

Une pile de communication type pour capteur est montrée sur la (figure 1.10). La première étape consiste à permettre à des capteurs adjacents de communiquer en utilisant la radio. C'est le rôle de la couche physique. Elle sélectionne la fréquence d'émission. Elle est également responsable de la détection du signal, de la modulation, de la conversion des bits en signaux électriques ou optiques, et inversement. Une fois qu'il est possible pour des paires de capteurs

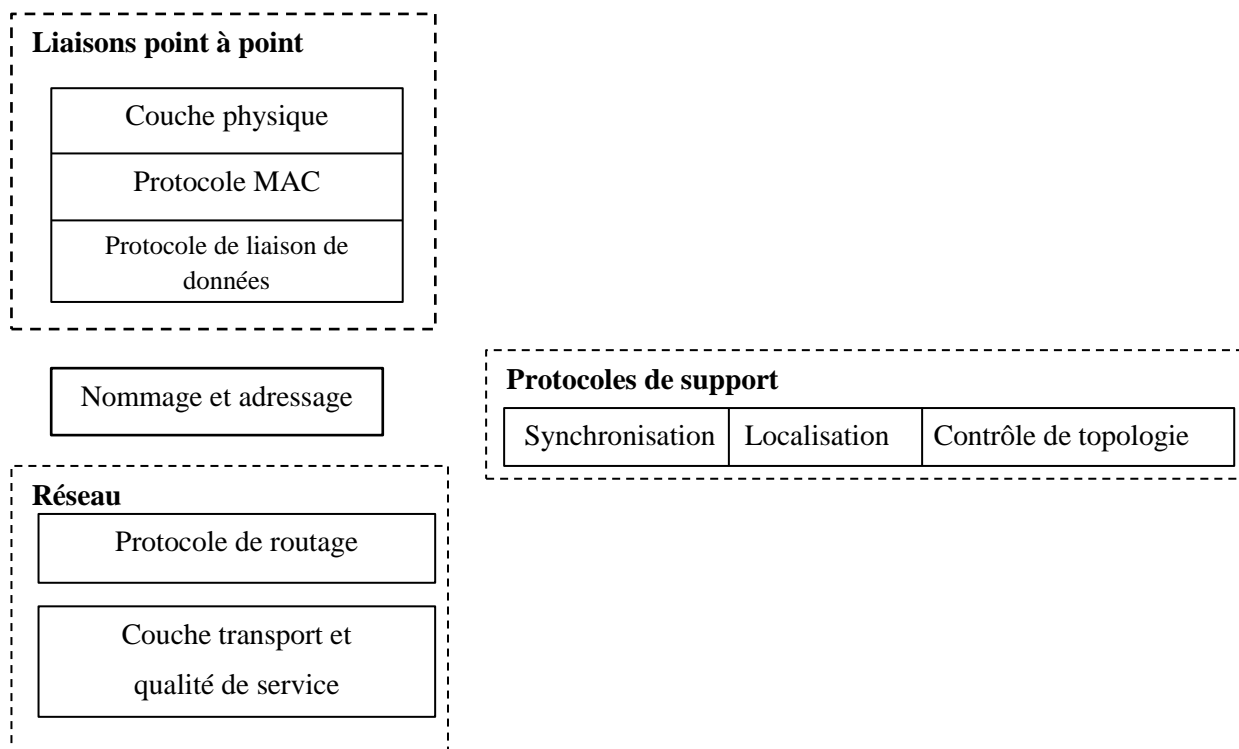
suffisamment proches de communiquer, il faut s'assurer que ceux qui se trouvent dans la même zone n'essaient pas d'émettre en même temps. En effet, cela pourrait causer des collisions. Il faut donc contrôler l'accès au médium pour des nœuds relativement proches.

○ **Le protocole MAC (Medium Access Protocol)** est en charge de la régulation de cet accès au médium pour éviter au maximum les collisions. Il doit donc s'assurer la fiabilité des communications point-à-point, maximiser l'utilisation de la bande passante, assurer l'équité dans l'attribution de la bande passante, minimiser la latence, et en priorité minimiser la consommation d'énergie. [23] Cependant, il n'est pas responsable de contrôler les données destinées aux couches supérieures.

Il s'agit du rôle du **protocole de liaison de données** qui assure le découpage des données en paquets. La taille des paquets a une influence non négligeable sur les performances du réseau. Elle assure également le contrôle d'erreur qui consiste à détecter les paquets non reçus par le destinataire ou trop erronés pour être utilisable. Elle peut aussi être responsable de la gestion de liens : découverte, mise en place et de la maintenance des liens entre nœuds voisins. Cet ensemble de protocoles permet donc de gérer les communications point-à-point. Si deux nœuds souhaitant communiquer ne sont pas à portée de communication directe, il faut être capable de sélectionner un certain nombre de relais responsable d'acheminer le message jusqu'à sa destination.[23] Le **protocole de routage** permet à un nœud de déterminer le chemin à emprunter pour envoyer un message à un nœud donné (le puits par exemple). Son rôle est donc essentiel : il a un effet important sur la fiabilité du réseau, la faible consommation énergétique des nœuds et plus généralement sur l'efficacité du réseau. L'utilisation d'un protocole de routage suppose donc qu'il existe un moyen d'identifier chaque nœud. L'**entité de nommage et d'adressage** est responsable d'attribuer à chaque nœud un identifiant permettant de faciliter les opérations de routage. Comme le protocole MAC, le protocole de routage ne s'occupe que d'acheminer les données au destinataire, sans en vérifier la consistance.

La **couche transport** gère les communications de bout en bout (contrôle d'erreurs, contrôle de flux, découpage en fragments pour les messages volumineux, etc). Elle assure ainsi la bonne réception des paquets et la qualité de service. Pour être capable d'assurer leur fonctions, ces protocoles peuvent requérir des informations particulières, telle que la position des nœuds pour un routage utilisant la position géographique pour sélectionner le prochain relais. Ces informations sont fournies par les protocoles de support. Par exemple, il peut exister un protocole de support s'assurant de la synchronisation entre les nœuds du réseau,

c'est-à-dire vérifiant que chaque nœud possède le même temps que les autres nœuds du réseau. Si le capteur n'est pas équipé d'un GPS (coûteux autant au point de vue pécunier qu'un niveau énergétique), un protocole peut lui fournir une approximation de sa position, ou de la position d'autres nœuds du réseau.



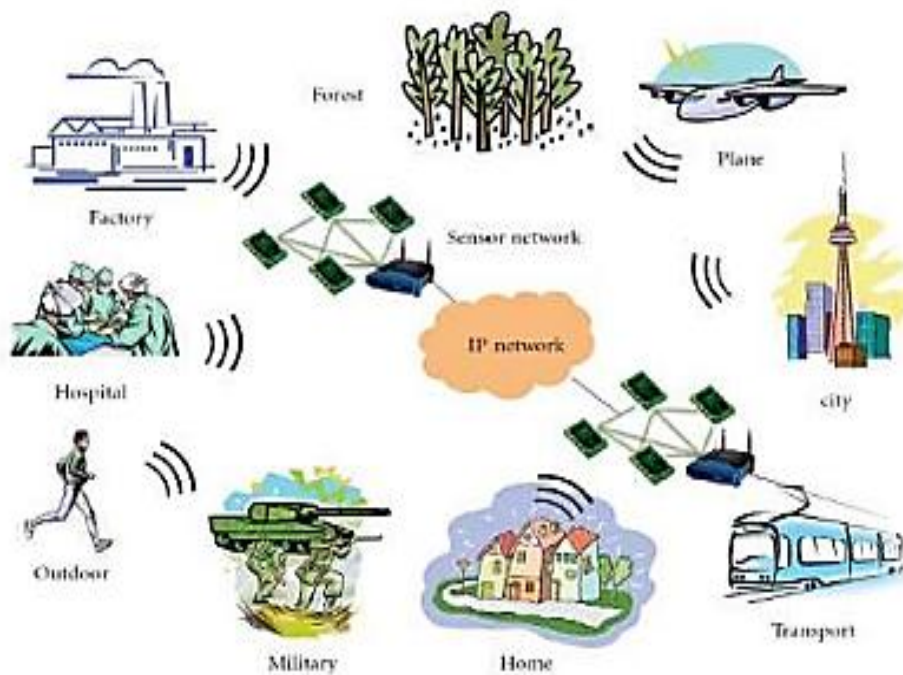
**Figure 1.10** Pile de communication d'un capteur

Lors de déploiement d'un réseau de capteurs, il peut être nécessaire de déployer un nombre important de capteurs pour être certains de recouvrir une zone donnée. Cependant, dans cet éventualité, chaque nœud possède un nombre conséquent de voisins dans sa zone de communication. [11] Dans la plus part de des piles de communication, chaque nœud prend connaissance de ce voisinage et maintenir des informations sur chacun de ces voisins, ce qui est couteux en énergie et en espace mémoire.

Le **contrôle de topologie** consiste donc à gérer les relations de voisinage et ainsi éliminer les liens considérés comme redondants. Cette élimination peut s'effectuer de diverses manières comme par l'établissement de hiérarchies : un nœud est responsable d'un ensemble de capteurs et en contrôle les communications avec les nœuds d'autres zones. Une autre manière est de limiter la portée de communication de chaque nœud, c'est-à-dire de réduire sa zone de communication.

## 6. Scénarios d'utilisation des réseaux de capteurs sans fils

Comme ces réseaux sont basés sur le déploiement dense d'un grand nombre de capteurs jetables et à coût réduit, la destruction de certains capteurs n'affectera pas une opération militaire entreprise, ce qui rend le concept de réseau de capteurs une meilleure approche pour les champs de bataille. (figure1.11) [46] De plus, il existe d'autres applications militaires auxquelles les réseaux de capteurs peuvent être appliqués :



**Figure 1.11** Scénarios d'utilisation des RCSFs

- **Le contrôle des forces, équipement et munition**

Les chefs des troupes et les commandants d'opérations militaires peuvent constamment surveiller l'état des troupes, les conditions et la disponibilité des équipements et de munitions en utilisant les réseaux de capteurs. Chaque troupe, véhicule, équipement ou munition critique peut porter plusieurs capteurs qui rapportent leurs états d'une manière régulière. [12] Ces rapports sont collectés au niveau d'un nœud spécial dans le réseau (nœud puits) et envoyés aux chefs de troupes par la suite. Ces rapports peuvent être également renvoyés aux responsables hiérarchiques d'ordre supérieur après avoir été synthétisés avec d'autres informations venant d'autres unités.

- **Reconnaissance et surveillance du champ de bataille**

Les réseaux de capteurs, peuvent être utilisés également pour couvrir tous les terrains critiques, les chemins et les détroits, afin de surveiller de près toutes les activités des forces ennemies.[47] De plus, de nouveaux réseaux de capteurs peuvent être facilement déployés sur les terrains pour répondre à l'évolution des opérations et l'application des nouveaux plans.

- **Ciblage**

Une autre application importante des réseaux de capteurs dans le domaine militaire consiste à leur incorporation dans les systèmes de guidage des munitions intelligentes.

- **Estimation des dégâts**

Juste après ou avant une opération militaire (attaque, bataille), des réseaux de capteurs peuvent être déployés sur le terrain ciblé pour permettre la collecte des informations liées au recensement et à l'estimation des dégâts.

- **Détection et reconnaissance des attaques nucléaires, biologiques et chimiques**

Lors d'une guerre biologique ou chimique, la détection précise et instantanée des agents chimiques pouvant être utilisés par les forces opposées est très importante. Dès lors, des réseaux composés de nœuds capteurs de réactions chimiques et biologiques peuvent être employés comme des systèmes d'alerte, fournissant ainsi aux utilisateurs du réseau un temps d'intervention critique qui leur permet de diminuer le risque de danger induit par ce genre d'attaques. [1] On peut aussi utiliser ces réseaux pour une reconnaissance détaillée de l'attaque détectée sans exposer aucune personne aux dangers de cette attaque (dangers de radiation dans les attaques nucléaires par exemple).

- **Applications dans l'environnement**

Diverses, sont les applications d'environnement qui peuvent bénéficier de la technologie des réseaux de capteurs sans fil, on peut citer par exemple, le cheminement des mouvement d'oiseaux, des petits animaux et des insectes ; le contrôle des aspects environnementaux qui peuvent affecter les récoltes et le bétail ; l'irrigation ; les macro-instruments utilisés pour la surveillances des terrains à grande échelle et les explorations planétaires[2] [4] ; la détection chimique et biologique ; la détection des incendies dans les forêts ; les travaux de recherche météorologiques et géophysiques ; la détection des inondations ; les études de pollution et la schématisation de la bio-complexité de l'environnement[6].

- **Détection des incendies de forêts**

Comme les capteurs peuvent être déployés d'une façon aléatoire, et dense dans n'importe quel type de forêt, ils peuvent facilement détecter et rapporter l'origine de l'incendie à l'utilisateur

avant qu'il se propage et devienne incontrôlable. Des millions de nœuds capteurs utilisant les fréquences radio ou la communication par voix optique peuvent être déployés et intégrés, ces nœuds doivent cependant pouvoir fonctionner d'une manière autonome pendant une longue durée qui peut aller jusqu'à des années, pour cela ils doivent être équipés par des systèmes de rechargement d'énergie efficaces tels que les panneaux solaires.[44] De plus, et pour faire une surveillance distribuée, ils doivent pouvoir collaborer et surmonter tous les obstacles qui existent tel que les rochers et les arbres.

- **Schématisation de la bio-complexité de l'environnement**

Cette opération requière des approches sophistiquées pour intégrer des informations précises à travers des échelles spatiales et temporelles [27]. En effet, les avancées technologiques dans le domaine de capture distante et de collection automatique de données ont permis une résolution temporelle, spectrale et spatiale plus élevée, tout en réduisant le coût de l'opération de schématisation par unité de surface. Grâce à ces avancées également, les capteurs ont la possibilité d'être connectés à Internet pour permettre aux utilisateurs distants d'observer et surveiller constamment la bio-complexité de l'environnement. [18] Bien que les systèmes de capture traditionnels basés généralement sur les images satellitaires soient utiles pour observer la biodiversité à grande échelle, ils n'ont pas une finesse de granularité suffisante pour observer la biodiversité à petites échelles qui constitue la grande part de la biodiversité dans l'écosystème. Pour cela, il s'avère nécessaire d'utiliser des capteurs sans fil pour une meilleure description de cette bio-complexité.

- **Détection des inondations**

Parmi les exemples d'utilisation des réseaux de capteurs pour la détection des inondations on trouve le système ALERT (Automated Local Evaluation in Real Time) [50] déployé aux états unis. Ce dernier englobe plusieurs types de capteurs hydrologiques qui détectent la pluie, le niveau d'eau, ainsi que d'autres capteurs météorologiques qui servent à la détection de la température, la pression, etc. tous ces types de capteurs fournissent les informations nécessaires à une base de données centralisée via une communication radio, un modèle de prévision des inondations est adopté pour analyser les données reçues et générer les avertissements éventuels.

- **Agriculture**

Les réseaux de capteurs sont capables d'apporter des bénéfices considérables au domaine d'agriculture, grâce à leur habilité de surveiller les taux de pesticides dans l'eau potable, le degré d'érosion du sol, et le niveau de pollution de l'air en temps réel.

- **Applications dans le domaine médical**

Les réseaux de capteurs peuvent être très utiles dans le domaine médical, ils peuvent par exemple fournir des interfaces d'aides aux handicapés, permettre la surveillance intégrée des patients, le diagnostic ; l'administration des médicaments au sein de l'hôpital ; le contrôle des mouvements et processus internes des insectes ou des autres petits animaux ; le contrôle des médecins et des patients dans l'hôpital ; ainsi que la télésurveillance et la collecte des informations physiologiques humaines.

- **Télésurveillance des informations physiologiques humaines**

En utilisant les réseaux de capteurs sans fil, cette opération fournit des informations qui peuvent être stockées pendant une longue période, et utilisées par la suite pour des fins multiples, notamment les explorations médicales. Les réseaux de capteurs installés peuvent également détecter et surveiller le comportement des personnes âgées et permettre une intervention rapide en cas de nécessité. Par conséquent, ces petits nœuds capteurs permettront aux sujets surveillés une meilleure liberté de mouvements, ainsi qu'aux médecins une identification plus rapide de certains symptômes prédéfinis. De plus, ils assurent aux sujets une meilleure qualité de vie comparée à celle dans les centres hospitaliers.

Loren & al. [44] décrivent une application biomédicale qui utilise les réseaux de capteurs et consiste à construire une rétine artificielle servant à aider les personnes malvoyantes, cette rétine contient une centaine de micro-capteurs munis d'une communication sans fil employée pour le contrôle, l'identification et la validation de l'image captée.

- **Le Suivi et la surveillance des médecins et des patients au sein de l'hôpital**

Chaque patient aura des petits nœuds capteurs légers qui lui sont attachés, où chacun de ces nœuds aura sa tâche spécifique, par exemple, l'un des capteurs peut être employé pour détecter les battements de cœur et un autre pour la pression du sang, etc,... Les médecins aussi, peuvent porter des capteurs permettant à d'autres médecins de les localiser.

- **L'administration des médicaments**

Si les nœuds capteurs peuvent être attachés aux médicaments, cela permettra de minimiser la probabilité de prescrire des mauvais traitements aux patients, car ces derniers auront, de la même manière, des capteurs qui identifient leurs allergies et les médicaments qui leur sont adéquats. Des systèmes informatisés tels que décrits dans [10] peuvent aider à diminuer les possibilités d'effets secondaires causés par les médicaments inadéquats

- **La surveillance des Habitats** L'utilisation des réseaux de capteurs dans ce champ de recherches a permis aux scientifiques d'étudier les comportements des animaux, oiseaux, ou

poissons dans leurs milieux, où il était presque impossible avant de le faire. Ce genre de recherches a un grand intérêt pour l'humanité, parce qu'il permet de surveiller et protéger les différentes espèces menacées d'extinction. Ce domaine peut apporter aussi un grand intérêt aux sociologues. Équiper des personnes avec des capteurs permet d'étudier les interactions humaines et les comportements sociaux.

- **L'industrie**

Les réseaux de capteurs sont largement utilisés dans le champ industriel, soit dans les chaînes de production, pour la sécurité, ou même dans les produits.

- **Contrôle et maintenances dans les équipements industriels**

Les robots industriels complexes sont équipés de centaines de capteurs qui sont généralement connectés par des câbles à un ordinateur principal qui assure le contrôle. Puisque les câbles sont plus chers et plus gênants pour les mouvements des robots, ils sont remplacé par des connections sans fil.

- **La sécurité industrielle**

Les grandes usines de nos jours sont équipées par des appareils très complexes, parfois des équipements pour des produits chimiques qui sont dans la plupart du temps toxiques, donc la moindre mauvaise manipulation peut conduire à une catastrophe. Déployé un réseau de capteurs dans un tel environnement permet de détecter et de réagir rapidement à un tel incendie.

- **Les moyens de transport**

La plupart des moyens de transport de nos jours (véhicule, avion, ...) sont équipés de dizaines même de centaines de capteurs/actionneurs reliés entre eux en réseau communiquant avec des stations de calcul, des bases de données, ou même avec des satellites pour améliorer la sécurité et l'efficacité du trafic.

- **Maison bureau et vie civile**

Les maisons et les bureaux de travail peuvent être équipés de dizaines de capteurs qui surveillent la température et l'humidité pour contrôler la climatisation et le chauffage. Les PCs peuvent être équipés de claviers et souris sans fil et peuvent être relié entre eux en réseau ad hoc pour faire des travaux collaboratifs et collecter des informations vers une station de calculs pour accomplir un traitement centralisé. [33]Des réseaux de capteurs peuvent être déployés dans les différentes structures de la ville: ponts, bâtiments, réservoirs d'eau et d'énergie (électricité, carburant, gaz), afin de les surveiller et détecter le moindre danger le plus tôt possible. Un réseau de capteurs de mouvement peut constituer un système d'alarme

distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne serait plus aussi simple, puisque il n'existe pas de point critique. Utiliser des réseaux de capteurs dans les routes et les rails permet une meilleure planification urbaine.

### • **Détection et surveillance des vols de voitures**

Les nœuds capteurs peuvent être déployés pour identifier et détecter les menaces de vols dans une région géographique, et rapporter ces menaces à un utilisateur distant à travers Internet pour les analyser.

### • **Gestion et contrôle de l'inventaire**

Chaque article dans le magasin pourrait avoir un nœud capteur qui lui est attaché, dès lors, les utilisateurs pourront facilement localiser l'article et calculer la quantité exacte de chaque catégorie d'article. Si des utilisateurs veulent insérer des nouveaux inventaires, ils n'auront qu'attacher les capteurs appropriés à ces inventaires.

### • **Surveillance et dépistage des véhicules**

Deux approches ont été décrites dans [40] pour la détection et le dépistage des véhicules. La première consiste à déterminer localement la ligne de roulement de la voiture dans un secteur donné, et l'envoyer par la suite à la station de base. Par contre, la deuxième renvoi directement les données brutes collectées par les nœuds capteurs à la station de base pour localiser le véhicule.

## **7. La gestion des RCSFs**

L'objectif des réseaux de capteurs sans fil est de surveiller, de contrôler un environnement physique et d'atteindre des objectifs spécifiques.

La gestion du réseau est le processus de suivi et de contrôle du comportement d'un réseau. Ce dernier pose des défis uniques pour la gestion de réseau qui font des techniques de gestion de réseau traditionnelles impraticables. Dans les réseaux traditionnels les principaux objectifs sont :

- ✓ Minimiser le temps de réaction
- ✓ Fournir des informations complètes.

Mais dans les réseaux de capteurs, l'objectif principal est de minimiser la consommation d'énergie et le moyen principal pour y parvenir est de réduire la quantité de communication entre les nœuds. Optimiser les propriétés opérationnelles et fonctionnelles de RCSFs peut exiger une solution unique pour chaque problème d'application. Les RCSFs sont très dynamiques et sujettes à des défauts, principalement en raison de la pénurie d'énergie, les interruptions de connectivité et les obstacles environnementaux. [45]Les défaillances de

réseau sont des événements communs plutôt que des exceptionnelles. Ainsi, dans les réseaux de capteurs, nous sommes principalement concernés par la surveillance et le contrôle de la communication des nœuds afin de :

- Optimiser l'efficacité de réseau.
- Assurer que le réseau fonctionne correctement.
- Maintenir la performance du réseau.
- Contrôler un grand nombre de nœuds sans intervention humaine.

### **8. Critères de conception de systèmes de gestion de RCSFs**

L'observateur est une entité du réseau ou de l'utilisateur final qui veut avoir des informations sur les données recueillies par nœuds de capteurs. Selon le type d'application, l'observateur peut envoyer une requête et recevoir une réponse de celui-ci. L'élément capteur génère des données d'un phénomène donné. Un RCSF peut recueillir des données de capteurs différents, telles que la température, la pression, le champ électromagnétique, etc.[39] Les RCSFs ont d'autres caractéristiques importantes en fonction de l'application. Certains d'entre eux sont la couverture, l'exactitude, la fidélité, la densité, l'auto-organisation, l'adaptation et l'emplacement.

Un système de gestion de réseau conçu pour les réseaux de capteurs doit prendre en compte les propriétés uniques des réseaux de capteurs. Les critères suivants sont utilisés pour évaluer les systèmes de gestion de réseaux de capteurs :

- **Longévité/ énergie :** l'énergie est une ressource critique dans les RCSFs, toutes les opérations effectuées dans le réseau doivent être économiques en énergie. La disponibilité du réseau peut être économique en énergie. La disponibilité du réseau peut être mesurée comme la quantité de temps où tous les nœuds de capteurs dans le réseau continuent d'obtenir et de transmettre des données de détection, donc un système devrait être capable de courir sur des nœuds de capteurs sans consommer trop d'énergie ou d'interférer avec le fonctionnement des nœuds de capteurs. Une opération légère prolonge la durée de vie du réseau.
- **Latence :** Il s'agit de l'intervalle de temps entre l'instant où le capteur reçoit les données et le moment où ils sont livrés à la destination, et il a deux composantes : à l'intérieur du réseau, du capteur aux sink et du sink à observateur. Selon le type d'application et la latence du réseau, les données reçues par l'observateur peuvent être sans valeur et doivent être jetés.
- **Précision :** Celle-ci n'indique ni la fiabilité ni l'exactitude d'un résultat. Elle peut être définie comme la fraction de la validité des résultats obtenus.[47] Des facteurs tels que les

conditions de l'environnement, lorsque les données sont obtenues de la communication du nœud de capteurs peuvent aussi dégrader la précision.

L'application joue un rôle important dans cette mesure, car elle est chargée d'établir la quantité d'énergie à dépenser dans l'obtention des données. En conséquence, le réseau doit s'adapter à la mesure de la précision définie par l'application et en fonction d'une limite supérieure de temps de latence.

• **La tolérance de panne :** Dans un RCSF, les nœuds peuvent échouer à cause de l'énergie, de la destruction physique, des problèmes de la communication, ou de l'inactivité ( un nœud devient suspendu). Même si ces situations se produisent, elles peuvent être souhaitables que le réseau continue à fonctionner correctement. Un système de gestion doit être élastique à la dynamique de réseau par sa reconfiguration selon les besoins.

• **Goodput :** C'est le rapport du nombre total de paquets reçus par l'observateur et le nombre total de paquets envoyés par tous les capteurs pendant une période de temps.

• **Adaptabilité et réactivité :** Un système devrait être en mesure de récupérer et de s'adapter à l'état actuel du réseau ou à l'évolution de ses conditions, y compris des changements dans la topologie du réseau, au niveau d'énergie de nœud, dans la couverture et dans les limites d'exposition de réseaux de capteurs.

• **Minimale de stockage de données :** Un modèle de données utilisé pour représenter celles de gestion soit être extensible et capable d'accueillir les informations nécessaires pour effectuer les fonctions de gestion, mais doit également respecter les contraintes de mémoire de réseaux de capteurs.

• **Evolutivité :** Un système devrait fonctionner efficacement dans n'importe quelle taille de réseau.

### 8.1 Les fonctions principales de gestion des RCSFs

Un système de gestion de réseau conçu pour les capteurs sans fil devrait fournir un ensemble de fonctions de gestion qui intègrent la configuration,[47] l'exploitation, l'administration, la sécurité, et la maintenance de tous les éléments et les services d'un réseau de capteurs. La tâche principale de la surveillance de RCSF est de recueillir des informations sur les paramètres suivants :

- Etats de nœud (par exemple, le niveau de la batterie).
- La topologie du réseau.
- La bande passante sans fil.
- L'état des liens, de la couverture et des limites d'exposition de réseaux de capteurs.

Un système de gestion de réseau de capteurs peut effectuer une variété de tâches de contrôle de la gestion axées sur les états du réseau collectés tels que :

- Le contrôle fréquent d'échantillonnage.
- Nœud de communication on/off (gestion de l'alimentation)
- Contrôle de l'utilisation de la bande passante sans fil (gestion du trafic).
- Effectuer la reconfiguration du réseau pour récupérer les défauts des nœuds et de la communication (gestion des pannes).

Le suivi des nœuds individuels dans un grand réseau de capteurs peut être difficile. Il suffit de contrôler le réseau en assurant une couverture de réseau spécifique. En outre, les nœuds de détection sont généralement déployés dans des conditions sévères ou à distance et la configuration des nœuds dans les réseaux de capteurs change dynamiquement. Ainsi, un système de gestion de réseau de capteurs devrait permettre au réseau de s'auto-former, de s'auto-organiser, et idéalement, de s'auto-configurer en cas de défaillances sans connaissance préalable de la topologie du réseau.

Malgré l'importance de la gestion des RCSFs, il n'existe aucune solution généralisée existante pour elle. Cependant, la plupart des applications de réseaux de capteurs sont conçues avec la gestion de réseau à l'esprit et donc il n'y a pas de couche de gestion de réseau supplémentaire qui est nécessaire.[12]

Les fonctions de gestion peuvent être automatiques, lorsqu'elles sont exécutées par un logiciel invoqué comme des renseignements obtenus à partir d'un modèle, semi-automatiques, lorsqu'elles sont exécutées par un opérateur humain assisté par un système qui fournit un modèle de réseau ou invoqué par un système de gestion, et manuel, lorsqu'elles sont exécuté à l'extérieur du système de gestion.

Une liste partielle des fonctions de gestion, sans aucun ordre particulier, est donnée ci-dessous :

- ✓ La fonction de surveillance de l'environnement.
- ✓ La fonction de surveillance de la zone de couverture.
- ✓ La fonction de définition de déploiement des nœuds.
- ✓ La fonction d'acquisition des exigences environnementales.
- ✓ La fonction de configuration des paramètres de fonctionnement du réseau.
- ✓ La fonction de découverte de la connectivité du réseau.
- ✓ La fonction horaire de l'exploitation de gestion.
- ✓ La fonction de découverte de la synchronisation de coopération.

- ✓ La fonction de l'énergie par la carte de génération.
- ✓ La fonction d'interface de l'utilisateur.
- ✓ La fonction du nœud de découverte de la localisation.
- ✓ La fonction du nœud de l'exploitation du contrôle d'état.
- ✓ La fonction de découverte de niveau d'énergie.

Nous définissons certaines fonctions, énumérées ci-dessous, qui permettent d'obtenir des caractéristiques liées à l'efficacité d'un RCSF. Certaines de ces fonctions quantitatives sont définies pour obtenir des paramètres présentés dans :

- ✓ Régler le réseau en fonction de temps.
- ✓ Heure de départ du réseau.
- ✓ La fonction de la fréquence des mises à jour.
- ✓ La fonction de l'exigence de mémoire
- ✓ La fonction de l'évolutivité du réseau.
- ✓ La fonction de la consommation d'énergie.

### **9. Conclusion**

Dans ce chapitre, nous avons fait une étude sur les réseaux de capteurs sans fil. Nous avons posé les briques de base et fédéré les concepts nécessaires à la compréhension des problématiques abordées dans le reste de ce manuscrit. Parmi ces concepts, nous avons vu les contraintes associées aux réseaux de capteurs. Nous avons remarqué à travers nos lectures que la contrainte d'énergie est le cheval de bataille pour la conception de tous types de protocoles liés aux réseaux de capteurs sans fil. En effet, l'aspect énergie est une métrique de performance très importante. Nous avons évoqué aussi le concept de réseaux de capteurs mobiles, ce type de réseau n'est pas largement étudié dans la littérature comparant à son prédécesseur, les réseaux de capteurs statiques. La plupart des solutions proposées pour la diffusion et la couverture de zone dans les réseaux de capteurs s'appuient sur une vue à plat où tous les nœuds sont considérés comme égaux et doivent contribuer ensemble à la gestion du réseau pour qu'ils puissent accomplir ses tâches. Par ailleurs, les capteurs peuvent apparaître, disparaître ou se déplacer indépendamment les uns des autres.

Et les ressources limitées des capteurs en termes de calcul, de stockage et d'énergie, ainsi que le mode de gestion du médium de communication partagé (bande passante limitée, collisions,...). Ces deux concepts, l'énergie et l'auto-organisation vont être des concepts clé pour le reste de cette thèse.

## **Chapitre 2**

# **Les Techniques De Conservation D'énergie Dans Les Réseaux De Capteurs Sans Fil**

### **1. Introduction**

Durant ces dernières décennies, les réseaux de capteurs ont bouleversé le monde. Le besoin d'un suivi continu des phénomènes naturels et aussi la surveillance dans différents domaines, ont renforcé l'intérêt pour cette nouvelle ère de l'informatique embarquée. En revanche, les réseaux de capteurs souffrent de leurs fragilités et de leurs énergies limitées. Les nœuds capteurs sont alimentés par des batteries limitées en énergie. Par ailleurs, le remplacement des batteries n'est pas une solution envisageable pour ces derniers, soit à cause du déploiement aléatoire des capteurs, ou à cause de l'hostilité de l'environnement où ils sont placés. [48] Toutefois, la mort d'un ou plusieurs nœuds capteurs interrompt partiellement la communication dans le réseau. De ce fait, une partie des données collectées sera perdue ce qui en résulte de la mort partielle du réseau. Maximiser la durée de vie du réseau constitue l'un des défis majeurs. La maximisation de la durée de vie d'un réseau de capteurs revient à minimiser les différentes sources d'énergie. En fait, un nœud capteur consomme de l'énergie pour accomplir son objectif dans le réseau. Son rôle principal est la collecte, le traitement et la transmission d'un ensemble de grandeur physique sur l'environnement qui l'entoure. Ces trois opérations constituent les principaux facteurs de consommation d'énergie. A partir de là, réduire l'énergie consommée par un nœud capteur revient à optimiser ces trois tâches. Dans ce contexte, plusieurs techniques de conservation d'énergie ont été proposées dans la littérature. Dans ce présent, nous décrivons la problématique de la consommation d'énergie dans les réseaux de capteurs. Nous présenterons aussi un panorama des techniques de conservation d'énergie proposées dans la littérature.

### **2. Facteurs intervenants dans la consommation d'énergie**

Il existe des facteurs qui induisent une consommation inutile de l'énergie (surconsommation). Ces facteurs sont nombreux, ils peuvent être engendrés lors de la détection lorsque celle-ci est

mal gérée [49], ou lors de la communication. En effet, la communication est sujette à plusieurs facteurs qui surconsomment de l'énergie surtout coté MAC. Certains de ces facteurs sont les causes majeures de la perte d'énergie et ils ont été recensé dans [49][51].

**a) Etat du module radio**

Le module radio est le composant du nœud capteur le plus consommateur en énergie puisque c'est lui qui assure la communication entre les nœuds. Le module radio opère dans quatre modes de fonctionnement : idle, transmission, réception et sommeil.

- Dans l'état idle : 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.

- Dans l'état transmission : la radio transmet un paquet. - Dans l'état réception : la radio reçoit un paquet.

- Dans l'état sommeil : la radio est mise hors tension. Il a été observé que dans le cas de la plupart des radios que le mode idle induit une consommation d'énergie importante, presque égale à la consommation en mode réception suite à l'écoute inutile du canal de transmission [50]. Ainsi, il est plus judicieux d'éteindre complètement la radio que de la laisser en mode idle quand elle n'est pas utilisée ni pour la transmission ni pour la réception des données.

**b) Notion de durée de vie du réseau**

La durée de vie du réseau dépend fortement de la durée de vie nodale. Cette dernière représente une métrique d'évaluation des performances très importante aussi bien dans les réseaux de capteurs que les réseaux Ad Hoc [51]. La vie nodale correspond à la durée de vie d'un des nœuds du réseau. Cette durée de vie dépend essentiellement de deux facteurs : l'énergie consommée en fonction du temps et la quantité d'énergie qui lui reste. La durée de vie moyenne d'un nœud continuellement actif est calculée comme suit [50 ] :

$$E = \frac{e}{p} \quad \dots \dots \dots \text{Équation 2.1}$$

Où : P est la puissance de transmission et E est l'énergie résiduelle du nœud. Pour la durée de vie du réseau il existe deux définitions principales :

- Pour certains elle représente le temps écoulé jusqu'à qu'un premier nœud tombe en panne
- Pour d'autres c'est la durée jusqu'à ce qu'il reste au plus une certaine fraction de nœuds survivants dans le réseau [50]. Dans la suite de ce manuscrit nous utiliserons la première définition.

### **c) Formes de dissipation d'énergie**

Détecter les événements dans l'environnement capté, élaborer un traitement de données local et rapide, et transmettre les résultats à l'utilisateur sont les principales tâches d'un nœud capteur dans le réseau de capteurs. Les étapes de consommation d'énergie par ce nœud peuvent être, dès lors, divisées en trois phases : le captage, la communication et le traitement des données [53].

### **d) L'énergie de captage**

L'unité de détection a pour rôle la collecte de mesures physiques de l'environnement surveillé. Cette dernière dépense son énergie dans la réalisation des opérations suivantes [60]

- 1) l'échantillonnage de données et la conversion en un signal électrique,
- 2) le traitement du signal

3) la conversion du signal de l'analogique au numérique. La plupart des chercheurs partent de l'idée que l'énergie consommée par l'échantillonnage des données est négligeable comparant à la communication [60].

En revanche, dans certains cas, l'énergie dépensée pour l'échantillonnage peut être dans le même ordre de grandeur ou supérieure à l'énergie consommée lors de la communication. A titre d'exemple : l'utilisation d'un capteur gourmand en énergie, des capteurs actifs (tels que les sonars, les capteurs d'images) ou aussi dans le cas où le temps d'échantillonnage est long. Par ailleurs, L'énergie consommée au moment du captage varie suivant la nature de l'application. La complexité et la nature de l'événement à détecter, joue également un rôle crucial pour déterminer la quantité d'énergie consommée pour l'échantillonnage. Contrôler un événement sporadique consomme moins d'énergie qu'un contrôle d'événement constant. Aussi, les environnements contenant un niveau de bruit élevé entraînent l'augmentation de l'énergie nécessaire pour la capture.

### **e) L'énergie de traitement (calcul)**

Cette unité est constituée d'un microcontrôleur (microprocesseur) et une mémoire. L'énergie résiduelle de cette unité est dépensée lors de la commutation et aux fuites [60][52]. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite désigne l'énergie consommée par l'unité de calcul dans le cas où aucun traitement n'est effectué. Enfin il est à noter qu'un nœud peut contenir des circuits additionnels pour le codage/décodage des données, [57] en plus de certains circuits spécifiques aux applications du réseau, dans tous ces cas, la conception des algorithmes et protocoles du réseau est influencée largement par

l'énergie consommée par ces circuits. L'énergie de commutation  $E_{switch}$  est exprimée comme suit :

$$E_{switch} = C_{tot} V_{dd}^2 \quad \dots\dots \text{Équation 2.2}$$

Où :  $C_{tot}$  est la capacité totale commutée par le calcul,  $V_{dd}^2$  est l'alimentation du voltage.

**f) Accès au support de transmission**

La couche MAC a un rôle très important pour la minimisation de l'énergie consommée. Un protocole MAC économe en énergie essaie d'utiliser le moins souvent possible le module radio. Les modules radio peuvent avoir plusieurs niveaux de consommation quand ils ne sont pas en mode émission ou réception, moins le nœud consomme moins il est réactif, c'est pour cela que les différents états existent pour assurer une flexibilité selon le degré de réactivité demandé par la couche MAC.[53] L'utilisation inutile du module provient de six sources essentielles : la retransmission, l'écoute passive, l'écoute abusive, la surcharge, la surémission et la taille des paquets.

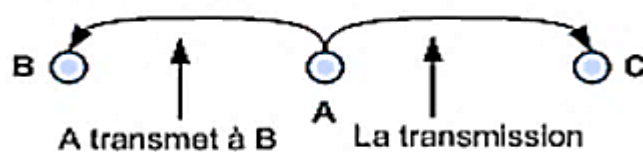
**g) La retransmission**

Les nœuds capteurs utilisent généralement une seule antenne radio, cependant ils partagent le même canal de transmission. La transmission simultanée des paquets par les nœuds voisins peut engendrer des collisions. Ainsi une quantité des données transmises sera perdue. La retransmission de ses données perdues générera une perte significative de l'énergie. L'écoute passive (idle listening).

**h) L'écoute passive du canal radio**

Dans l'attente d'une éventuelle réception (le mode idle décrit précédemment) engendre une perte importante des capacités des nœuds en termes d'énergie. Ceci est couteux et inutile dans le cas des réseaux à faible charge de trafic. De ce fait, basculer les nœuds capteurs en mode sommeil est une solution mais la transition entre les modes consomme également de l'énergie. Pour cette raison la fréquence de transition entre les modes doit rester raisonnable.

**i) L'écoute abusive (overhearing)**



**Figure 2.1** L'écoute abusive dans une transmission [63]

L'écoute abusive se produit quand un nœud reçoit des paquets qui ne lui sont pas destinés (figure 2.1). Le coût de l'écoute abusive peut être important dans le cas d'un réseau dense et avec une charge de trafic importante.

### **j) La surcharge (L'overhead)**

Plusieurs protocoles de la couche MAC, utilisent les paquets de contrôle pour maintenir une bonne communication entre les nœuds (signalisation, connectivité, établissement de plans d'accès et évitement de collisions). L'échange de paquets nécessite une énergie additionnelle. Par ailleurs, comme ces paquets ne transportent pas directement des données, ils réduisent également le débit utile effectif.

#### **- La surémission (Overemitting)**

Le phénomène de surémission est produit quand des nœuds envoient des messages à des destinations qui ne sont pas prêtes à les recevoir, en effet ces messages sont considérés inutiles et consomment d'avantage de l'énergie.

#### **- La taille des paquets**

La taille des paquets a un effet sur la consommation d'énergie. En effet, si la taille des paquets est réduite, le nombre de paquets de contrôle échangés augmente ce qui génèrera un overhead. Dans le cas contraire, une taille grande des paquets nécessite l'utilisation d'une grande puissance de transmission.

### **3. Les méthodes de conservation d'énergie dans les réseaux de capteurs sans fils**

Comme on l'a déjà cité en introduction, les trois tâches principales, d'un nœud capteur, qui consomment de l'énergie sont: la capture, le traitement et la communication. La consommation d'énergie du module de détection dépend de la spécificité du capteur. Dans de nombreux cas, elle est négligeable par rapport à l'énergie consommée par le module de traitement et par-dessus tout, le module de communication. Dans d'autre cas, l'énergie dépensée par la détection peut être comparable ou supérieure à celle dépensée par la transmission. Plusieurs approches ont été proposées afin d'optimiser l'énergie au niveau de ces trois tâches. Par ailleurs, plusieurs classifications de ces dernières ont été proposées dans la littérature. Nous avons choisi la classification faite dans [37] comme montrée dans la (figure 2.2).

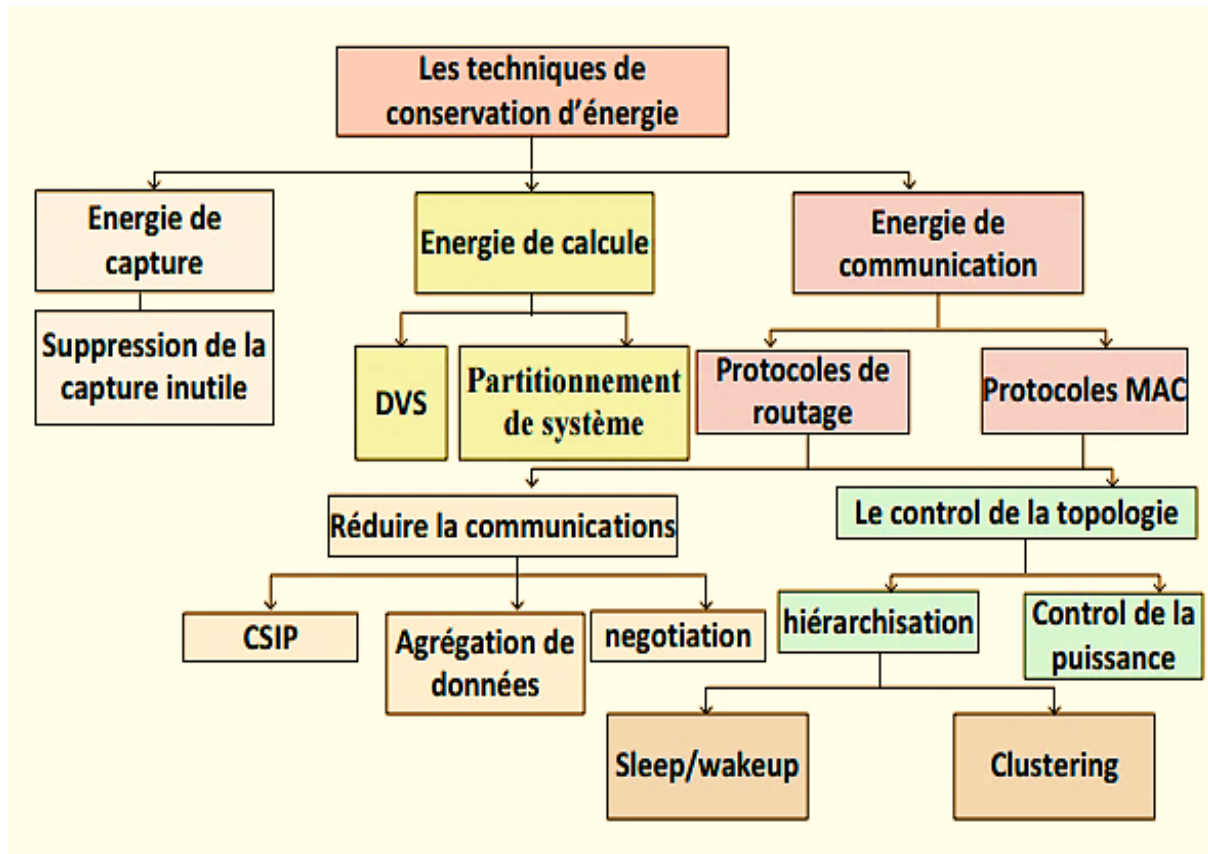


Figure 2.2 Les méthodes de conservation d'énergie dans les réseaux de capteurs sans fil [37]

### 3.1. Les méthodes proposées pour optimiser l'énergie de capture

Le module de détection affecte la consommation d'énergie de deux manières :

#### - L'échantillonnage inutile

Vu la densité du réseau de capteurs, les données échantillonnées ont souvent de fortes corrélations spatiales et/ou temporelles. Il est donc inutile de communiquer les informations redondantes à la station de base. En fait, même si le coût de la détection est négligeable, la communication des données redondantes implique une surconsommation d'énergie. - La consommation électrique du module de détection dans le cas des capteurs gourmands en énergie. Des techniques visant à réduire l'échantillonnage inutile ont été conçues pour minimiser l'énergie dépensée par le module de détection.

#### - Enlever l'échantillonnage inutile Selon la classification

Faite dans [37] il existe trois principales techniques : échantillonnage adaptatif, échantillonnage hiérarchique et l'échantillonnage actif fondé sur un modèle (Model-based active sampling).

1) *Echantillonnage adaptatif* Les approches proposées pour l'échantillonnage adaptatif exploitent les corrélations spatiales et temporelles des données pour réduire la quantité de données à acquérir par le module de détection. A titre d'exemple, les données intéressantes

peuvent changer lentement dans le temps, dans ce cas, des corrélations temporelles peuvent être exploitées pour réduire l'échantillonnage inutile. Tandis que, la corrélation spatiale peut être utilisée quand le phénomène étudié ne change pas brusquement entre les régions couvertes par des nœuds voisins.[53][54] Il est évident, les corrélations spatiales et temporelles peuvent être exploitées conjointement, pour réduire efficacement la quantité de données à acquérir. Vu que ce type de technique utilise des corrélations spatiales et temporelles, elles sont plus appropriées pour des types de réseaux où les changements ne sont pas trop fréquents.

**2) Echantillonnage hiérarchique** Le but de ce type d'approches consiste à trouver un compromis entre la consommation d'énergie et la précision. Étant donné que les capteurs qui donnent de très bonne précision sont gourmands en énergie, l'idée est d'utiliser différents types de capteurs avec des caractéristiques de puissance différentes. Un exemple d'utilisation de ce type d'approche : la détection des voitures soupçonnées. Dans ce cas, des capteurs simples (tels que le magnétomètre à faible puissance) peuvent être utilisés pour détecter les véhicules. [55] Et pour avoir plus de précision, des capteurs plus puissants sont utilisés. L'échantillonnage hiérarchique est plus approprié aux applications qui nécessitent beaucoup de précision. Cette technique peut coupler entre la conservation d'énergie et la précision.

**3) l'échantillonnage actif** fondé sur un modèle (Model-based active sampling) Ces approches utilisent un mécanisme similaire à la prédiction de données. Un modèle du phénomène mesuré est construit à base des données échantillonnées. Ce modèle est ensuite utilisé pour prédire les données futures avec une certaine précision en réduisant ainsi le nombre d'échantillonnage. L'idée de base est la suivante : Le capteur

- Analyse la relation spatiotemporelle parmi les mesures effectuées par le capteur,
- Utilise cette connaissance pour optimiser l'échantillonnage. Les techniques basées sur l'utilisation de modèle permettent non seulement de réduire le nombre de données échantillonnées mais aussi de réduire le nombre de communication malgré que ce ne soit pas leur objectif principal. Cependant, dans ce type d'approche on ne peut pas évaluer les données avec certitudes car elles sont dérivées du modèle construit.

### **3.2.La méthode proposée pour optimiser l'énergie de calcul**

Comme mentionné précédemment, l'énergie de calcul est dépensée lors de la commutation et aux fuites (l'énergie de fuite désigne l'énergie consommée par l'unité de calcul dans le cas où aucun traitement n'est effectué) [37]. Cette dernière peut être optimisée en utilisant deux

techniques: technique d'adaptation dynamique de la vitesse et de la tension du processeur DVS (Dynamique Voltage Scalling) et technique de Partitionnement du système.

*a).1 Techniques d'adaptation dynamique de la vitesse et de la tension du processeur (DVS)* Comme mentionné précédemment, l'énergie dépensée par l'unité de calcul dépend très fortement de la tension d'alimentation (le voltage). De ce fait, minimiser la dépense énergétique du calcul est lié à la minimisation du voltage. C'est dans ce sens que le DSV a été proposé et déployé dans des microprocesseurs [38]. Le DVS permet aux processeurs de changer leur voltage et d'ajuster leur fréquence selon la demande de l'application sans pour autant dégrader les performances de ces derniers. De cette manière, la technique DVS aide les capteurs à bien conserver leurs énergies. L'idée de base consiste à régler la tension d'alimentation de sorte que le processeur fonctionne à un rythme long quand la charge du travail du processeur est faible [63]. Par contre, si la charge du travail est grande, alors le DVS contrôle le processeur pour qu'il travaille à une grande vitesse. Cette technique est appliquée dans des systèmes mobiles et des dispositifs embarqués.

- **Une autre approche appelée CO-DVS conçue** pour les systèmes temps réel a été proposée par Wei Li et al. dans [63]. En général, l'approche DVS utilisée dans les systèmes temps réel vise à réduire l'énergie au niveau de chaque nœud individuellement ; en ignorant d'autre contraintes comme le délai de bout-en-bout (end-to-end deadline) de l'application ou/et la consommation d'énergie au niveau des voisins. Le but principal de l'approche CO-DVS consiste à permettre aux nœuds du réseau (la station de base et les autres nœuds du réseau) de coordonner leurs DVS pour réduire l'énergie consommée et la latence du système. La technique du CO-DVS est destinée pour les applications temps réel où la notion de deadline et d'opportunité (timeliness) est très importante. Dans ce type d'application, en plus de la contrainte d'énergie nous trouvons la contrainte de latence. Cette technique a permis d'optimiser l'énergie totale dépensée car elle prend en considération l'énergie résiduelle des nœuds et aussi la latence du système en utilisant la coordination entre les différent CPU3 [63]. Cette technique a été testée dans deux types de réseau homogène et hétérogène (homogène utilise la même source d'énergie, et hétérogène chacun a sa propre source d'énergie). En appliquant la technique CO-DVS dans un système non coopératif, les résultats montrent que l'épuisement d'énergie d'un nœud critique (source) est de l'ordre de 51% plus rapide que d'un système coopératif. Généralement les approches proposées dans les systèmes temps réel ne sont pas directement applicable pour les systèmes d'exploitations (OS) proposés car les caractéristiques de la charge de travail doivent être précisément spécifiées au

ordonnanceur (la périodicité, deadlines...) d'autres approches ont été proposées où tous ces paramètres sont définies automatiquement en se basant sur le ratio du temps d'inactivité et le temps de disponibilité dans le processeur. Krisztián Flautner et al., proposent dans [64] une approche (Automatic Performance Setting for Dynamic Voltage Scaling) qui vise à contrôler automatiquement le DVS implémenté dans les processeurs dans le but de contrôler la puissance sans dégrader la performance du processeur.

**a).2 Techniques de partitionnement du système** Les techniques de partitionnement du système peuvent être utilisées pour réduire l'énergie dépensée dans le sous-système de calcul. Deux techniques sont utilisées [37], la première consiste à transférer un calcul prohibitif en temps de calcul vers une station de base qui n'a pas de contraintes énergétiques et qui possède une grande capacité de calcul. Tandis que, la deuxième partage le calcul entre les nœuds au lieu de placer des centres de calcul. Dans [38] les auteurs : K. Flautner, R. Uhlig, S. Reinhardt and T. Mudge [64] proposent un partitionnement du calcul dans un réseau de capteurs distribués organisés en clusters. Ce mécanisme a été implémenté et testé dans une application de localisation par estimation de ligne de Bering (LOB) pour des sources acoustiques. Les étapes de l'estimation de la ligne de Bering LOB sont résumées comme suit :

- Transformer les données collectées par les capteurs en fréquences en utilisant 1024 pt. FFT (transformé de Fourier). La transformée de Fourier rapide (sigle anglais : FFT ou Fast Fourier Transform) est un algorithme de calcul de la transformée de Fourier discrète (TFD). Cet algorithme est couramment utilisé en traitement numérique du signal pour transformer des données discrètes du domaine temporel dans le domaine fréquentiel, en particulier dans les analyseurs de spectre. Son efficacité permet de réaliser des filtres en modifiant le spectre et en utilisant la transformée inverse.

- Après le coefficient FFT est utilisé par l'algorithme de beamforming et l'estimation de ligne de Bering (LOB) pour calculer le résultat final. A la fin, ce résultat sera envoyé à l'utilisateur final. Dans un système non partitionné ce calcul s'effectue au niveau du cluster-head, qui est le capteur responsable de la collecte et l'envoi des données à l'utilisateur final. Ce calcul au niveau d'un seul nœud génère une latence et l'épuisement rapide de son énergie. C'est la raison pour laquelle l'approche proposée vise à partager le calcul du coefficient FFT entre les capteurs. Chaque capteur calcule son FFT et l'envoi au cluster-head, ce dernier fusionne ces coefficients. Ensuite, le cluster-head envoie le résultat du LOB à la station de base. Au niveau de la station de base, plusieurs LOB peuvent être combinées afin de pouvoir calculer la position de la source. En revanche,

l'utilisation du partitionnement dans ce cas, n'optimise pas la consommation d'énergie car le partitionnement du calcul ne diminue pas la quantité de données qui nécessite d'être transmise.

### 3.3. Les méthodes proposées pour minimiser l'énergie de communication

Des mesures expérimentales ont montré que généralement c'est la transmission des données qui consomme le plus d'énergie. Comme cité précédemment cette dernière est influencée par plusieurs facteurs qui ont un lien très étroit avec la couche MAC et la couche réseau. De ce fait, la conception des protocoles de conservation d'énergie doit prendre en considération les contraintes liées à la couche MAC et réseau. La minimisation de l'énergie dépensée côté MAC et réseau peut être atteinte soit en réduisant le nombre d'émission/réception des messages ou par le contrôle de la topologie. Réduire le nombre de communication Réduire le nombre de communication signifie moins d'utilisation de la radio. Ainsi, la consommation d'énergie peut être optimisée d'une manière significative, la réduction du nombre de communication peut être achevée grâce à trois techniques : agrégation de données, Négociation et CSIP (*Collaborative Signal and Information Processing*).

**a) Agrégation de données** Les nœuds voisins dans les réseaux de capteurs, étant donné qu'ils sont très corrélés spatialement et temporellement, peuvent générer les mêmes données qui seront ensuite transférées à la station de base (sink), à un moment donné les nœuds intermédiaires peuvent avoir les mêmes données reçues des nœuds sources.[40] Pour éliminer cette redondance et minimiser ainsi la quantité de données transférées à la station de base, des techniques d'agrégation de données sont utilisées. Généralement les méthodes d'agrégation de données visent à éliminer la duplication des données, en les agrégeant au niveau des nœuds intermédiaires. [60] Ceci, minimisera l'énergie dépensée pour la transmission des données qui représente 70% de l'énergie consommée par le nœud capteur.

**b) Implosion** Dans l'inondation classique, les nœuds peuvent recevoir la même information de différents nœuds cela provoquera ce qu'on appelle l'implosion au niveau des nœuds.

**c) Chevauchement (Overlap)** Le chevauchement se produit dans le cas où les nœuds couvrent une région du chevauchement (*overlapping region*), et les nœuds collectent les mêmes données.

**d) L'ignorance des ressources (Resource blindness)** Généralement dans l'inondation classique les nœuds n'adaptent pas leurs activités en fonction de leurs taux d'énergie à un instant donné.

**d) Le contrôle de topologie** Pour le contrôle de topologie nous avons deux mécanismes : le contrôle de puissance et la hiérarchisation.

**e) Le contrôle de puissance** Les techniques de contrôle de puissance visent essentiellement à optimiser la puissance utilisée de la radio, elles permettent aux nœuds d'ajuster leurs puissances radio dans le but de minimiser la consommation d'énergie. Les techniques de contrôle de puissance n'ont pas seulement un effet sur la consommation énergétique des nœuds capteurs mais aussi influent la connectivité du réseau. En outre, ils doivent assurer une transmission avec succès des paquets de données vers une destination. Les protocoles ATPC (Adaptive Transmission Power for Wireless Sensor Networks) [38] LMA et LMN (Local Mean Algorithm and Local Mean of Neighbors Algorithm) [37] font partie de cette catégorie.

**f) Les méthodes du groupement (clustering)** Les Algorithmes du groupement (clustering) organisent le réseau en sous réseaux (clusters), plus homogènes selon une métrique ou une combinaison de métriques, formant ainsi une topologie virtuelle. Chaque cluster identifie un nœud particulier appelé cluster-head. Le cluster-head, permet de coordonner entre les nœuds membres de son cluster, d'agréger leurs données collectées et de les transmettre par la suite à la station de base. De ce fait, seulement les nœuds leader seront responsables de l'acheminement de l'information collectée vers la station de base. Minimisant ainsi l'énergie consommée par les nœuds capteurs. Il existe plusieurs méthodes de formation de clusters. La méthode la plus répandue. [39] s'exécute comme suit :

- Chaque nœud découvre son voisinage en utilisant des messages Hello.
- Chaque nœud décide, en se basant sur des informations locales sur la topologie, s'il va devenir cluster-head ou pas.
- Le nœud élu comme cluster-head diffuse un message dans son voisinage invitant ses voisins qui ne sont pas encore affiliés à un autre cluster de le rejoindre.

**g) L'élection du cluster-head** La phase d'élection du cluster-head appelée aussi la phase Set-up, utilise une métrique (le degré de connectivité, l'énergie restante, mobilité...) ou une combinaison de métriques pour choisir un nœud leader qui est le cluster-head.

**h) Communication intra-cluster et inter-cluster** Chaque cluster-head est chargé d'assurer une bonne communication au sein de son cluster. Par ailleurs, il doit assurer le routage entre les différents cluster-heads en maintenant les informations liées au routage. De plus, étant donné que les cluster-heads ne sont pas directement reliés, des nœuds passerelles doivent être choisis pour assurer la communication entre les cluster-heads.

i) **La maintenance des clusters** Une mise à jour des clusters est réalisée soit dans le but de s'adapter aux changements de la topologie (dans le cas où un cluster-head ou un membre du cluster migre d'un cluster à un autre) soit pour équilibrer la consommation d'énergie dans le réseau (si le cluster-head garde son statut le plus longtemps possible, sa batterie sera épuisée, et ainsi il perdra son rôle). Parmi les protocoles proposés dans cette catégorie : LEACH [60].

j) **Les régimes de rendez-vous programmés (Scheduled rendez-vous)** Les protocoles du rendez-vous programmés exigent que tous les nœuds voisins se réveillent en même temps. Le principal avantage de ces régimes est quand un nœud est réveillé, tous ses voisins seront réveillés aussi. Typiquement les nœuds voisins se réveillent suivant un ordonnancement de réveil et demeurent actifs durant un intervalle de temps bien précis pour l'échange de données ensuite ils se redorment jusqu'au prochain rendez-vous Les protocoles du rendez-vous programmés proposés dans la littérature diffèrent dans la manière dont les nœuds du réseau se réveil et dorment pendant leurs durées de vie. Le plus simple est l'utilisation d'un modèle entièrement synchronisé. Dans ce cas, tous les nœuds du réseau se réveillent en même temps, selon un schéma périodique (periodic pattern). Un régime de réveil entièrement synchronisé est également utilisé dans des protocoles MAC tels que SMAC [39]

- **S-MAC Le protocole S-MAC (Sensor-MAC)** est conçu pour les réseaux de capteurs sans fil pour fournir une méthode d'accès économe en énergie. Il a été conçu pour atteindre les objectifs suivants: minimiser l'énergie consommée pendant qu'il supporte un bon passage à l'échelle et éviter les collisions. Chaque nœud met en mode sommeil sa radio pendant un certain temps, après il se réveil pour vérifier si l'un de ses voisin va lui envoyer un paquet, ce protocole nécessite une synchronisation entre les nœuds, chaque nœuds a la liberté de choisir sa propre synchronisation, tous les nœuds échangent leurs synchronisations afin qu'ils se mettent d'accord sur les périodes du sommeil/réveil, (le protocole S-MAC est un protocole entièrement synchronisé cela signifie que tous les nœuds voisins se réveillent et dorment en même temps), les nœuds doivent mettre à jour, périodiquement, leurs ordonnancements. La mise à jour est faite par l'échange de paquet SYNC qui contiendra le ID du nœud expéditeur plus le temps de son prochain sommeil. Pour remédier aux problèmes de collision, les nœuds accèdent au médium en utilisant le CSMA/CA de IEEE 802.11 avec le mécanisme RTS/CTS (Request-To-Send / Clear-ToSend), les nœuds se mettent à l'état sommeil après qu'ils entendent le paquet RTS ou CTS. S-MAC apporte une amélioration par rapport au CSMA/CA de la norme 802.11 en termes d'économie d'énergie.

Cependant, les messages de synchronisation et les messages RTS/CTS génèrent une surcharge du réseau.

- **T-MAC Dans le protocole T-MAC (Timeout-MAC) [39]** chaque nœud se réveille périodiquement pour communiquer avec ses voisins après il dort. Les nœuds communiquent en utilisant le mécanisme RTS (Request-To-Send), CTS (Clear-To-Send), Data, et ACK (Acknowledgement) qui permet d'éliminer les collisions et aussi assurer une bonne communication des paquets. Dans la période active le nœud continue à écouter le canal et potentiellement transmet des données. La période active se termine quand aucun événement d'activation n'est produit durant une période de temps TA. T-MAC fait passer les nœuds en mode sommeil sans s'être assuré que ses voisins n'ont plus de données à lui envoyer. En effet, les données à envoyer du voisin ont pu être retardées à cause d'un échec d'accès au canal. Ce problème est appelé le sommeil prématuré dans [39]. Pour remédier à ce problème, T-MAC propose d'envoyer un FRTS (Future Request To Send) après la réception d'un CTS pour informer la destination de ne pas changer son mode en sommeil après la période de temps TA. Pour éviter une collision entre les données et le FRTS, le nœud qui devait envoyer après le CTS reporte sa transmission pour la durée d'envoi d'un FRTS. Une autre solution est de prendre l'initiative en transmettant un RTS à la réception d'un RTS. Au lieu de répondre par un CTS, le nœud peut décider d'envoyer un RTS pour éviter que le destinataire des messages de ce nœud dorme au bout de TA. Les résultats ont montré que T-MAC réduit la consommation énergétique jusqu'à 96% par rapport à CSMA/CA de 802.11, et que T-MAC consomme autant que S-MAC sous une forte charge du trafic. En revanche, T-MAC génère plus de trafic de contrôle que S-MAC. Dans SMAC et T-MAC, les nœuds choisissent leur période de réveil soit aléatoirement, soit de manière qui permet aux nœuds voisins de se réveiller en même temps. Cela induit des retards sur le délai de bout-en-bout.
- **D-MAC (Data gathering MAC) [39]** propose un séquençement des périodes d'activité qui favorise la collecte d'informations dans une topologie arborescente. Les périodes de réveil des nœuds sont décalées selon leurs positions dans l'arborescence de collecte de données. Les nœuds du même niveau sont synchronisés à se réveiller en même temps. Les fils d'un nœud accèdent au médium en même temps en utilisant des délais (appelés backoff) aléatoires pour éviter les collisions systématiques. De ce fait, D-MAC sous-estime la probabilité de collisions en supposant que les nœuds fils peuvent se détecter. DMAC suppose aussi que la durée nécessaire pour envoyer le trafic cumulé dans les nœuds les plus proches de la racine reste inférieure à la durée du cycle d'activité.

**k) Les régimes asynchrones**

Un protocole *sleep/wakeup asynchrone* permet à chaque nœud de se réveiller indépendamment des autres, en garantissant que les voisins se chevauchent toujours dans les périodes actives au sein d'un nombre de cycles spécifiés. Aucun échange d'information n'est alors nécessaire entre les nœuds. RAW (Random Asynchronous Wakeup) [40], AWP (Asynchronous Wakeup Protocol) sont des régimes *sleep/wakeup asynchrone*.

**4. Taxonomie des protocoles de routage**

Pendant la conception d'un protocole de routage pour un réseau de capteurs sans fil, il est nécessaire de prendre en compte l'énergie et les ressources limitées des capteurs, la qualité du canal sans fil, la possibilité de perdre des paquets et la latence [11].

L'efficacité en consommation d'énergie représente une métrique de performance significative qui influence directement la durée de vie du réseau en entier. C'est pourquoi, les concepteurs de protocole de routage peuvent négliger les autres métriques de performance telles que les délais de transmission de données au détriment du facteur de consommation d'énergie. Le but d'un protocole de routage est de trouver les routes vers la station de base qui optimisent la métrique : consommation d'énergie.

Plusieurs stratégies de routage ont été proposées pour les réseaux de capteurs sans fil. Leurs principes de fonctionnement diffèrent suivant la philosophie de la classe à laquelle ils appartiennent. Ces approches peuvent être classifiées suivant : la structure (architecture) du réseau, les fonctions des protocoles, l'établissement des routes et l'initiateur des communications. Récemment, les protocoles de routage conçus pour les RCSF ont été largement étudiés. Les méthodes employées peuvent être classifiées suivant plusieurs critères qui sont illustrés dans le tableau suivant.[13]

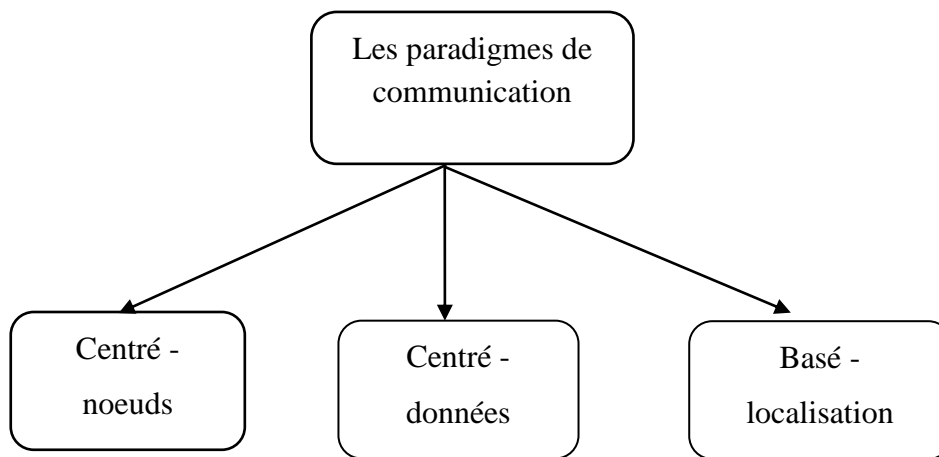
**4.1. Classification selon les paradigmes de communication**

Le paradigme de communication est déterminé par les contraintes sous lesquelles les nœuds du réseau sont interrogés. Dans les RCSF, il peut être classé comme étant Centré-nœuds, centré-données, basés-localisation. (figure2.3) Il existe également quelques protocoles basés sur la qualité de service, qui tentent de garantir certaines exigences des applications au moment du routage.

**4.1.1. Centré-nœuds**

Ce modèle est utilisé dans les réseaux conventionnels où il est important de connaître les nœuds communicants. Cependant, ce paradigme ne reflète pas la vision des RCSFs quant à leurs applications où la donnée transmise est plus importante que l'émetteur. Néanmoins, le

paradigme centré-nœuds n'est pas totalement écarté, car certaines applications nécessitent une interrogation individuelle des nœuds [10].



**Figure 2.3** Taxonomie des protocoles de routage selon les paradigmes de communication [39].

#### **4.1.2. Centré-données**

Ce modèle est utilisé dans les réseaux où il n'existe pas un système d'identification global, et cela dans presque toutes les applications des RCSFs. En effet, il n'est généralement pas possible d'attribuer les identifiants globaux (comme les adresses IP) pour chaque nœud à cause de nombre élevé de capteurs déployés. Ainsi, l'identité de chaque nœud n'est pas aussi importante que les données qui lui sont associées. Ce manque d'identification, avec le déploiement aléatoire des nœuds, font qu'il est difficile de sélectionner un ensemble de nœuds pour être interrogé. Par conséquent, les données sont généralement transmises de chaque nœud avec un taux important de redondances à l'intérieur de la région de déploiement. Ainsi, des protocoles de routage centrés-données ont été proposés pour être en mesure de sélectionner un bon ensemble de nœuds demandés, sans l'utilisation d'identifiants globaux. Ils visent également à utiliser l'agrégation de données pour éviter le gaspillage d'énergie dû aux redondances de données [11].

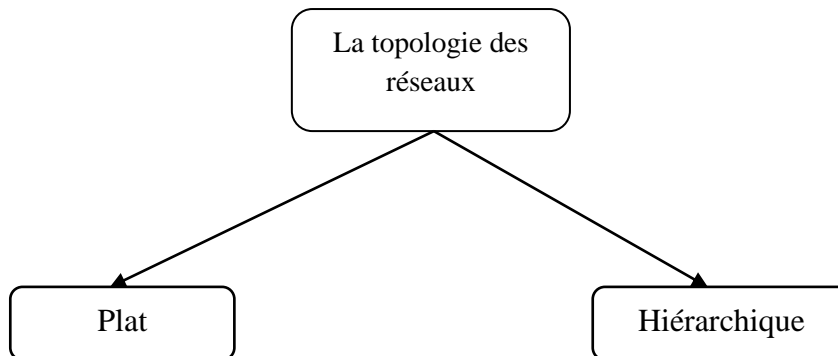
#### **4.1.3. Basé-localisation**

Ce paradigme est utilisé dans les applications où il est plus intéressant d'interroger le système en se basant sur la localisation des nœuds, et où on peut tirer profit des positions des nœuds pour prendre des décisions qui minimisent le nombre de messages transmis pendant le routage. Avant d'envoyer ses données à un nœud destination, le nœud source utilise un mécanisme pour déterminer sa localisation. Il est donc nécessaire de se pencher sur une solution de localisation géographique dont le degré de précision dépend de l'application visée.

Il existe deux techniques de localisation : absolue où on peut utiliser un système GPS (*Global Positioning System*) [9], ou, relative où les nœuds sont localisés approximativement suivant la direction ou la durée lors d'une transmission [10].

#### **4.2. Classification selon la topologie du réseau**

La topologie détermine l'organisation logique adaptée par les protocoles de routage afin d'exécuter les différentes opérations de découverte de routes et de transmission de données. Elle joue un rôle significatif dans le fonctionnement d'un protocole. La topologie peut être hiérarchique ou plate. (Figure2.4)



**Figure 2.4** Classification selon la topologie du réseau [39]

##### **4.2.1. Topologie plate**

Dans cette topologie tous les nœuds sont considérés homogènes et communiquent entre eux sans aucun autre intermédiaire, et seul le nœud puits est chargé de la collecte de données issues des différents nœuds capteurs afin de les transmettre vers les centres de traitement. Au cas où la destination (D) ne fait pas partie du voisinage de la source(S), les données seront transmises en utilisant des sauts multiples comme l'illustre la figure 2.5.

Les topologies plates sont caractérisées par la simplicité des algorithmes exécutés par les protocoles de routage. Et comme les RCSFs souffrent des changements brusques de la topologie, une organisation plate permet la possibilité de construire différents chemins des sources vers le nœud puits. Cependant, les réseaux plats présentent des inconvénients comme celui défini par le problème de *Hotspot*. En effet, tous les nœuds sont homogènes et il n'y a que le nœud puits qui est chargé de la récolte d'informations, et, ces dernières passent forcément par les nœuds qui entourent le nœud puits et qui seront de ce fait épuisés. Par ailleurs, les nœuds doivent accomplir plusieurs tâches en même temps ce qui pourrait rapidement épuiser leurs ressources énergétiques et dégrader ainsi les performances du réseau. Plusieurs protocoles rentrent dans cette catégorie comme: *Directed Diffusion* [13] et SAR (*Sequential Assignment Routing*).

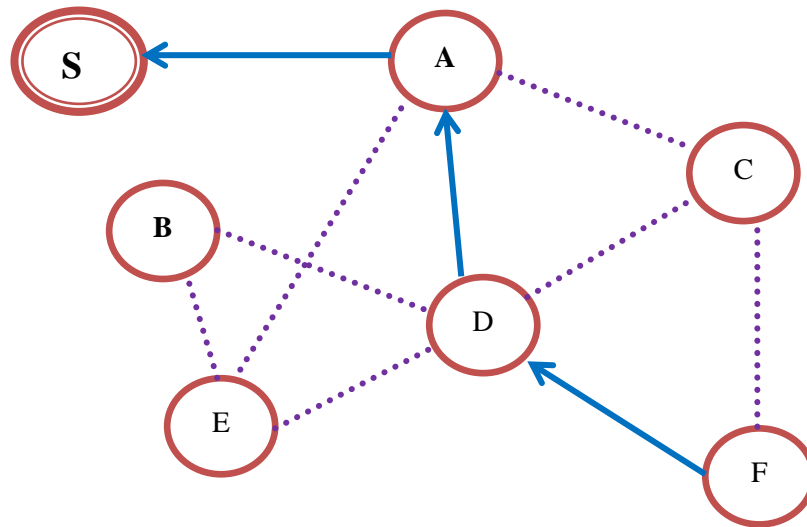


Figure 2.5 Routage plat [39]

#### 4.2.2. Topologie hiérarchique

Dans cette architecture il y a la notion de cluster : le réseau est subdivisé en clusters et chaque cluster est constitué de nœuds simples et d'un nœud leader (Figure 2.6). Seul le cluster-head (appelé aussi nœud leader ou encore nœud chef) communique avec les autres nœuds ou avec la station de base. Tous les nœuds d'un cluster envoient les données au cluster-head [9]. Les avantages de cette classe sont que la scalabilité et la consommation efficace de l'énergie sont assurés.

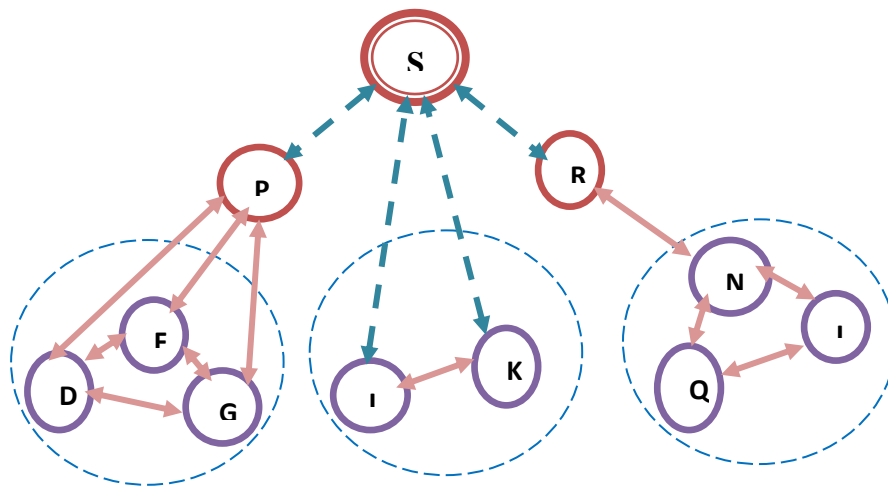


Figure 2.6 Routage hiérarchique [5]

Les nœuds leader sont soit des super nœuds (des nœuds plus puissants comparés aux nœuds simples, i.e. possède un processeur plus puissant une mémoire plus importante et des ressources d'énergie plus importantes) ou un simple nœud et alors ils sont élus périodiquement suivant leur quantité d'énergie restante (le nœud dont la quantité d'énergie restante est plus importante est élu cluster-head). Le cluster-head est le nœud responsable de la coordination des activités à l'intérieur du cluster et de la dissémination de l'information entre les clusters vers la station de base.

L'avantage du routage hiérarchique est que les données d'un ensemble vont être combinées par le cluster-head avant leur arrivée aux nœuds puits ce qui allège le travail de ce dernier, ainsi que celui des nœuds qui l'entourent. De plus, contrairement aux réseaux plats, un réseau hiérarchique possède une forte scalabilité. En effet, l'ajout des nœuds ne dégrade pas les performances du réseau car le réseau peut gérer seulement les nouveaux nœuds (par exemple, en les groupant et les associant à un cluster-head) sans qu'il affecte tous les nœuds restants du réseau. Cependant, les nœuds élus comme cluster-head consomment plus d'énergie que les autres nœuds. S'ils jouent toujours le rôle d'un cluster-head, ils vont être épuisés à un moment donné. Par conséquent, le réseau va être divisé, ce qui implique le découpage du réseau en secteurs inaccessibles. Parmi les protocoles qui utilisent cette topologie: LEACH (*Low-Energy Adaptive Clustering Hierarchy*).

### **a) Définitions**

- Qu'est-ce qu'un cluster ? : Un cluster est constitué d'un ensemble de machines (encore appelé nœuds) qui communiquent entre elles dans un réseau fermé. c-à-d un cluster est défini par un ensemble de nœuds et possède un nœud nommé nœud-chef ou Cluster Head (CH). Le rôle du CH est de faire le relais entre les nœuds du cluster et la station de base directement ou via d'autres CHs.
- **Qu'est-ce que le clustering?** C'est l'organisation de réseau en cluster, donc le regroupement des nœuds plus proche entre eux géographiquement en cluster. De manière générale, le partitionnement d'un réseau s'effectue sur le principe suivant: Les nœuds sont divisés en cluster, et certains nœuds, nommés chefs de cluster. L'ensemble des cluster-head est appelé l'ensemble dominant, c'est le backbone du réseau. Dans les réseaux de capteur le clustering permet de regrouper les capteurs proches géographiquement en clusters et d'utiliser différents schémas de routage au sein des clusters et entre les clusters. Les membres d'un cluster sont les nœuds se trouvant à la portée du cluster-head. Le choix de ce dernier se base sur une métrique spécifique ou une combinaison de plusieurs métriques.[14] En effet, l'élection des cluster-heads s'appuie sur différents critères et informations sur le réseau : le

niveau d'énergie d'un capteur, la connexion avec les autres capteurs, la position géographique, etc.

### **b) Communication des clusters**

#### **✓ Communications intra-cluster**

La transmission de données au sein d'un cluster peut se faire en utilisant des communications à un seul saut ou des communications multi-sauts.

##### **1. Communications à un seul saut**

Dans ce cas, chaque cluster-head se charge des communications à l'intérieur de son cluster. Chaque CH établit un plan de transmission pour les membres de son cluster, ce qui permet à l'antenne radio de chaque nœud non-CH d'être éteinte en dehors de son temps de transmission. En effet, le CH crée une table TDMA pour allouer un temps de transmission à chaque membre du cluster. Étant donné que chaque nœud connaît d'avance le slot de temps qu'il va occuper, il pourra se mettre à l'état 'sleep' en dehors du slot qui lui est réservé. La table TDMA ainsi créée, pour chaque cluster, sera diffusée à tous les membres du cluster.

##### **2. Communications multi-sauts**

Dans certains protocoles utilisant des communications à un seul saut, les nœuds les plus éloignés du cluster-head meurent plus rapidement que les nœuds les plus proches. Ceci a conduit à l'élaboration de protocoles de routage utilisant des communications multi-sauts à l'intérieur des clusters.[53] Dans ce cas, les nœuds communiquent uniquement avec leurs proches voisins et non pas directement avec leur CH ce qui économise davantage l'énergie. La tâche des nœuds membres ne se limite pas uniquement à la capture des données mais concerne aussi le routage des paquets et leur acheminement vers le cluster-head.[39]

#### **✓ Communications inter-cluster**

Nous pouvons distinguer deux techniques de communications pour l'acheminement des paquets des cluster-head vers la station de base.

##### **1. Transmissions à un seul saut**

La plupart des protocoles hiérarchiques ne traitent que le routage à un seul saut des cluster-heads vers la station de base. En effet, ils supposent que les CHs peuvent communiquer directement avec la SB.

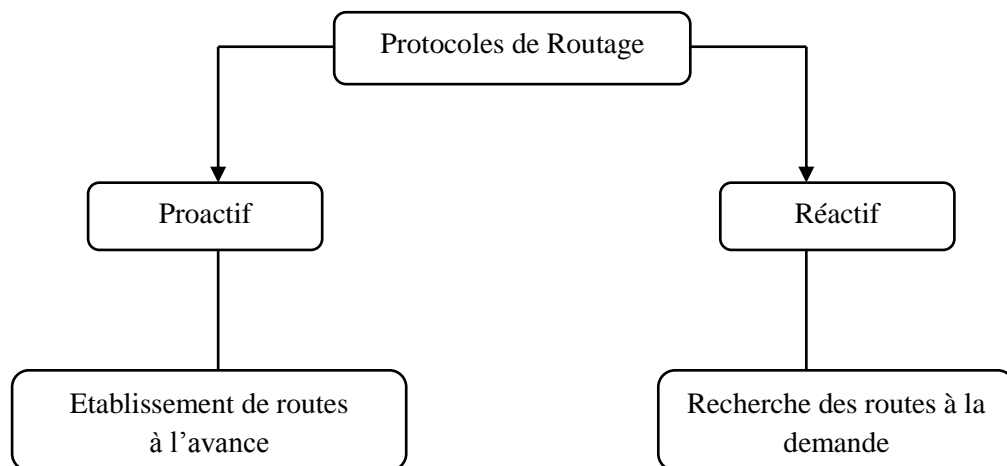
##### **2. Routage multi-sauts**

Afin de remédier au problème d'épuisement de l'énergie au niveau des CHs dans les réseaux à routage uni-saut entre les CHs et la station de base, d'autres protocoles ont été mis en place

utilisant un routage multi-sauts. Ces protocoles choisissent le chemin optimal entre le CH et la station de base à travers d'autres CHs considérés comme des relais.

### **4.3. Classification selon la méthode d'établissement de routes**

Suivant la manière de création et de maintenance de routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en catégories (figure 2.7) : protocoles proactifs, réactifs ou hybrides.



**Figure 2.7** Classification selon la méthode d'établissement de routes [39].

#### **4.3.1. Protocoles proactifs**

Les protocoles de routage proactifs établissent au préalable les meilleures routes pour chaque nœud vers toutes les destinations possibles. Ces protocoles maintiennent en permanence une vision globale de l'état du réseau grâce à une gestion périodique des tables de routage et l'échange des messages de contrôle. [56] [58] Ceci induit un contrôle excessif d'autant plus qu'ils sont particulièrement utilisés pour les réseaux denses. De plus, ils présentent un autre inconvénient dû à la sauvegarde des routes même si elles ne sont pas utilisées.

- Routage classique dans les architectures filaires
- Échange périodique de paquets (inondation)
- Routes maintenues vers toutes les destinations

#### **4.3.2. Protocoles réactifs**

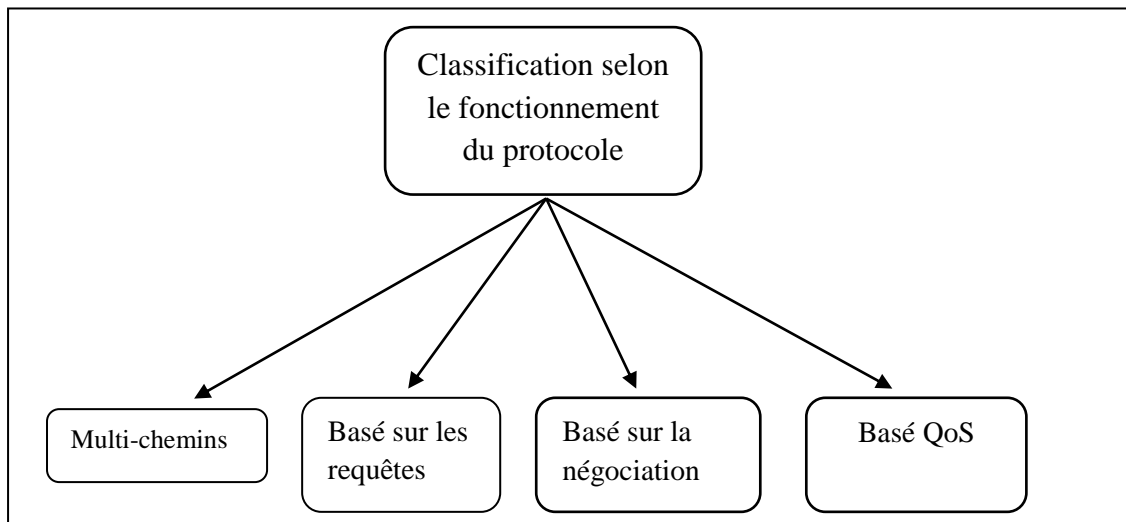
Les protocoles de routage réactifs maintiennent des routes à la demande. Lorsque le réseau a besoin d'une route, une procédure de découverte est lancée. Une fois la route n'est plus utilisée, elle sera immédiatement détruite ce qui permet une conservation d'énergie.

### 4.3.3. Protocoles hybrides

Les protocoles hybrides combinant entre les deux techniques précédentes utilisent des méthodes proactives pour l'établissement de la route dans le proche voisinage (par exemple le voisinage à deux ou trois sauts) et des méthodes réactives au-delà de la zone de voisinage.

### 4.4. Classification selon le mode de fonctionnement du protocole

Le mode de fonctionnement définit la manière avec laquelle les données sont propagées dans le réseau. (figure2.8) Selon ce critère, les réseaux de capteurs sans fil peuvent être regroupés en quatre catégories [13].



**Figure 2.8** Classification selon le mode de fonctionnement du protocole [13]

#### 4.4.1. Routage multi-chemins (*Multipath*)

Les protocoles basés sur ce type de routage utilisent des chemins multiples au lieu d'un seul dans le but d'assurer une bonne fiabilité et d'augmenter les performances du réseau. Des chemins alternatifs sont créés entre la source et la destination et sont maintenus grâce à l'envoi périodique de messages de contrôle. Malgré leur grande tolérance aux pannes, ces protocoles requièrent plus de ressources énergétiques et plus de messages de contrôle.

#### 4.4.2. Routage basé sur les requêtes

Dans ce type de protocole, le puits propage des requêtes vers les capteurs. Ces derniers ayant des données à transmettre, répondent en émettant les données via le chemin inverse des requêtes. Les deux protocoles : *Directed Diffusion* et *Rumour Routing* se basent sur ce principe.

#### 4.4.3. Routage basé sur la négociation

Les protocoles basés sur la négociation utilisent des descripteurs de données de haut niveau (métadonnées) afin de décrire la donnée avant de l'émettre. Ainsi, le récepteur éventuel peut décider de recevoir ou pas le paquet. [59] Cette procédure garantit que seules les informations

utiles seront transmises et élimine la redondance des données. Le protocole SPIN appartient à la classe des protocoles de routage basés sur la négociation.

#### **4.4.4. Routage basé-QoS**

Les protocoles de routage basés-QoS sont utilisés dans les applications qui ont des exigences temps-réel. Par exemple, dans le domaine de la sécurité, la détection d'intrusion doit être acheminée au plus bref délai vers le nœud puits. Ce type de protocoles essaye de répondre à quelques exigences de qualité de service (délai de transmission ou niveau de fiabilité) et doit faire l'équilibre avec la consommation d'énergie. [39] Le protocole SPEED est l'un des premiers protocoles géographiques basé sur la qualité de service.

### **Conclusion**

La complexité du routage dans les réseaux de capteurs sans fil amène les développeurs à prendre en considération les caractéristiques de ce type de réseaux ainsi que les exigences des applications auxquelles sont destinés ces réseaux. Les limites imposées par l'architecture matérielle des micros capteurs, en particulier celle de l'énergie, sont des facteurs primordiaux à prendre en compte durant la conception d'un protocole de routage. Dans ce chapitre, nous avons présenté différents facteurs influant sur la conception d'un protocole de routage, une étude détaillée des critères de classification et des différentes classes de protocoles de routage ainsi que les contraintes à prendre en considération lors de leur mise en œuvre.

Plusieurs protocoles de routage ont été proposés pour les réseaux de capteurs sans fil. Chaque protocole est adapté à une situation bien précise et doit tenir compte du type de l'application, du modèle de livraison de données, de la topologie du réseau, etc. L'objectif de la plupart des protocoles est d'optimiser la consommation d'énergie afin d'assurer une durée de vie plus longue au réseau. Cependant, certains protocoles visent à améliorer la qualité de service et à minimiser les temps de transmission.

Dans le chapitre suivant, nous nous intéressons au cas des protocoles de routage adaptés aux topologies hiérarchiques.

# Chapitre 3

## Les Réseaux Hiérarchiques et Clustering

### « Travaux Antérieurs »

#### 1. Introduction

Le routage est un élément clé de la communication du réseau. Il vise à garantir la livraison des messages à partir d'une source vers certaines destinations. Cela implique deux fonctions: (1) La découverte des routes à partir d'une source vers les destinations, et (2) la transmission des messages via les itinéraires découverts. Ces deux fonctions doivent être fixées afin d'assurer le bon fonctionnement du réseau. Dans les réseaux traditionnels, les fonctions de routage sont effectuées par des noeuds spéciaux, appelés routeurs qui sont souvent protégés physiquement. [58] Contrairement à cela, dans le réseau de capteurs sans fil, les capteurs eux-mêmes sont des noeuds effectuent les fonctions de routage. Cependant, en raison des faits que les réseaux de capteurs fonctionnent souvent dans un environnement sans surveillance et les nœuds de capteurs ne sont généralement pas inviolable, les nœuds de capteurs peuvent être facilement compromis. Ainsi, les nœuds ne peuvent pas être supposés exécuter le protocole de routage fidèlement, quelle tâche. En plus d'assurer la livraison de message, des protocoles de routage pour les réseaux de capteurs ont des objectifs supplémentaires. En particulier, certains protocoles sont concernés par les exigences en temps réel et visent à réduire au minimum le délai de livraison de message, tandis que d'autres essaient de maximiser la durée de vie du réseau en minimisant le temps de livraison de message, tandis que d'autres essaient de maximiser la durée de vie du réseau en réduisant au minimum et à équilibrer la consommation d'énergie des nœuds. Ces objectifs peuvent être satisfaits en utilisant différentes techniques de routage.

#### 2. Facteurs de conception de protocoles de routage

La mise en œuvre d'algorithmes de routage pour assurer la connexion des réseaux de capteurs sans fil doit tenir compte de certaines contraintes. La contrainte énergétique étant la plus

importante, un protocole de routage pour réseaux de capteurs doit assurer un prolongement de la durée de vie du réseau ainsi que le maintien de sa connectivité. Cependant, d'autres facteurs doivent être pris en considération, parmi lesquels nous citons [06] :

### **2.1. Le déploiement des nœuds**

Le déploiement des nœuds est spécifique à une application et peut affecter les performances du protocole de routage. Il peut être déterministe ou aléatoire. Dans le cas d'un déploiement déterministe, les nœuds sont placés manuellement et les données sont routées suivant des chemins prédéterminés. Dans le cas d'un déploiement aléatoire, [39] les routes sont construites dynamiquement. Ainsi certaines régions du champ de captage peuvent bénéficier d'une meilleure connectivité par rapport à d'autres et les données captées dans ces régions peuvent être routées plus facilement.

### **2.2 Le modèle de livraison de données**

Ce facteur définit la manière dont les données captées sont transmises à travers le réseau. Le modèle de transmission de données peut être classé dans l'une des catégories suivantes :

- a)* Basé sur le temps (Time-Driven) : les nœuds envoient les données périodiquement suivant un taux de transmission prédéterminé.
- b)* Basé sur les événements (Event-Driven) : les nœuds capteurs envoient les paquets de données lors de la détection d'un événement.
- c)* Basé sur les requêtes (Query-Driven) : les capteurs ne livrent les données que lorsqu'ils reçoivent une requête émise par le nœud puits.
- d)* Hybride : combinaison des trois modèles précédents.

### **2.3 La tolérance aux pannes**

La tolérance aux pannes dans un réseau de capteurs est la capacité de ce dernier à maintenir son bon fonctionnement malgré la présence de quelques défaillances. Ces défaillances peuvent survenir par manque d'énergie ou en raison de dommages physiques ou d'interférences environnementales. [47] En effet, la panne de quelques nœuds entraîne la perte des liens de communication et ainsi un changement significatif dans la topologie du réseau. Afin de maintenir le bon fonctionnement du réseau, les protocoles conçus doivent alors s'adapter à la nouvelle topologie du réseau en formant de nouvelles routes entre les nœuds.

### **2.4 L'hétérogénéité des nœuds**

Généralement, les nœuds d'un RCSF sont homogènes et disposent des mêmes capacités de calcul, de stockage et de ressources énergétiques. Cependant, selon les besoins des applications, les nœuds peuvent avoir des rôles différents. Ainsi, selon la tâche assignée au capteur, les besoins en ressources de calcul, de stockage, de communication et d'énergie peuvent varier d'un nœud à un autre. Pour remédier à ce problème, une solution envisagée par

certaines applications consiste à intégrer des nœuds spéciaux plus puissants que les autres et qui seront chargés d'effectuer les tâches les plus coûteuses en termes de ressources énergétiques. Cependant, l'intégration d'un ensemble de nœuds hétérogènes dans un seul réseau impose de nouvelles contraintes liées au routage de données. Par conséquent, la conception des protocoles de routage doit prendre en compte les différents types de nœuds, et les contraintes qui en résultent.

#### **2.5 La consommation d'énergie**

La consommation d'énergie s'effectue essentiellement au niveau de la capture, du traitement et de la transmission des données. Les études concernant les réseaux de capteurs ont montré que la phase de communication est la plus consommatrice en énergie. Vu que l'énergie des capteurs n'est pas renouvelable, les opérations de capture, de traitement et de routage doivent être correctement évaluées afin d'éviter l'épuisement prématuré de cette énergie et de prolonger la durée de vie du réseau. Les techniques de conservation d'énergie lors de la communication et le calcul sont donc d'une importance majeure dans un réseau de capteurs sans fil. L'efficacité énergétique représente alors une métrique de performance significative car elle influence directement la durée de vie du réseau. Pour cela, au moment du développement d'un protocole de routage, les autres métriques telles que le délai de bout en bout peuvent être négligées au détriment du facteur de consommation d'énergie. Les protocoles de routage doivent donc utiliser des mécanismes efficaces en consommation d'énergie en choisissant par exemple les chemins les moins consommateurs en énergie et en réduisant le nombre de messages de contrôle.

#### **2.6 La scalabilité**

Le nombre de nœuds dans les réseaux de capteurs est souvent élevé (des centaines de nœuds voire même des milliers). [39]Un protocole de routage doit assurer le bon fonctionnement du réseau quel que soit le nombre de nœuds.

#### **2.7 La connectivité**

La densité élevée des nœuds dans les réseaux de capteurs exclut la possibilité de l'isolement entre eux. Par conséquent, chaque nœud doit être fortement connecté pour éviter son isolation. Cependant, leur dispersement parfois aléatoire fait qu'ils ne sont pas uniformément répartis. Par conséquent, [39]certaines régions bénéficient d'une grande connectivité contrairement à d'autres. Les protocoles de routage mis en place doivent donc permettre l'auto-organisation des nœuds par rapport à la distribution aléatoire et à la topologie dynamique du réseau.

## **2.8 La mobilité**

La mobilité des nœuds et/ou de la station de base est parfois nécessaire dans les réseaux de capteurs sans fil. C'est le cas pour des applications de type 'detection/tracking' où la cible change de position d'une façon continue. Le routage dans de telles applications devient très difficile, car il faut acheminer des paquets de et vers des nœuds mobiles. Dans ces applications, la stabilité des chemins devient un défi aussi important que la conservation d'énergie.

## **2.9 L'agrégation des données**

L'agrégation de données est très importante dans les réseaux de capteurs sans fil. La densité importante des nœuds peut causer des redondances de données dans le réseau. Un nœud du réseau peut recevoir la même donnée à partir de plusieurs nœuds. La transmission de la même donnée par plusieurs capteurs implique alors une perte d'énergie inutile.[14] L'agrégation de données peut être réalisée en supprimant les paquets redondants ou en choisissant le maximum ou le minimum ou encore la moyenne des valeurs perçues. Lors de la conception d'un protocole de routage, il faut prévoir la ou les techniques d'agrégation et éventuellement choisir des nœuds concernés par ces opérations.

## **2.10 La qualité de service**

Certaines applications des réseaux de capteurs sans fil exigent que les données perçues doivent atteindre la station de base durant un temps limité pour que la donnée soit utile et acceptable. Dans ce cas, le temps de réponse constitue un paramètre très important dans le protocole de routage. Cependant, dans plusieurs applications, la conservation d'énergie, qui est directement liée à la durée de vie du réseau est considérée relativement plus importante que la qualité de service. Ainsi, le réseau peut être appelé à réduire la qualité de service des données envoyées afin d'augmenter la durée de vie du réseau.

## **3. Classification des algorithmes de clustering**

Les technologies de clustering pour les réseaux de capteurs proposées dans la littérature peuvent être généralement classifiées selon le modèle opérationnel global du réseau et l'objectif du regroupement des nœuds représenté par le nombre et les propriétés des groupes créés.

### **3.1. Modèle de réseau**

Différentes architectures et conceptions (objectifs/contraintes) ont été considérées pour différentes applications de RCSFs.

### **3.1.1. Dynamique d'un réseau**

Fondamentalement, les RCSFs se composent de trois composants principaux : nœuds de captage, la station de base et les événements à surveiller. Hormis les quelques installations qui utilisent les nœuds mobiles, la plupart des architectures de réseaux supposent que les nœuds sont stationnaires (fixes). [33] Parfois, il est jugé nécessaire de favoriser la mobilité de la station de base ou des cluster-heads (CHs). La mobilité des nœuds serait un challenge pour le clustering puisque les membres d'un cluster changeant dynamiquement obligeant ainsi les CHs à évoluer avec le temps. D'autre part, les événements surveillés par un nœud peuvent être intermittents ou continus selon l'application. Par exemple, dans une application de détection d'une cible, l'événement (phénomène) est dynamique tandis que la surveillance d'un feu de forêt est un événement intermittent. [38][46] La surveillance des événements intermittents permet au réseau de fonctionner en mode réactif, produisant du trafic seulement lors du reportage. Les événements continus dans la plupart des applications exigent un reportage périodique et produisent par conséquent du trafic significatif à router vers une station de base (sink). Bien que les événements continus rendent la plupart du temps les clusters stables, cela peut inégalement charger les CHs relativement aux nœuds dans le cluster et une rotation du rôle de CH peut être exigée si le CH est aléatoirement sélectionné de la population des capteurs. Les événements intermittents favoriseraient des stratégies de clustering adaptatives si le nombre d'événements varie de manière significative.

### **3.1.2. Traitement des données dans le réseau**

Puisque les nœuds pourraient produire des données redondantes significatives, des paquets semblables provenant de nœuds multiples peuvent être agrégés de sorte que le nombre de transmissions soit réduit. L'agrégation de données combine des données de sources différentes en employant des fonctions telles que la suppression (éliminant les duplications). Sachant que le calcul consomme moins d'énergie que la communication, l'agrégation de données permet ainsi d'économiser une énergie substantielle. Cette technique a été utilisée dans un certain nombre de protocoles de routages pour réaliser une économie énergétique et optimiser le trafic des messages. [20] Dans quelques architectures de réseau, toutes les fonctions d'agrégation sont assignées à des nœuds plus puissants et plus spécialisés. Il sera intuitif de s'attendre à ce que les CHs exécutent de telle agrégation de données qui peuvent limiter le choix d'un cluster-head seulement aux nœuds spécialisés ou exiger la limitation du nombre de nœuds par cluster afin de s'assurer que le ClusterHead ne soient pas surchargés. Il peut être parfois nécessaire de faire tourner le rôle de CH parmi les nœuds du cluster.

### **3.1.3. Déploiement et capacités des nœuds**

Une autre considération est le déploiement topologique des nœuds qui est dépendant de l'application et affecte le besoin et l'objectif du clustering de réseau. Le déploiement est soit déterministe ou auto organisationnel. Dans les situations déterminantes, les nœuds capteurs sont manuellement placés et les données routées par des chemins prédéterminés. Par conséquent, le clustering dans une telle situation est présélectionné ou inutile. Cependant dans les systèmes auto organisationnels, les nœuds sont aléatoirement dispersés créant ainsi une infrastructure de manière ad-hoc. Dans cette infrastructure, la position de la station de base ou du CH est également cruciale en termes d'efficacité énergétique et de performance. Quand la distribution des nœuds n'est pas uniforme, un clustering optimal devient une question présente pour assurer une consommation énergétique efficace dans un réseau. En outre, dans quelques installations des fonctionnalités différentes peuvent être associées aux nœuds déployés et le choix de CH peut être contraignant. [39] Dans les réseaux de nœuds homogènes, i.e. tous les nœuds ont des capacités égales en termes de calcul, communication et puissance, les CHs sont sélectionnés parmi les capteurs déployés. Souvent dans ce cas, les CHs sont soigneusement éprouvés, par exemple exclus des fonctions de captage, afin d'éviter d'épuiser rapidement leur énergie. De même, la portée de communication des nœuds est habituellement limitée et un CH ne peut pas atteindre directement la station de base. Même si un nœud peut le faire il vaut encore mieux utiliser un routage multi sauts. Par conséquent, la connectivité inter-CHs devient un facteur important qui affecte un schéma de clustering. (Figure 3.1) D'autre part, les WSNs hétérogènes peuvent imposer plus de contraintes au processus de clustering. Puisque certains nœuds peuvent être dédiés à des tâches spéciales ou être dotés de capacités spécifiques. [38] Il se peut alors qu'il faille éviter de tels nœuds afin qu'il puisse conserver leurs ressources ou de limiter le choix des CHs à un sous-ensemble de ces nœuds.

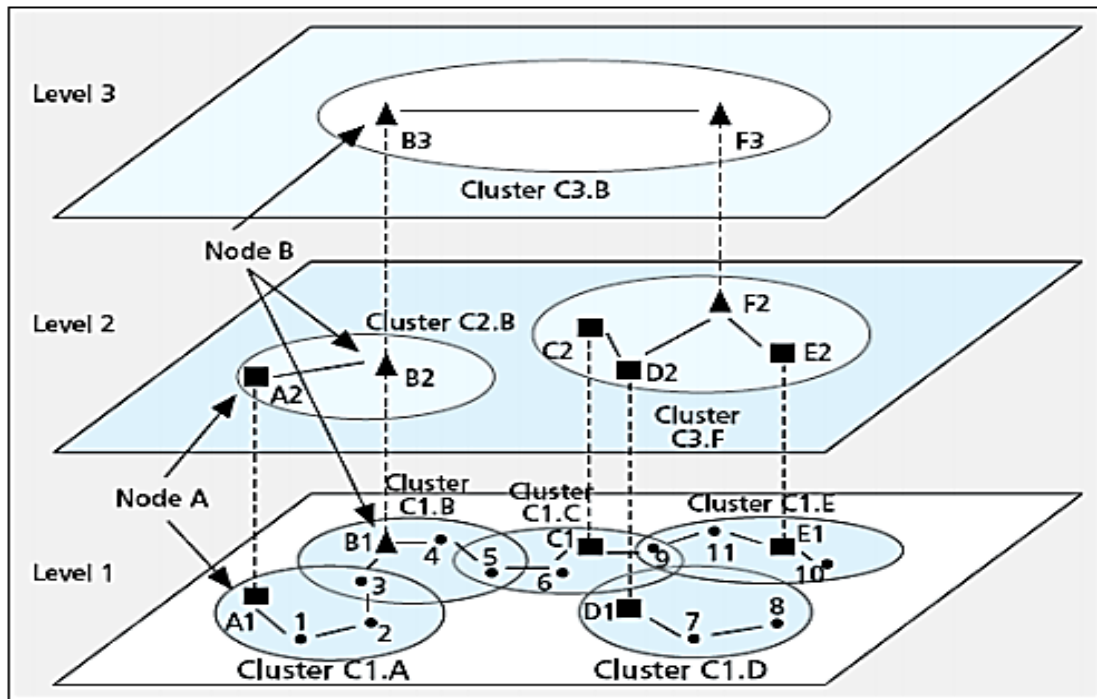


Figure 3.1 Topologie hiérarchique

### 3.2. Objectifs du clustering

Dans la littérature, les algorithmes de clustering proposés varient en fonction de leurs objectifs. Souvent l'objectif de clustering est défini afin de répondre aux exigences des applications.

#### 3.2.1. Equilibrage de la charge

La distribution des nœuds parmi les clusters est habituellement un objectif d'une organisation du réseau ou les CHs effectuent des fonctions de traitements de données ou d'activités significatives intra-cluster. Il est donc possible de répartir ces tâches entre cluster afin de réaliser un équilibrage de la charge à des fins d'objectif de performance attendu. Dans pareil cas, établir des clusters de même taille devient crucial pour éviter l'épuisement de batteries et prolonger la durée de vie de réseau.[61] De même, la distribution des nœuds peut également influencer la latence de disponibilité des données. Quand les CHs exécutent l'agrégation de données, il est impératif d'avoir un même nombre de nœuds dans les clusters de sorte que les données rapportées par les nœuds deviennent disponibles presque en même temps pour un traitement ultérieur au niveau de la station de base

#### 3.2.2. Tolérance aux fautes

Dans beaucoup d'applications, les WSNs sont opérationnels dans des environnements précaires et les nœuds sont couramment exposés aux risques de dommages physiques et de disfonctionnement. Tolérer une défaillance de CHs est nécessaire dans de telles applications

afin d'éviter la perte de données importantes captées par les nœuds. La manière la plus intuitive de recouvrer (survivre) à une défaillance de CH est re-clusteriser (reconfigurer) le réseau. Cependant, le re-clustering constitue, non seulement, un overhead en ressources sur les nœuds, mais il est souvent très perturbateur de l'opération en cours. Par conséquent, des techniques contemporaines et plus appropriées de tolérance aux fautes seraient d'un grand intérêt. [23]

Le choix d'un backup CH et le rôle d'un tel CH de réserve jouera pendant l'opération normale du réseau un rôle varié. Lorsque les CHs ont une longue portée radio, les CHs voisins peuvent adopter les nœuds du cluster défaillant. La rotation du rôle de CH parmi les nœuds dans le cluster peut également être un moyen d'assurer la tolérance aux fautes en plus de d'équilibrage de charge.

#### **3.2.3. Connectivité et délai réduit**

A moins que les CHs aient des possibilités de communications à longue portée, par exemple un lien satellite, la connectivité inter-CHs est une exigence importante dans beaucoup d'application.

C'est particulièrement vrai quand les CHs sont sélectionnés parmi les nœuds du réseau. Le but de la connectivité dans un RCFS peut être juste limité à assurer la disponibilité d'un chemin de chaque CH à la station de base ou bien être plus restrictive en imposant une limite à la longueur du chemin.

#### **3.2.4. Nombre minimal de cluster**

Cet objectif est particulièrement commun quand les CHs sont des nœuds spécialisés riches en ressources. Le concepteur de réseau préfère souvent employer un nombre minimum de ces nœuds. La limitation peut également être due à la complexité de déploiement de ce type particulier de nœuds.

#### **3.2.5. Longévité maximal du réseau**

Puisque les nœuds sont à énergie contraignante, la durée de vie de réseaux en dépend essentiellement. Quand les CHs sont plus riches en ressources que les nœuds capteurs, il est impératif de réduire au minimum l'énergie induite par la communication intra-cluster. Si possible, les CHs devraient être placés près de la plupart des nœuds de leurs clusters. D'autre part quand les CHs sont élus parmi les nœuds réguliers du réseau, leur durée de vie peut être prolongée en limitant leur charge comme indiqué précédemment. Une gestion de clusters combinée avec une découverte de route a également été considérée pour maximiser la durée

de vie du réseau. Un clustering adaptatif constitue également un choix fiable pour prolonger la longévité du réseau. [39]

### **3.3. Taxonomie des attributs de clustering**

Dans cette section nous allons énumérer l'ensemble des attributs qui peuvent être utilisés pour catégoriser et différencier les algorithmes de clustering dans les WSNs. Basé sur la discussion ci-dessus, nous pouvons identifier les attributs suivants :

#### **3.3.1. Propriétés de cluster**

##### **✓ Nombre de cluster**

Dans quelques approches éditées l'ensemble de CHs sont prédéterminés et le nombre de clusters est ainsi déterminé à l'avance. La sélection aléatoire de CHs parmi les nœuds déployés produit habituellement un nombre variable de clusters.

##### **✓ La stabilité**

Quand le nombre de clusters varie et les nœuds membres d'un cluster évolue dans le temps, le schéma de clustering est dit adaptatif. Autrement, il est considéré fixe puisque les nœuds ne s'échangent pas parmi les clusters et le nombre de clusters reste invariant tout au long de la durée de vie du réseau.

##### **✓ Topologie intra-cluster**

Certaines approches de clustering sont basées sur les communications directes entre un nœud et son CH. Cependant, la connectivité multi-saut de nœud à CH est parfois exigés, particulièrement quand la portée de communication est limitée et/ou le nombre de CHs est borné.[23]

##### **✓ Connectivité inter-CH**

Quand un CH n'a pas de possibilités de communication à longue portée, sa connectivité à la station de base doit être assurée. Dans ce cas, l'approche de clustering doit assurer la praticabilité d'établir un itinéraire inter-CH et de chaque CH à la station de base. Certains travaux supposent que les CHs pourraient atteindre directement la station de base.

#### **3.3.2. Possibilités de cluster-head**

##### **✓ Mobilité**

Quand un CH est mobile, ses membres changent dynamiquement et les clusters devraient être continuellement maintenus. Autrement dit, des CHs stationnaires tendent à produire des clusters stables et facilitent la gestion intra et inter clusters. Parfois, des CHs peuvent se déplacer sur des distances limitées pour se repositionner à des fins de meilleure gestion des performances.

### ✓ **Types de nœuds**

Comme indiqué auparavant, dans certaines installations, un sous-ensemble de nœuds déployés est désigné comme CHs tandis que dans d'autres, les CHs sont équipés de ressources de calcul et de communication plus significatives.[66]

### ✓ **Rôle**

Un CH peut simplement agir en tant que relais pour le trafic produit par les nœuds capteurs sans exécuter l'agrégation des données rassemblées de nœuds. Parfois, un CH agit en tant que sink ou station de base exécutant des actions sur des phénomènes ou la détection des cibles.

### **3.3.3. Processus de clustering**

#### ✓ **Méthodologie**

Quand les CHs sont justes des nœuds réguliers, (Figure 3.2) le clustering doit être exécuté de façon distribuée sans coordination. Dans certaines approches, une autorité centralisée partitionne les nœuds en offline et contrôle l'adhésion à un cluster. Les approches hybrides peuvent également exister particulièrement quand les CHs sont riches en ressources. [60] Dans ce cas, la coordination inter-CHs est réalisée d'une façon distribuée, tandis que chaque CH se charge individuellement de former son propre cluster.

#### ✓ **Objectifs de regroupement des noeuds**

Plusieurs objectifs ont été poursuivis pour former des clusters, la tolérance aux fautes, l'équilibrage de charge, la connectivité du réseau, la sociabilité, le routage, etc.

#### ✓ **Choix d'un cluster- head**

Les CHs peuvent être pré-assignés ou sélectionnés aléatoirement parmi l'ensemble de nœuds déployés.

#### ✓ **Complexité d'algorithme**

Selon l'objectif et la méthodologie, de nombreux algorithmes de clustering ont été proposés. La complexité et le taux convergence de ces algorithmes peut être constantes ou dépendant du nombre de CHs et/ou de nœuds capteurs.

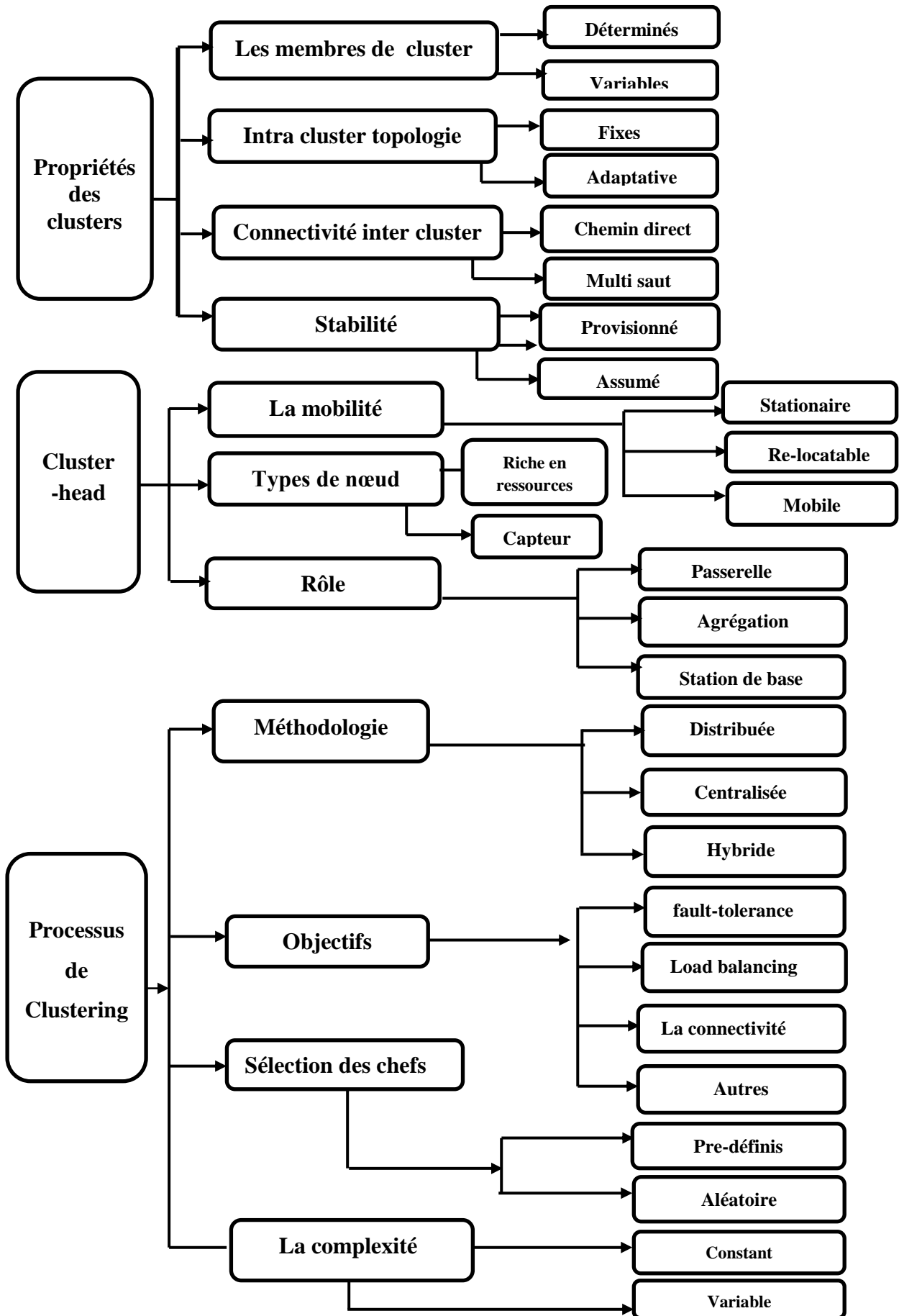


Figure 3.2 Taxonomie des attributs de clustering

## **4. Les protocoles de routage hiérarchiques proposés pour minimiser l'énergie dans les RCSFs « Travaux antérieurs »**

### **4.1. Energy-aware routing in cluster-based sensor networks**

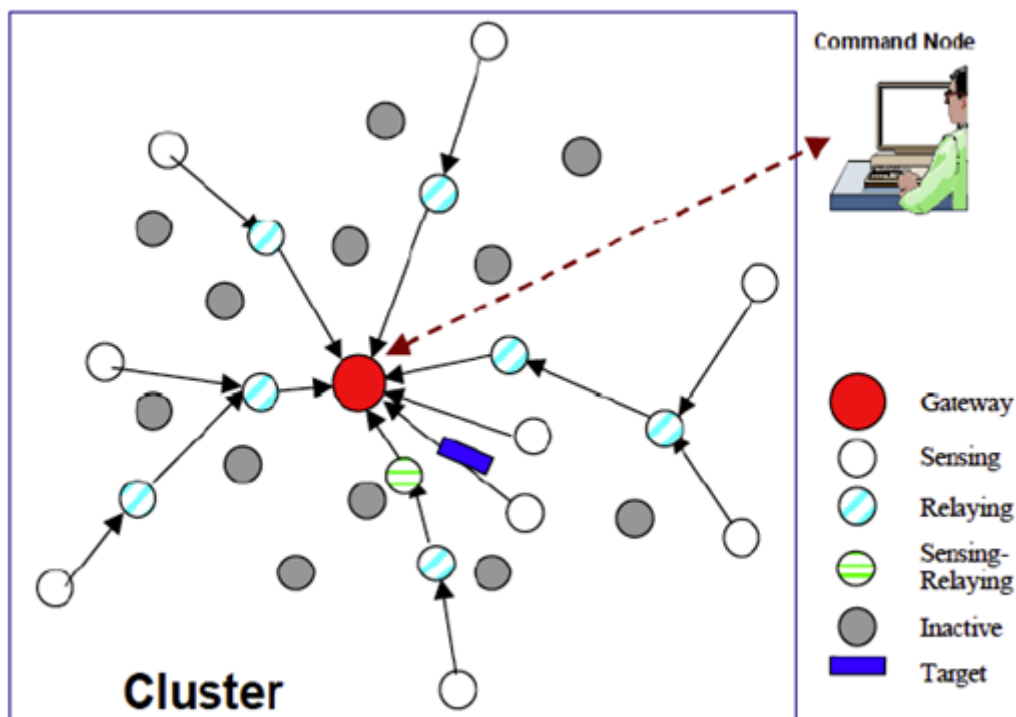
Younis et al. [62] ont proposé un algorithme de routage hiérarchique différent basé sur une architecture à trois niveaux. Les capteurs sont regroupés en clusters avant l'opération du routage.

L'algorithme utilise des CHs (passerelles), qui sont moins contraintes en énergie que les capteurs et supposées connaître l'emplacement des noeuds capteurs. Les passerelles maintiennent les états des capteurs et mettent en place des itinéraires multi-sauts pour la collecte des données. Un protocole MAC à base de TDMA est utilisé par les noeuds pour envoyer les données à la passerelle. Celle-ci informe chaque noeud sur les slots dans lesquels il devrait écouter la transmission aux autres noeuds et les slots que le noeud peut utiliser pour sa propre transmission. Le noeud de commande (sink) communique seulement avec les passerelles.

Le capteur est supposé être capable de fonctionner dans un mode actif ou dans un mode d'attente à faible puissance. Les circuits de détection et de traitement peuvent être mis sous tension et hors tension. En outre, l'émetteur et le récepteur radio peuvent être indépendamment activés et désactivés et la puissance d'émission peut être programmée en fonction de la gamme requise. Les noeuds capteurs dans un cluster peuvent être dans l'un des quatre états principaux : détecter uniquement, diffuser seulement, détecter et diffuser, et inactif. Dans l'état de détection, le noeud capte l'environnement et génère des données à un débit constant. Dans l'état de diffusion, le noeud ne détecte pas la cible mais son circuit de communication est présent pour transmettre les données provenant d'autres noeuds actifs. Quand un noeud est à la fois sensible et qui transmet les messages provenant d'autres noeuds, il est considéré dans l'état de détection-diffusion. Sinon, le noeud est inactif et peut désactiver son circuit de détection et de communication. La (figure 3.3) montre un exemple de l'état des capteurs et des liaisons au sein d'un cluster typique pour une application de suivi de cible. Une fonction de coût est définie entre deux noeuds quelconques en termes de consommation d'énergie, d'optimisation des retards et autres paramètres de performance. En utilisant cette fonction de coût comme coût d'un lien, un chemin de moindre coût est trouvé entre les noeuds capteur et la passerelle. La passerelle surveille en permanence le niveau d'énergie disponible à chaque capteur qui est actif dans le traitement des données, dans la détection, ou dans la

transmission de paquets de données. Le réacheminement est déclenché par un événement lié à une application nécessitant différents ensembles de capteurs pour sonder l'environnement ou par l'épuisement de la batterie d'un nœud actif.

Une variante de cette approche de routage a été proposée dans [62]. L'algorithme limite la portée de transmission minimum afin de limiter le retard. Les résultats des simulations ont démontré que cette approche fonctionne toujours bien au regard des deux métriques liées à l'énergie, par exemple, la durée de vie du réseau, ainsi que les métriques contemporaines, par exemple, le débit et le retard de bout en bout. Les résultats ont également indiqué que la combinaison de l'approche de routage avec l'arbitrage moyen basé sur le temps peut encore augmenter la durée de vie du réseau par un ordre de grandeur. Cependant, une telle approche suppose un simple modèle de propagation, ce qui pourrait nécessiter le déploiement de nombreuses passerelles pour assurer une couverture élevée du capteur.



**Figure 3.3** Cluster typique dans un réseau de capteurs [62]

#### **4.2. Algorithme FCM (Fuzzy C-Means)**

FCM (Fuzzy C-Means) est un algorithme de classification non supervisée floue. Il introduit la notion de sous-ensemble flou dans la définition des clusters : chaque nœud dans la zone de déploiement appartient à chaque groupe avec un certain degré, et tous les clusters sont caractérisés par leur centre de gravité.

### Chapitre 3: Les réseaux hiérarchiques et clustering « travaux antérieurs »

---

Comme d'autres algorithmes non supervisés de classification, FCM utilise un critère de minimisation des distances intra-clusters et maximise les distances inter-clusters, mais délivre un degré d'appartenance à chaque cluster pour chaque noeud. [19] Cet algorithme nécessite la connaissance préalable du nombre de clusters et génère les clusters selon un processus itératif en minimisant une fonction objective. Ainsi, il offre une partition floue de l'environnement en donnant à chaque noeud un degré compris entre 0 et 1 dans un cluster donné. Le cluster, qui est associé à un noeud, est celui dont le degré d'appartenance est le plus élevé.

FCM est basé sur la minimisation de la fonction objective suivante:

$$J_m = \sum_{i=1}^c \sum_{j=1}^N U_{ij}^m \cdot d_{ij}^2 \quad \text{Équation 3.1}$$

Où  $m$  est un nombre réel supérieur à 1,  $U_{ij}$  est le degré d'appartenance du noeud  $j$  au cluster  $i$ ,  $c$  est le nombre de clusters,  $N$  est le nombre de noeuds et  $d_{ij}$  est la distance euclidienne entre

le noeud  $j$  et le centre du cluster  $i$ .

L'algorithme se compose des actions suivantes :

1. Fixer une matrice d'appartenance arbitraire.
2. Calculer les centres des clusters en utilisant l'équation suivante :

$$z_j = \frac{\sum_{i=1}^N u_{ij}^m o_i}{\sum_{i=1}^N u_{ij}^m} \quad \text{Équation 3.2}$$

3. Réajuster la matrice d'appartenance selon la position des centres est effectué conformément à l'équation ci-dessous :

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{d_{ij}}{d_{kj}} \right)^{\frac{2}{m-1}}} \quad \text{Équation 3.3}$$

4. Calculer la minimisation et retourner à l'étape 2 s'il n'y a pas de critère de convergence.

➤ Dai et al [9] créent un arbre balancé des capteurs. Les capteurs diffusent, périodiquement, leurs existences et les informations de voisinage. Après la collection de ces informations, la station de base construit le graphe  $G(V, E)$ . Un algorithme est, par la suite, exécuté sur

G pour construire un arbre balancé de charge. Cet arbre prend en considération juste l'existence des capteurs et leurs informations de voisinage sans se soucier de la charge, en termes d'énergie des capteurs.

- Les auteurs dans [10] et [11] tentent d'équilibrer l'énergie des capteurs en considérant quelques informations dans le processus de routage. Alors que les auteurs dans [10] considèrent la localisation, l'énergie de la charge de travail des capteurs comme métriques pour la sélection du prochain capteur qui va router l'information à chaque communication inter capteurs jusqu'à la station de base, les auteurs dans [11] prennent en considération le reste de l'énergie des capteurs comme métrique, la capacité du buffer, le délai de transmission et la qualité de lieu. Ces deux travaux considèrent que ces informations doivent être échangées entre les capteurs afin de procéder aux routages. Par conséquent, cela prend plus de temps et rajoute des consommations inutiles d'énergie ce qui n'a pas été considéré au moment de l'amélioration du processus de routage et l'équilibrage des énergies des capteurs.
- les travaux de Heinzelman et al. [60] qui présentent aujourd'hui de bons résultats en termes d'économie d'énergie. Ces derniers ont proposé le protocole LEACH et une version centralisée de ce protocole, appelé LEACH-C. Tous deux s'appuient sur le clustering, qui consiste, à partitionner le réseau en groupes (clusters). Les nœuds transmettent leurs données vers des représentants de groupes dits clusterheads (CHs), qui à leur tour envoient ces données vers la destination désirée ou la station de base. Dans certaines applications les clusterheads font des traitements simples (agrégations par exemple) sur les données reçues avant de les retransmettre à la station de base. Cette approche permet la réutilisation de la bande passante. Elle offre aussi une meilleure allocation de ressources et aide à améliorer le contrôle de l'énergie dans le réseau. Dans LEACH, les nœuds s'auto-élisent pour être des CHs. Ils se basent sur le pourcentage désiré de CHs et le nombre d'itérations au cours duquel un nœud a pris le rôle de CH. Ainsi, un nœud  $n$  prend une valeur aléatoire entre 0 et 1. Si cette valeur est inférieure au seuil  $T(n)$ , le nœud se déclare CH. Les CHs informent leur voisinage de leur élection. Tous les autres nœuds décident de choisir le CH le plus proche pour joindre son groupe. Les CHs reçoivent les réponses des nœuds. Ils créent des tables TDMA en se basant sur le nombre de nœuds constituant le cluster. Chaque nœud transmet ses données vers le CH de son cluster en fonction des intervalles de temps spécifiés dans les tables TDMA. Les nœuds éteignent leurs antennes en attendant leurs slots. Cette technique permet ainsi de minimiser la dissipation de l'énergie. Par contre, les CHs laissent leurs récepteurs en état de marche pour recevoir toutes les données des nœuds

appartenant à leur cluster. Une fois que toutes les données sont reçues, elles sont compressées par les CHs et transmises à la station de base. Chaque CH choisit aléatoirement un code dans une liste de codes de propagation CDMA, il le transmet aux nœuds appartenant à son cluster afin de l'utiliser pour leurs transmissions, ceci afin de minimiser les interférences entre les messages des CHs les plus proches.

#### **4.3. HEER: A Delay-aware and Energy Efficient Routing Protocol for Wireless Sensor Networks**

Les auteurs (Yi et al., 2016) [1] proposent un protocole de clustering amélioré, prenant en compte la durée de vie et la consommation d'énergie, nommé protocole d'acheminement à efficacité énergétique selon la méthode de Hamilton (HEER). HEER forme des clusters dans la phase d'initialisation du réseau et il relie les membres de chaque cluster avec un chemin Hamilton, construit à l'aide d'un algorithme gourmand, pour la transmission de données. Aucune réformation en cluster n'est nécessaire et les membres sur le chemin tourneront pour devenir ClusterHead.

➤ La contribution de cet protocole est triple: premièrement, ils adoptent le concept de Hamilton Path pour lier des membres dans chaque cluster formée sans avoir besoin de l'information de position globale du nœud individuel, ce qui réduit la distance de transmission pour chaque CM ainsi que la puissance / CHs. Deuxièmement, ils conçoivent HEER d'une manière que les clusters ne sont formés qu'une seule fois de sorte que les frais généraux pour l'administration du réseau est considérablement réduit. Enfin, ils contrôlent la taille du cluster et la longueur de trame maximale dans 6LowPan et la charge utile de données de chaque CM de sorte que l'utilisation de la capacité de transmission de réseau est maximisée. HEER parcourt tous les nœuds une fois dans le cluster pour chaque tour de sorte que les nœuds ne sont pas accessibles à plusieurs reprises. Cette fonction réduit l'accès fréquent au CH. En outre, HEER ne forme des clusters qu'au premier tour, l'énergie pour la reformulation des clusters peut être sauvegardée.(Figure 3.4) [1] La durée de vie de RCSF peut donc être prolongée. Les résultats indiquent que le protocole HEER a des performances optimales dans tous les scénarios, en termes de durée de vie du réseau.

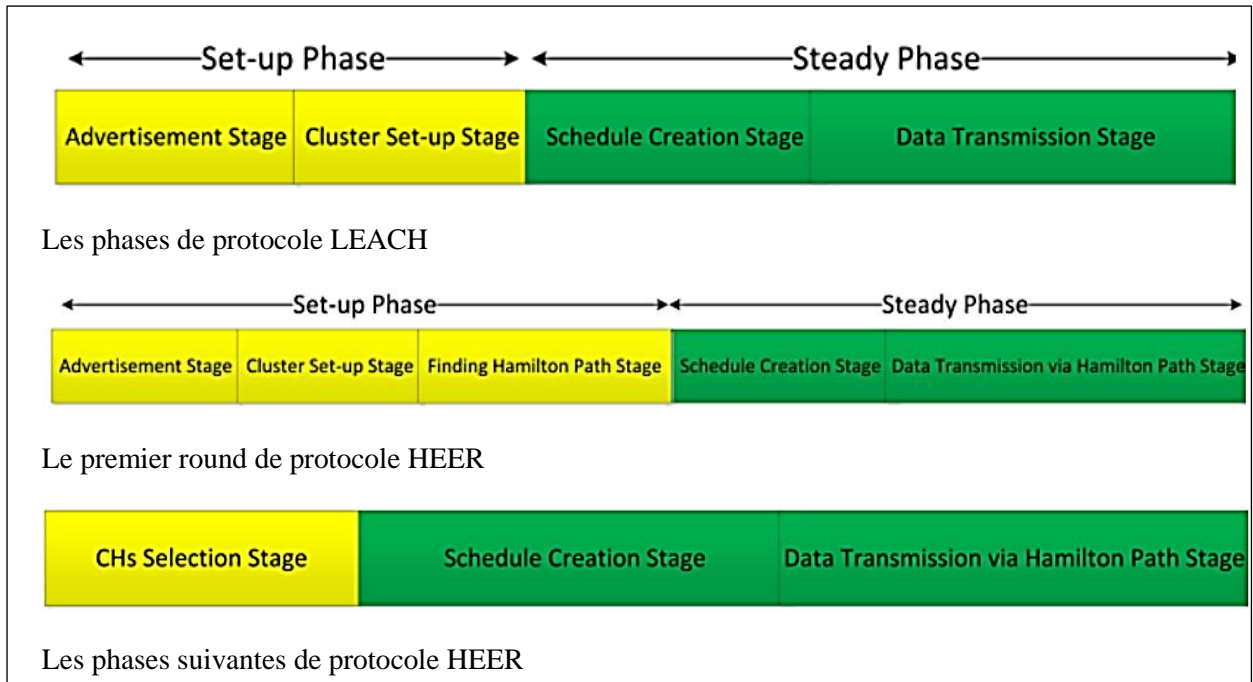


Figure 3.4 Les phases de protocole HEER [1]

#### 4.4. ICP: Instantaneous clustering protocol for wireless sensor networks

Les auteurs (*linghe et al., 2016*) [2] étudient le problème du regroupement des clusters en considérant à la fois le temps et l'efficacité énergétique. Ils proposent un nouveau protocole de clustering instantané (ICP) qui regroupe les nœuds de capteurs en cluster à un seul saut de manière parallèle. ICP peut instantanément terminer le regroupement en raison de deux conceptions clés. Tout d'abord, pour déterminer les têtes de cluster localement. Les méthodes existantes exigent une longue durée pour élire le CH. Pour renoncer à la consommation de vote, une tête de cluster dans le PIC est localement déterminée par la probabilité pré-assignée et son état actuel.[6] Deuxièmement, minimiser la quantité de transmissions. Les transmissions en parallèle à partir de différentes puissances et l'acquisition (ACK) de plusieurs membres du cluster entraînent une consommation de temps et d'énergie sévère. Au contraire, l'ICP se débarrasse le mécanisme ACK, au lieu de cela, seules les chefs de cluster se disputent la diffusion pendant une période donnée. Cette période est élaborée pour garantir la connectivité. Des expériences sur un nœud testé et des simulations sur des WSN à grande échelle sont largement conduites pour évaluer l'ICP. [2] Les résultats de performance démontrent que l'ICP surpasse de manière significative les méthodes de clustering existantes en réduisant jusqu'à 55% de consommation de temps et 89% de transmissions pour économiser l'énergie. (figure 3.5)

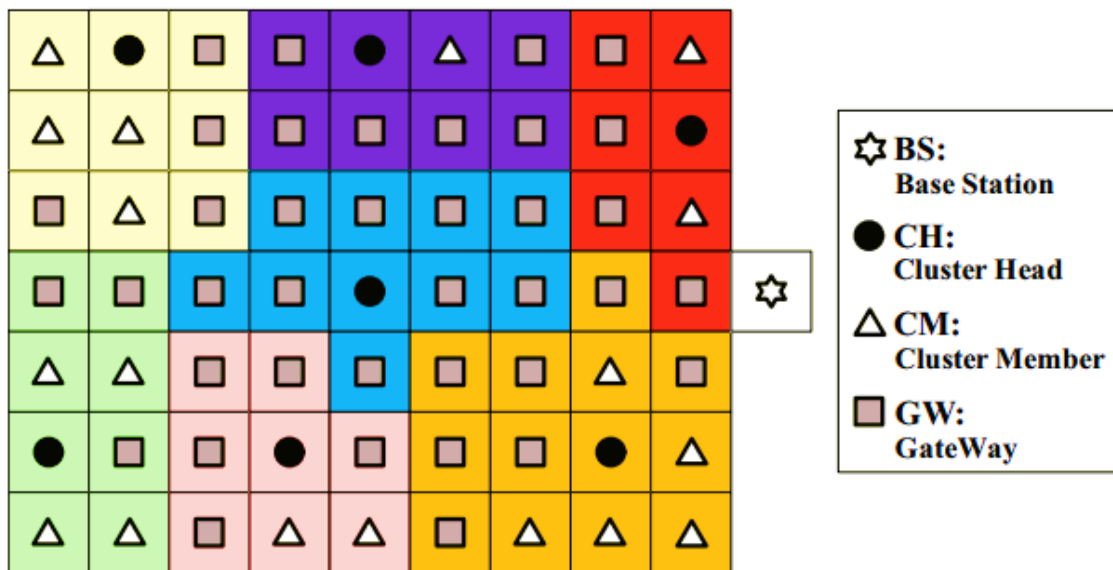


Figure 3.5 : un exemple de résultat d' ICP [2]

#### 4.5. Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors

Les auteurs (Pengfei et al., 2013) [3], ils présentent plusieurs algorithmes de clustering efficaces pour maximiser la durée de vie des WSN, c'est-à-dire la durée jusqu'à ce qu'un certain pourcentage des nœuds meurent. Plus précisément, un algorithme d'optimisation est proposé pour maximiser la durée de vie d'un réseau à cluster unique, suivi d'une extension pour gérer des réseaux multi-clusters. Ensuite, l'auteur étudie le problème commun de prolonger la durée de vie du réseau en introduisant des nœuds de collecte d'énergie (EH). (figure 3.6)

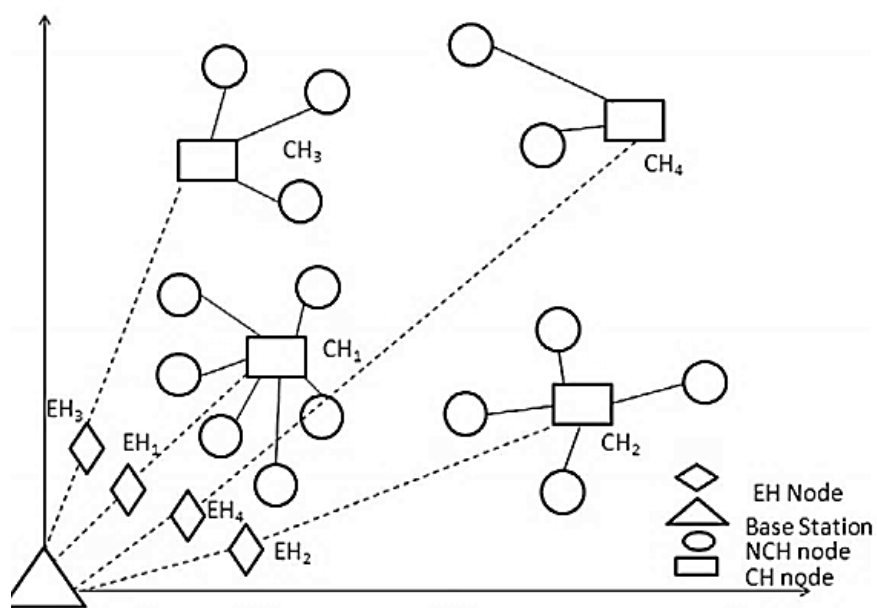
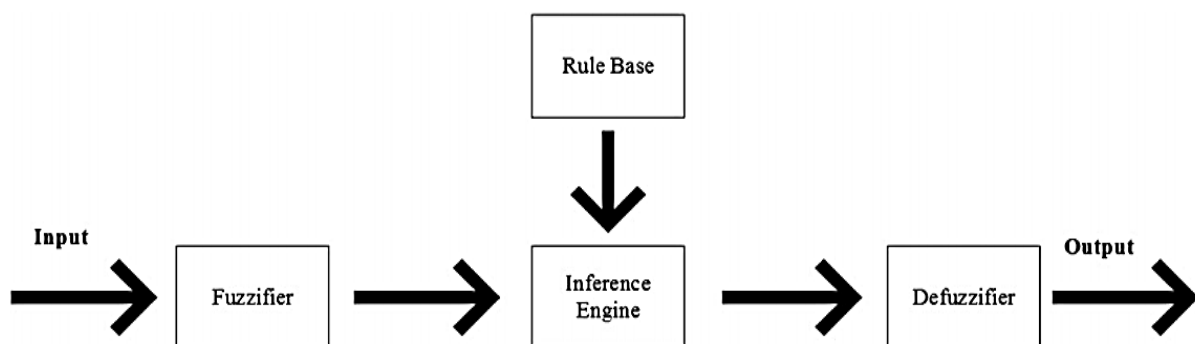


Figure 3.6 Le model de réseau d'algorithme de clustering [3]

Un algorithme est proposé pour maximiser la durée de vie du réseau où les noeuds EH servent de noeuds de relais dédiés pour les têtes de cluster (CHs). [3] L'analyse théorique et les résultats de simulation étendus montrent que les algorithmes proposés peuvent atteindre efficacement des solutions optimales ou sous-optimales, ce qui contribue à fournir des repères utiles pour diverses conceptions de schémas de clusters centralisés et distribués.

#### **4.6. An Energy Aware Distributed Clustering Algorithm Using Fuzzy Logic for Wireless Sensor Networks with Non-uniform Node Distribution**

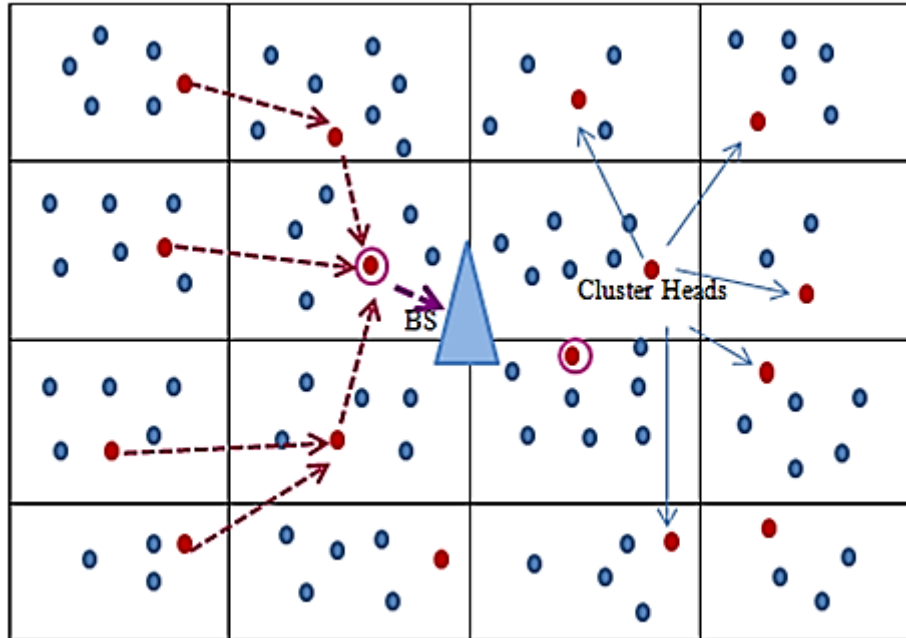
Dans le travail de (Nokhanji *et al.*, 2015) [4] un algorithme de routage EADC-FL (Energy Aware Distributed Clustering) utilisant la logique floue (EADC-FL) est proposé pour le réseau de capteurs sans fils avec une distribution de noeud non uniforme. EADC-FL sélectionne les noeuds les plus appropriés en tant que tête de cluster pour économiser plus d'énergie en phase de transmission de données. Il considère l'énergie des noeuds capteurs comme le paramètre principal pour élire les cluster-head candidats. Ensuite, il prend en considération le degré de noeud et la centralité des noeuds de capteurs en utilisant la logique floue pour identifier les cluster-heads finales.[4] EADC-FL construit les clusters de tailles égales pour équilibrer la consommation d'énergie des membres du cluster et utilise un algorithme de routage basé sur le cluster pour équilibrer la consommation d'énergie entre les chefs de cluster. En outre, l'algorithme proposé effectue le regroupement à la demande au lieu de chaque cycle afin de réduire la dissipation d'énergie du réseau en raison de la surcharge de phases de regroupement consécutives. (figure 3.7) Les résultats démontrent que EADC-FL diminue de façon significative la consommation d'énergie et prolonge la durée de vie du réseau du réseau avec une distribution de noeud non uniforme.



**Figure 3.7** La structure de Fuzzy System [4]

#### 4.7. FEMCHRP: Fuzzy Based Energy Efficient Multiple Cluster Head Selection Routing Protocol for Wireless Sensor Networks

Les auteurs (*Sohail et al., 2015*) [5] proposent un nouveau protocole de routage appelé Fuzzy Based Energy Efficient Multiple Cluster Head Selection Routing Protocol (FEMCHRP) pour les réseaux de capteurs sans fils. Le processus de routage implique le regroupement des noeuds et la sélection des noeuds Cluster Head (CH) de ces clusters qui envoient toutes les informations au chef de grappe (CHL) (figure 3,8). Ensuite, les chefs de tête de cluster envoient des données agrégées à la station de base (BS). La sélection des cluster-heads des clusters est réalisée en utilisant la logique floue et le processus de transmission de données est exécuté par le chemin d'énergie le plus court qui est sélectionné en appliquant l'algorithme de Dijkstra. Le nouveau protocole de routage organise des clusters afin que tous les nœuds puissent être inclus dans ces clusters. Il choisit CHs pour chaque cluster en utilisant la logique Fuzzy, qui est basée sur la plus grande reprise d'énergie et la distance minimale de la BS. [5] Dans ce protocole, l'algorithme de Dijkstra est appliqué pour trouver un chemin d'énergie plus court de chaque noeud. Ensuite, il regroupe les membres du cluster et les CH selon les chemins d'énergie les plus courts et enfin, les membres du cluster envoient des paquets de données aux CH.



**Figure 3.8** Le modèle de réseau du protocole FEMCHRP [5]

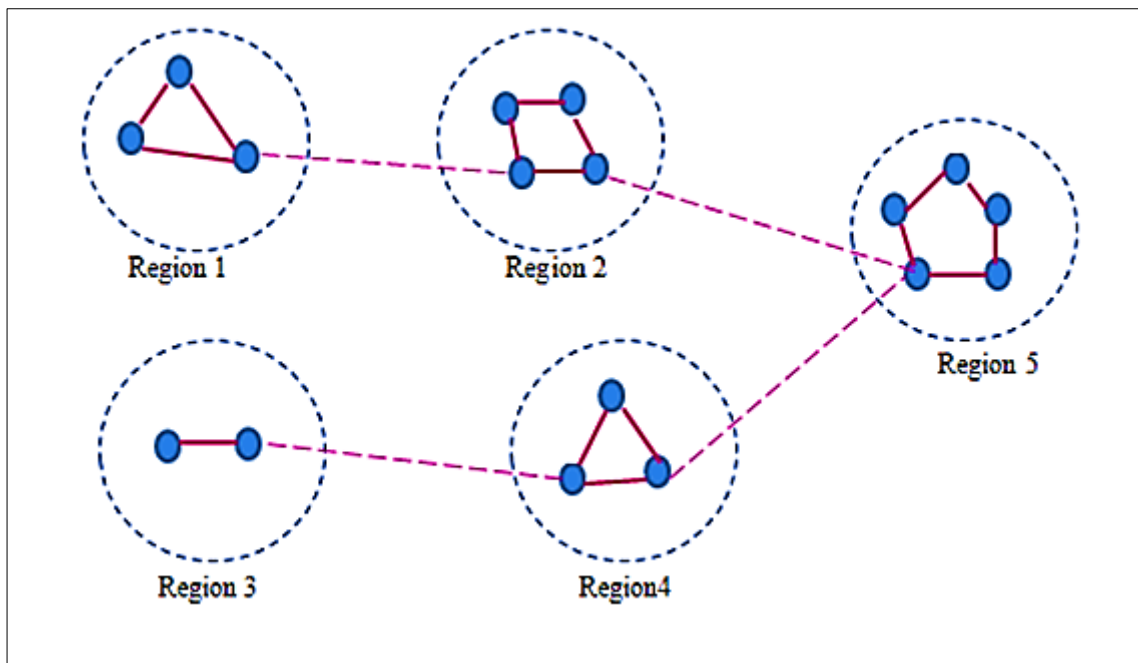
Ce protocole, multiples CHL sont choisis par la BS en utilisant la logique floue basée sur la reprise de la plus haute énergie et la distance minimale de BS de chaque CH. Chaque CHL peut transmettre des données directement ou par d'autres CHL à la BS, selon le chemin d'énergie le

plus court. Ils simulent ce réseau pour analyser la durée de vie du réseau, l'énergie résiduelle moyenne et la dissipation d'énergie de cette étude. Les résultats de la simulation montrent que ce protocole est préférable pour prolonger la durée de vie du réseau et équilibrer la consommation d'énergie par rapport au BCDCP, CELRP et ECHERP.

#### **4.8. Adaptive energy aware cluster-based routing protocol for wireless sensor networks**

##### **DEEAC**

Les auteurs [65] présentent un protocole de routage adaptatif basé sur le cluster adapté à l'énergie pour les WSN, vise à améliorer la durée de vie du réseau et les performances de livraison de données avec un minimum de frais généraux. Contrairement à la plupart des travaux existants, le protocole proposé améliore la conservation de l'énergie et l'information de routage dans certaines facettes. Dans la première phase, en fonction de la taille du réseau, le champ du capteur entier est divisé en grappes de taille équilibrée. Ensuite, sur la base d'un score pondéré, les noeuds optimisés sont sélectionnés comme un ensemble de CH dans la région délimitée, ce qui diminue les frais généraux du réseau et la consommation d'énergie. [65] Dans la deuxième phase, des chemins de routage multi-sauts sont découverts aux niveaux intra-cluster et inter-clusters qui raccourcissent la distance de transmission, améliorent la conversation énergétique avec la transmission de données fiables dans le champ du réseau entier. Enfin, à la fin, pour améliorer la connectivité du réseau, le rôle des CHs est tourné entre les nœuds concurrents sur demande. (figure3.9)



**Figure 3.9** Organisation en cluster avec chaines [65]

#### 4.9. SHaRP: A Hierarchical and Role Based Secure Routing Protocol for Mobile Wireless Sensor Networks

Les auteurs (*Hiren et al., 2016*) [7] proposent un protocole de routage hiérarchique sécurisé et basé sur des rôles pour le réseau de capteurs mobile sans fil. (SHaRP) est une combinaison d'approche cryptographique symétrique et asymétrique. Dans le cadre SHaRP, le champ du capteur est divisé en quelques clusters logiques et chaque cluster contient des noeuds avec des rôles différents tels que le noeud de capteur ordinaire (OSN), la tête de cluster (CH) et le noeud de passerelle (GN) utilisé par les noeuds avec des rôles distincts pour le chiffrement des messages et le décryptage des messages. [7] La sécurité globale a été confiée à différents rôles d'une manière répartie au lieu de s'appuyer sur une autorité et c'est pourquoi cette approche est à l'abri d'un seul point de problème d'échec. (figure 3.10)

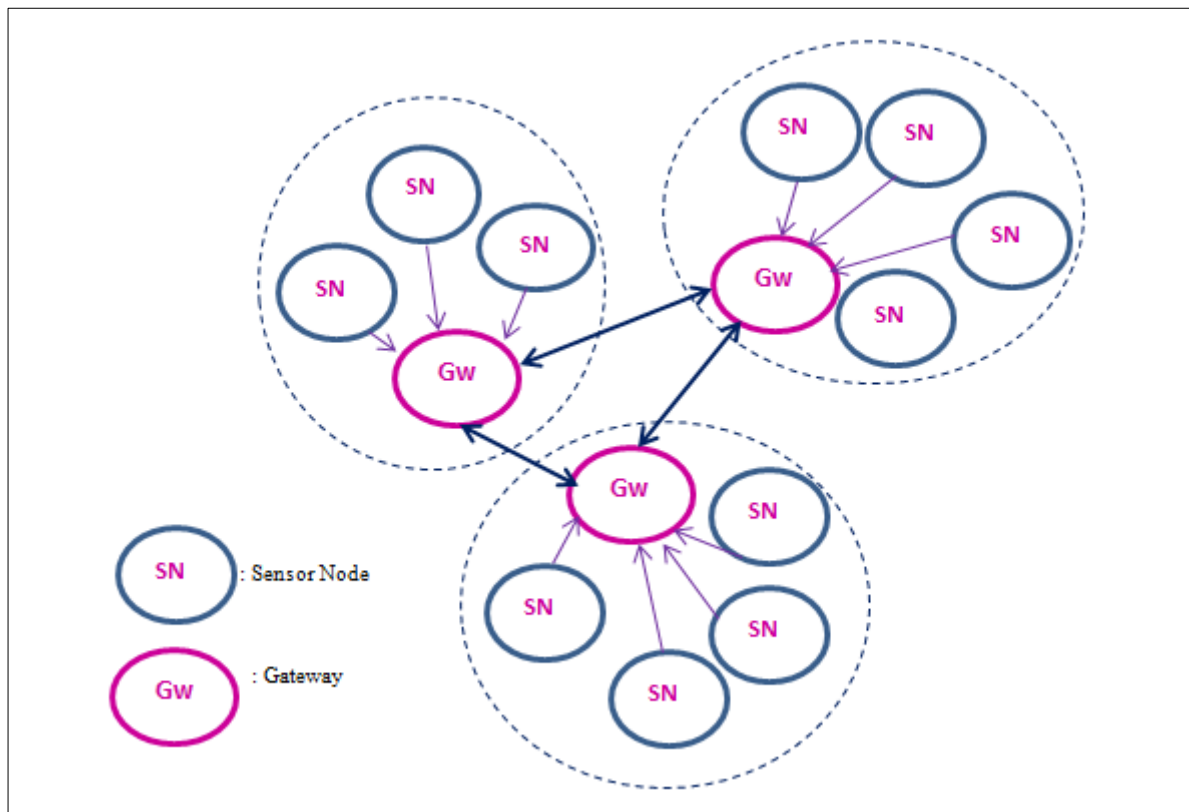


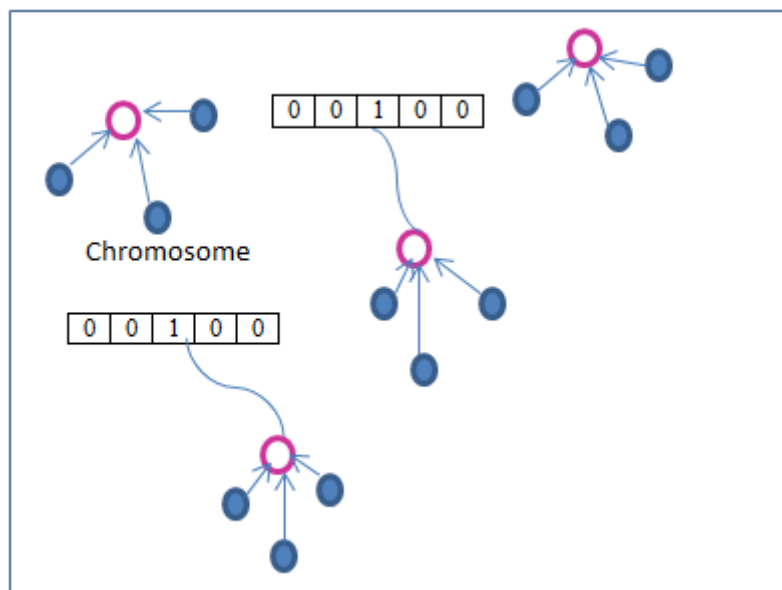
Figure 3.10 : la méthode de clustering dans SHaRP [7]

Cela améliore également la sécurité dans le système réseau. L'auteur suppose que la station de base utilise un algorithme de clustering efficace. Dans ce travail, ils se concentrent sur l'acheminement d'une partie du système dans lequel la décision concernant quel nœud à qui et quand les données doivent être transmises est la principale préoccupation. Chaque cluster dispose d'un noeud de passerelle (GN), de deux noeuds de tête de cluster (CH) et d'un noeud de capteur ordinaire (OSN). Ces différents rôles tels que GN, CH ou OSN pour différents noeuds

de capteurs sont attribués par la station de base. Un champ de capteurs groupé typique est représenté dans la figure 3.10, la plupart des processus de cryptage-déchiffrement de message sont symétriques. Les mêmes clés sont utilisées pour le cryptage ainsi que le processus de déchiffrement. La génération de clé et le processus de cryptage-déchiffrement pour la paire CH et GN sont basés sur la cryptographie à courbe elliptique (ECC) et il s'agit d'une instance de cryptographie à clé asymétrique. Ils utilisent une combinaison de différents schémas de gestion de clés tels qu'une clé maître simple, une clé à paires et une clé basée sur un groupe.

#### **4.10. GASONeC: A Genetic Algorithm-Based, Dynamic Clustering Method Towards Improved WSN Longevity**

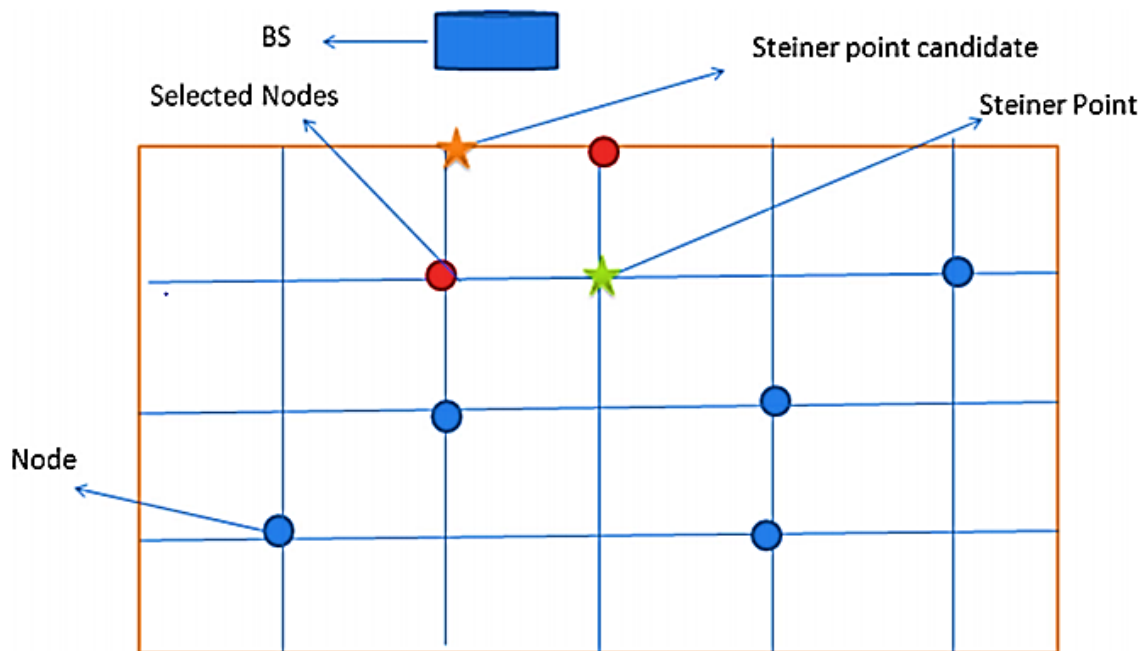
Les auteurs (*Xiaohui et al., 2016*) [8] proposent une méthode basée sur l'algorithme génétique, auto-organisée de regroupement de réseau (GASONeC) qui fournit un cadre pour optimiser dynamiquement les clusters de nœuds de capteurs sans fil. Dans GASONeC, l'énergie résiduelle, la dépense d'énergie attendue, la distance à la station de base et le nombre de nœuds dans le voisinage sont employées à la recherche d'une structure de réseau dynamique optimale. L'équilibrage de ces facteurs est la clé de l'organisation des nœuds en clusters appropriées et la désignation d'un nœud de substitution comme cluster-head. Dans GASONeC, un chromosome binaire est utilisé pour spécifier les têtes de cluster dans le réseau, dans lequel a représenté une tête de cluster et un zéro représente un noeud de membre à un cluster. Un exemple est montré dans (Figure 3.11) et les têtes de cluster sont mises en surbrillance avec des cercles remplis.



**Figure 3.11 :** Le modèle de protocole GASONeC [8]

#### 4.11. MRRCE: Multi-hop Routing Algorithm Using Steiner Points for Reducing Energy Consumption in Wireless Sensor Networks

Les auteurs (*Elham et al., 2015*) [9] proposent un schéma de routage d'efficacité énergétique multi-sauts qui considère les facteurs d'énergie résiduelle et de distance pour développer des clusters et CH optimales pour atteindre l'efficacité et prolonger la durée de vie du réseau. Par MRRCE, les rounds dans lesquels le premier noeud meurt et le dernier noeud meurt sont fortement augmentés. (Figure 3.12) La transmission multi-sauts améliore l'efficacité énergétique des RCSFs en organisant les CH dans un routage hiérarchique multi-sauts, chaque CH peut avoir plusieurs chemins vers la BS, en utilisant l'algorithme de routage MRRCE en considérant la distance et l'énergie résiduelle de CH, ils trouvent toujours le meilleur chemin entre chaque CH à son niveau supérieur et BS, de sorte que cela empêche la perte d'énergie supplémentaire dans la communication réseau. [9] L'auteur a étudié certains algorithmes de routage tels que LEACH, DIRECT, HEED, EHEED et TDTCGE. DIRECT utilise un routage à un seul saut pour envoyer les données à la BS afin que les noeuds qui sont plus près de la BS brûlent leurs batteries rapidement et par conséquent, la durée de vie du réseau sans fil va diminuer. Les algorithmes LEACH, HEED, EHEED utilisent le processus probabiliste pour choisir CH de sorte qu'il ne va pas à la sélection CH absolument optimale.



**Figure 3.12 :** Processus de sélection des points Steiner [9]

#### **4.12. Effect of multi-path fading model on T-ANT clustering protocol for WSN**

Ce protocole (Kumar et al., 2016) utilise à la fois la structure hiérarchique et l'inspiration biologique. Cet article étudie l'effet du modèle de décoloration multi-chemin sur le protocole T-ANT et l'étude comparative des résultats avec le protocole T-ANT dans un modèle isotrope pour les WSN dans un environnement simulé sur la plate-forme MATLAB. En particulier, ils s'intéressent à l'étude de l'effet du modèle de décoloration multi-chemin sur la condition physique en grappes, l'aptitude à l'élection des chefs de grappes et la répartition de la charge de travail parmi les noeuds de capteurs. Les résultats montrent que le protocole T-ANT dans le modèle d'évanouissement multi-chemin effectue peu plus bas que le protocole T-ANT dans le modèle isotrope sans affecter les propriétés de fitness de clustering. T-ANT clustering protocole fonctionne en rondes. Un cycle unique comprend une phase de configuration de groupe et une phase d'état stationnaire. La phase de configuration du cluster se compose d'un composant de libération de fourmi et d'un composant de cluster de fourmi. La composante de formage des fourmis se compose de deux composantes, à savoir l'élection CH et la formation de grappes. Ce protocole utilise des agents sociaux nommés fourmis pour contrôler le nombre d'élection de CH dans le WSN. Après la libération des fourmis pendant la phase de libération des fourmis, le noeud du capteur vérifie s'il possède une fourmi. Le noeud qui possède la fourmi devient CH et les noeuds restants se regroupent avec le meilleur CH approprié. Le nombre fixe de fourmis est inondé dans le réseau, donc, un nombre fixe de CHs sont formés. En phase stationnaire, les données sont collectées à partir des noeuds non CH, agrégées et transférées à la station de base via CH à des intervalles de temps fixes.

#### **4.13. MECP: Mobility- and Energy-Conscious Clustering Protocol for Wireless Networks**

Les auteurs (Singh et al., 2015) [10] présentent un protocole de regroupement distribué pour les réseaux de capteurs mobiles sans fil. Une grande majorité de la recherche dans les algorithmes de clustering et de routage pour RCSFs supposent un réseau statique et donc ils sont rendus inefficaces dans les cas de réseaux de capteurs très mobiles, qui sont un aspect abordé ici. MECP est un protocole économe en énergie et adapté à la mobilité et utilise des informations sur le mouvement des noeuds capteurs et de l'énergie résiduelle comme attributs dans la formation du réseau. Il fournit également un mécanisme de tolérance aux pannes pour diminuer la perte de données par paquets en cas de défaillance de la tête de cluster. Chaque noeud prend des décisions sur la base de deux paramètres, à savoir l'énergie résiduelle et la vitesse relative par rapport à ses voisins. [10] Les coûts de communication intra-cluster sont également pris en compte pour le processus de clustering afin d'augmenter l'efficacité de la consommation

d'énergie. Par exemple, le coût peut être une fonction de la distance de CH ou densité de nœud du groupe. Ils réservent les niveaux de puissance inférieurs pour la communication intra-cluster entre un nœud normal et un CH pour réduire les coûts de communication. En outre, ils réservent les niveaux de puissance d'émission plus élevés pour la communication inter-cluster. Si le niveau de puissance utilisé pour la communication intra-cluster est fixe, le coût de communication pour un nœud peut être déterminé par le degré de nœud ( $D$ ). [10] Par exemple, pour créer des clusters denses, le coût de communication avec un CH est fixé proportionnellement à  $1 / D$  de sorte que les clusters denses ont un coût de communication plus faible. Alternativement, pour le cluster à équilibrage de charge, le coût de communication est établi proportionnellement à  $D$  de sorte que les clusters denses ont un coût de communication plus élevé. [10] Par conséquent, un nouveau nœud préférerait rejoindre un cluster où le coût de la communication est plus faible.

#### **4.14. GFTCRA :A grid based clustering and routing algorithm for solving hot spot problem in wireless sensor networks**

Les auteurs (*Srikanth et al., 2016*) [11] abordent le problème des points chauds et ils proposent des algorithmes de regroupement et de routage basé sur une grille, de manière combinée appelée GFTCRA (de cluster à tolérance de panne sur la base de la grille et des algorithmes de routage) , qui prend soin de l'échec de la SHC. (Figure 3.13) Les algorithmes suivent une approche distribuée. Ils présentent également une gestion de temps d'exécution distribuée pour tous les nœuds de capteurs de n'importe quel cluster en cas d'échec de leurs CHs. L'algorithme de routage est également montré pour tolérer l'échec soudain des CH. [11] Les algorithmes sont testés par simulation avec divers scénarios de WSN et les résultats de simulation montrent que la méthode proposée fonctionne mieux que deux autres algorithmes basés sur la grille en termes de durée de vie du réseau, de consommation d'énergie et de nombre de nœuds capteurs morts.

Les principaux avantages de l'algorithme proposé par rapport aux algorithmes existants sont les suivants: L'algorithme est distribué dans la nature qui ne nécessite que l'information locale et évite les informations globales du réseau. Il aborde le problème des points chauds en considérant tous les problèmes, à savoir, l'efficacité énergétique, l'équilibrage de charge et la tolérance aux pannes ensemble. Il évite les frais généraux de regroupement mais peut toujours gérer le routage des données en cas de défaillance des nœuds de capteurs et des CH.

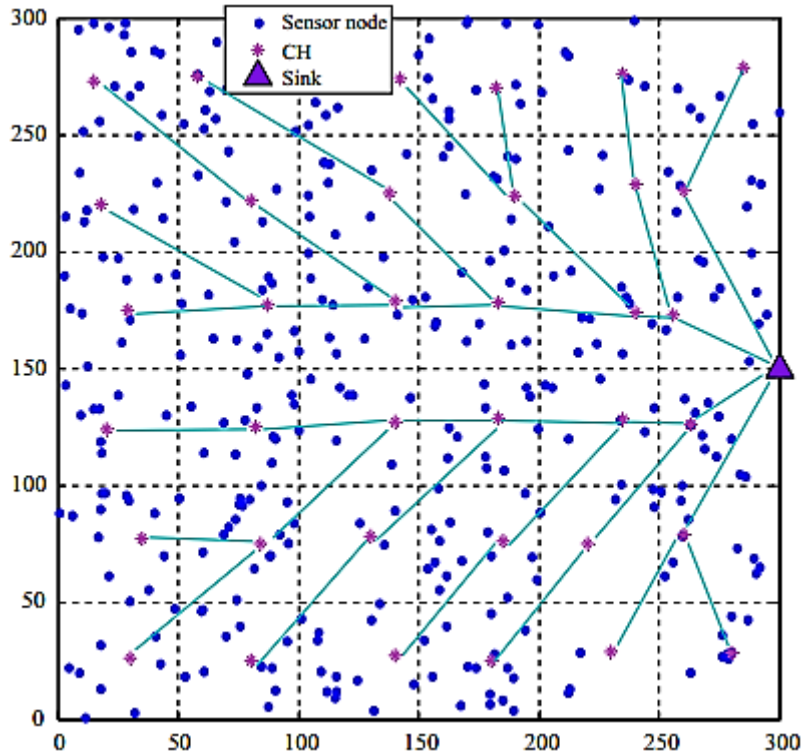


Figure 3.13 : Construction des grids GFTCRA [11]

#### 4.15. An Energy Efficient Cluster Head Selection Technique Using Network Trust and Swarm Intelligence

Les auteurs (*Juliana et al., 2016*) [12] proposent un mécanisme d'affectation spéciale pour la sélection de cluster-head dans un RCSF en utilisant l'algorithme 'Artificial Bee Colony'. Les résultats montrent que la méthode proposée évite les nœuds avec un comportement égoïste et une nature malveillante qui permettent le trou noir, le déni de service ou la perte de paquets.

Dans ce travail, ils proposent une approche de clustering utilisant l'optimisation des colonies d'abeilles artificielles avec les hypothèses suivantes: Les nœuds sont distribués de façon aléatoire sur deux dimensions.

Tous les nœuds sont similaires avec une plage de transmission uniforme  $r$ , les canaux sont symétriques. [12] L'algorithme de regroupement se compose de deux étapes, à savoir l'élection CH et la formation de cluster, où les deux sont exécutés pendant la phase de configuration de chaque round. Le succès de la livraison et la latence sont directement proportionnels au nombre de sauts (*Nhops*). Le nombre de trafic acheminé par agrégation des informations provenant d'autres CH est réduit. Cependant, la surcharge du CH près de SB est plus élevée en raison de la transmission supplémentaire de données provenant de nœuds plus éloignés (Figure 3.14).

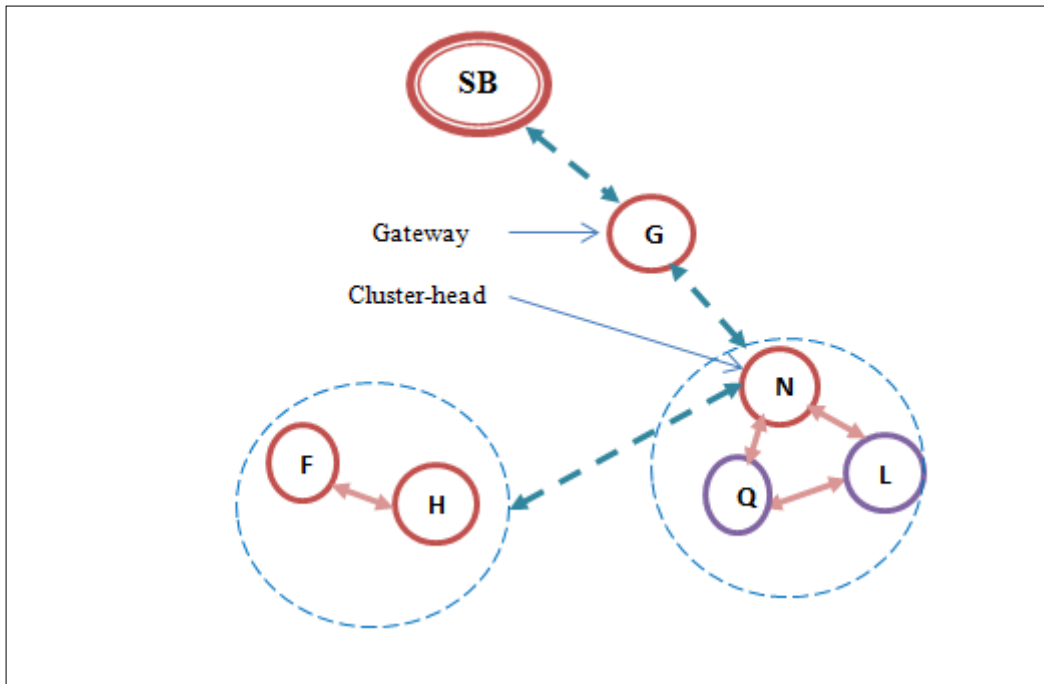


Figure 3.14 : Structure de Clustering avec gateway [12]

#### 4.16. H-CERP: Energy Efficient Scheme for Industrial WSN Applications

Les auteurs (Subha et al., 2016) [13] proposent un hybride énergétique clustering protocole de routage (H-CERP) pour RCSFs hétérogènes. H-CERP est conçu pour former des clusters efficaces avec moins de comptage de CHs que l'estimation optimale et utilise la communication multi-sauts avec des nœuds passerelles pour communiquer avec la station de base (Figure3.15). Cette nouvelle approche rend le système plus avantageux là où la durée de vie du réseau et la couverture des capteurs sont très vitales sans coût supplémentaire. [13] En déployant H-CERP dans un environnement conçu, les résultats extraits montrent des résultats prometteurs en termes de consommation d'énergie, d'énergie résiduelle et de durée de vie des nœuds par rapport aux méthodes génériques tels que LEACH, PEGASIS et d'autres techniques récents. L'algorithme proposé fonctionne en rounds. Chaque round effectue les étapes suivantes: Périodiquement, la station de base lance un nouveau cycle en incrémentant le nombre de tours. Le nombre optimal de groupes et le nombre de CH peuvent être estimés à partir du protocole conventionnel de LEACH. Pour améliorer la durée de vie du réseau, l'H-CERP maintient le nombre de groupes et le nombre de CH inférieur à la valeur optimale. Dès qu'un CH est formé, il sélectionne un nœud de passerelle le plus proche. Les nœuds du capteur se réveillent, détectent les données et transmettent les données détectées aux CH respectifs. Le CH regroupe les données reçues de tous les membres du cluster et envoie ensuite les données aux nœuds de passerelle sur la base d'une communication univoque. [13] Les nœuds de passerelle envoient ensuite les données à la BS.

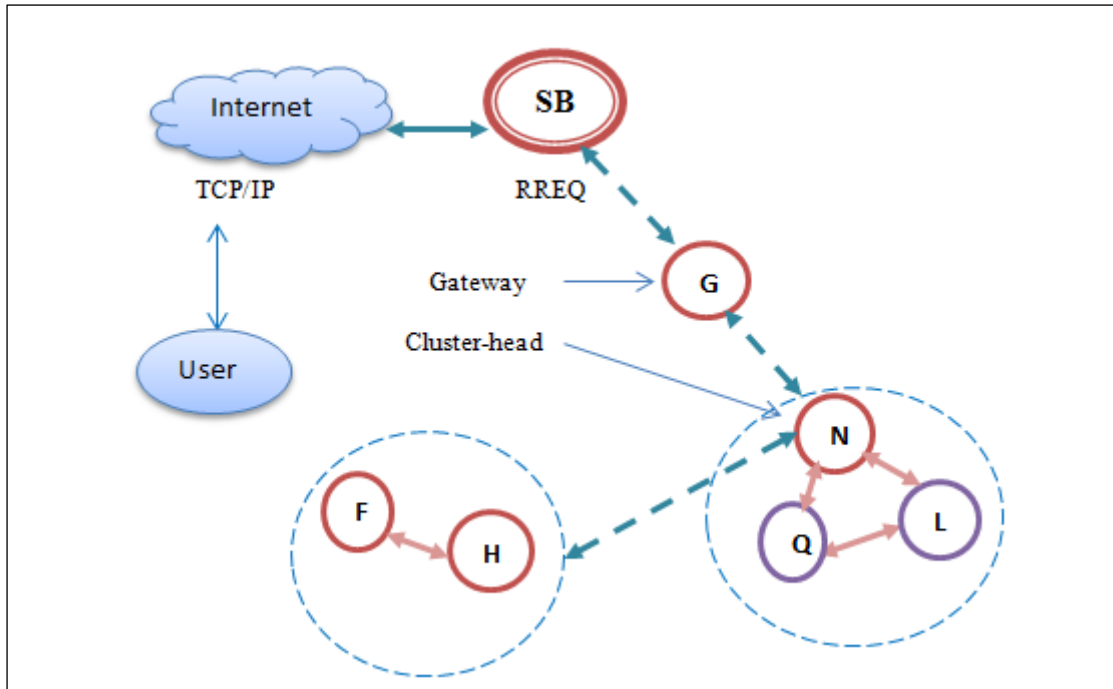


Figure 3.15 : Le modèle de réseau de protocole H-CERP [13]

#### 4.17. EDDUCA: An Energy-Efficient Unequal Clustering Algorithm Using ‘Sierpinski Triangle’ for WSNs

Les auteurs (*Guiloufi et al., 2016*) [14] proposent un nouvel algorithme (EDDUCA) visant à équilibrer la consommation d'énergie et de maximiser la durée de vie du réseau. EDDUCA utilise la méthode du «triangle de Sierpinski» pour diviser le réseau en grappes inégales. Les résultats obtenus indiquent que EDDUCA peut efficacement équilibrer la consommation d'énergie et donc prolonger la durée de vie du réseau. Minimiser la consommation d'énergie: habituellement, [14] les algorithmes de clustering utilisent des grappes dynamiques. Dans ce cas, le nombre de clusters et de membres du cluster change dynamiquement.

Équilibrer la consommation d'énergie entre clusters: ils ont supposé que les nœuds sont uniformément déployés dans la zone carrée. Dans la phase de formation des clusters, ils définiront la méthode de regroupement inégal comme montré dans (Figure 3.16).

Les grappes les plus proches de la BS ont une taille inférieure à celle des autres grappes. Dans ce cas, ils auront plus d'énergie pour réaliser la communication inter-cluster. De plus, étant donné que les noeuds sont déployés uniformément, lorsque deux clusters sont situés à la même distance de la station de base, ils ont la même taille et le même nombre de noeuds. Ces deux conditions permettent d'équilibrer la consommation d'énergie entre les clusters du réseau. L'algorithme de clustering inégal utilisant le triangle *Sierpinski* a été conçu pour minimiser la consommation d'énergie, équilibrer la consommation d'énergie entre les grappes et

équilibrer la consommation d'énergie entre les nœuds. [14] Les clusters conçus sont statiques et ont des tailles inégales. Les clusters ayant la même distance à la BS ont des tailles égales. Les CHs sont sélectionnés en fonction de leur réserve d'énergie, de leur degré et de leur distance par rapport aux centres des grappes. Les résultats de la simulation montrent que l'algorithme proposé est efficace pour équilibrer la consommation d'énergie et prolonger la durée de vie du réseau.

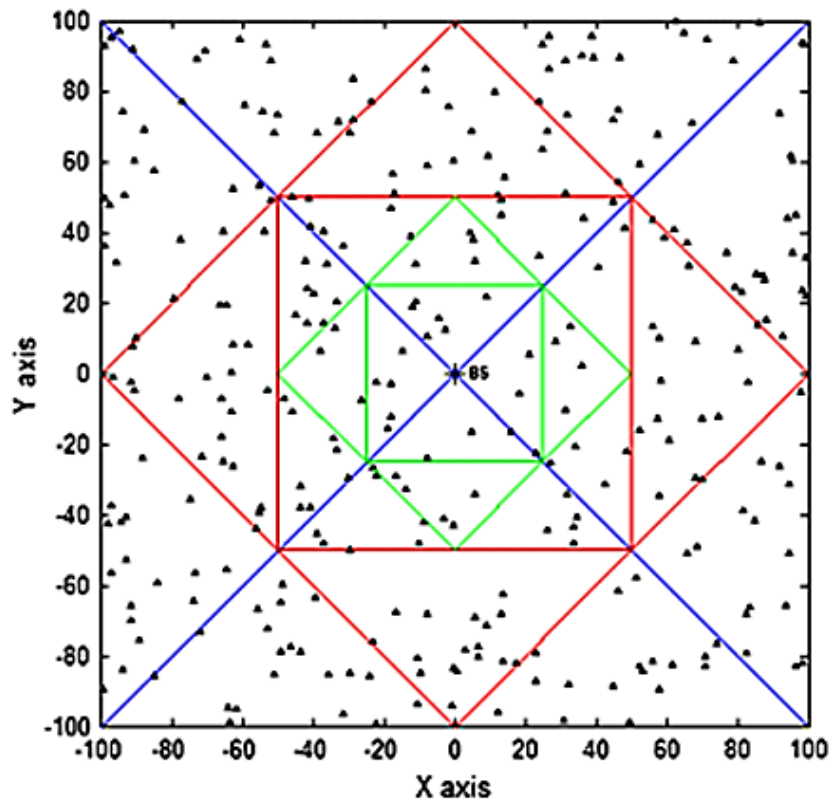


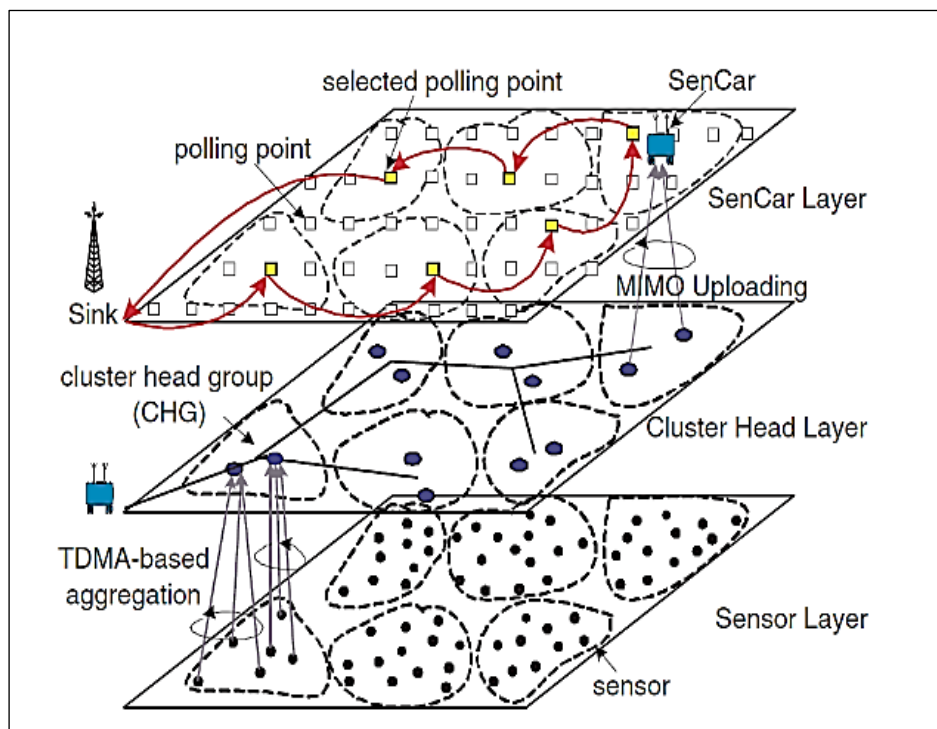
Figure 3.16 : La phase finale de protocole DDUCA [14]

#### **4.18. EE-CDRDG: An Energy Efficient Clustering with Delay Reduction in Data Gathering Using Mobile Sensor Node**

Les auteurs (Sivakumar et al., 2016) [15] proposent une approche de réduction des retards dans la collecte de données (EE-CDRDG) en utilisant la méthode de clustering hiérarchique. Le MSN collecte d'abord des données de cluster-head ayant une énergie plus faible par rapport à d'autres cluster-heads, il réduit la perte de données en utilisant le routage dynamique. Ainsi, l'algorithme proposé permet d'augmenter la durée de vie du réseau avec moins d'utilisation d'énergie pour la communication et de réduire le débordement de la mémoire tampon. Cette technique se concentre sur l'acheminement efficace de données à partir de nœuds capteurs vers la station de base. Les données sont transmises de la tête du cluster au nœud du capteur à l'aide

de la stratégie de routage des données énergétiques. Cela réduit la consommation excessive de temps.

La méthode proposée comporte plusieurs étapes de collecte de données: envoi de message d'initiation, nomination de la tête de cluster, formation de grappe, enregistrement du chemin pour SenCar, agrégation de données si un élément mobile est utilisé pour la collecte de données. Le travail proposé utilise plusieurs nœuds de capteurs pour recueillir des données qui réduisent le temps de transfert des données. C'est à cause de deux antennes qui collectent des données de deux CH simultanément et à leur tour réduit le retard s'est produit pendant la communication et utilise l'énergie efficacement. [15] Si la formation de grappe commence, il sera possible d'élire plus d'une tête de grappe pour chaque grappe, car plus d'un noeud peut avoir le même niveau d'énergie. La tête de grappe agrège les données collectées et les stocke. Après agrégation des données, la tête de cluster envoie un message de notification au noeud de capteur mobile et passe en mode veille. Le nœud de capteur mobile envoie un signal de réveil au CH. Lors de la communication avec le CH, il entre en état de veille et télécharge les données vers Mobile sensor Node. Si deux en-têtes de cluster envoient une notification en même temps, le noeud de capteur mobile peut recueillir simultanément les données des deux CH (figure 3.17).



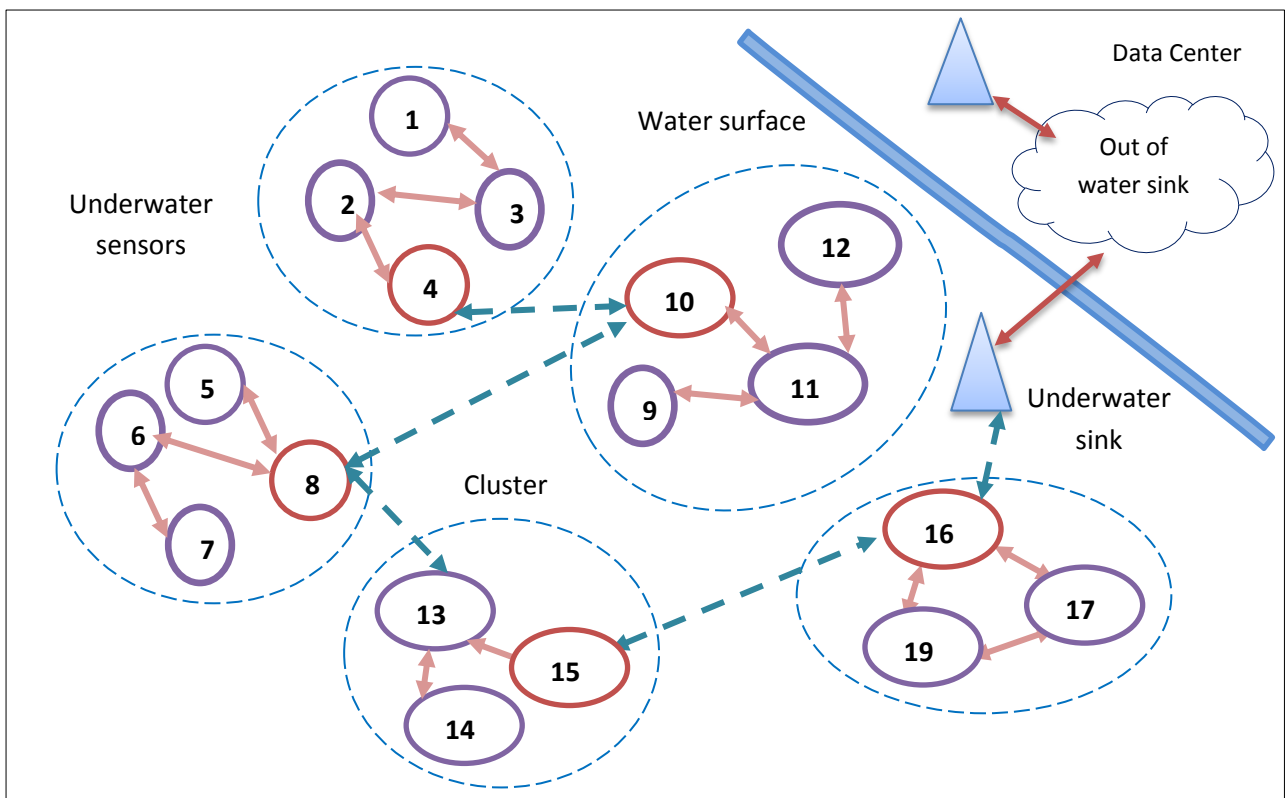
**Figure 3.17:** le schéma de protocole EE-CDRDG [15]

Le nœud du capteur mobile collecte des données dynamiquement en utilisant le schéma de routage dynamique du véhicule. Tout d'abord, le nœud du capteur mobile rassemble les

données du CH ayant un niveau d'énergie inférieur et se déplace vers un autre CH. Si le CH a un problème d'alimentation et de débordement de mémoire tampon, il enverrait un message d'avertissement au noeud de capteur Mobile pour recueillir des données immédiatement. Le noeud du capteur mobile collecte les données immédiatement et retourne à la station de base pour le téléchargement des données.

#### **4.19. Energy Efficient Architecture for Intra and Inter Cluster Communication for Wireless Sensor Networks**

Dans cet article (Goyal Et al., 2016) [16], La principale force de ce papier est de fournir une sélection optimale pour CH ainsi que des communications intra et inter-cluster optimales basées sur l'énergie et de multiples chemins. Méthode de protocole de regroupement de chemin de routage moyen minimum est implémentée pour la communication intra-cluster dans le réseau. Méthode hiérarchique de routage multi-chemin-LEACH est implémentée pour la communication entre clusters dans le réseau. La simulation expérimentale montre que le protocole proposé améliore la performance de l'UWSN en termes de délai de paquets de bout en bout, de consommation d'énergie et de débit de paquets. L'approche proposée est de nature adaptative car elle réduit la consommation moyenne d'énergie et le délai de bout en bout, améliorant ainsi le taux de livraison des paquets.



**Figure 3.18:** Architecture de communication de cluster intra et inter pour UWSN. [16]

La communication intra et inter-cluster globale peut être illustrée à l'aide de l'architecture montrée dans la (Figure 3.18), où les nœuds sont déployés de manière aléatoire dans la zone, de détecter les changements qui peuvent être destructeurs. Les informations recueillies par un seul ou plusieurs sauts seront transmis à la tête du cluster. Le CH reçoit et regroupe les données de tous ses membres. Ensuite, la tête de groupe transmet un message sommaire à l'UW-Sink qui est relié verticalement à des puits off-shore / hors de l'eau. C'est ce qu'on appelle la communication intra-cluster.

Si le CH est éloigné de l'UW-Sink, le message récapitulatif ne peut pas être transmis directement à l'UW-Sink. Pour effectuer la communication inter-cluster; Cluster head transmet des données agrégées à ses grappes voisines. Consécutivement, ces clusters transmettent les données aux clusters de leur portée, et ainsi de suite, jusqu'à ce que les données agrégées atteignent UW-Sink. [16] Les informations agrégées sont finalement transmises au centre de données via des liaisons radio ou des communications par satellite.

## **5. Avantages et Inconvénients des protocoles proposés**

### ☞ **Avantages**

- **Protocole auto-organisateur basé sur le groupement adaptatif:** les nœuds prennent leurs décisions de façon autonome et agissent de manière locale et n'ont pas besoin d'une information globale ni d'un système de localisation pour opérer de façon efficace. De plus, la collecte de données est faite périodiquement (l'utilisateur n'a pas besoin de toutes les données immédiatement). Pour exploiter cette caractéristique, ce protocole introduit un groupement adaptatif, c'est-à-dire, il réorganise les groupes après un intervalle de temps aléatoire, en utilisant des contraintes énergétiques afin d'avoir une dissipation d'énergie uniforme à travers tout le réseau.
- **Rotation des rôles de chefs de groupes:** La rotation des rôles de chefs de groupes s'avère un facteur important pour l'organisation des nœuds. Ce rôle est épuisant en termes de d'énergie car les CH sont actifs tout au long de leur élection. Puisque le nœud puits est généralement loin du champ de surveillance, les CHs diffusent une quantité plus importante d'énergie pour lui transmettre leurs données. Donc, si les CHs sont choisis d'une manière fixe, leur énergie s'épuisera rapidement ce qui conduit à leur défaillance. Par conséquent, tous les autres nœuds seront sans CH et donc inutiles. C'est pourquoi, les algorithmes de groupement (*clustering*) étudiés jusqu'ici adoptent la rotation du rôle de chefs de groupes.

- **Faible énergie pour l'accès au média:** Le mécanisme de groupes permet aux nœuds d'effectuer des communications sur de petites distances avec leurs CH afin d'optimiser l'utilisation du médium de communication en la faisant gérer localement par un CH pour minimiser les interférences et les collisions.
- **Compression locale (agrégation) :** Les CH compressent les données arrivant de leurs membres, et envoient un paquet d'agrégation au nœud puits afin de réduire la quantité d'informations à transmettre. Cela permet de réduire la complexité des algorithmes de routage, de simplifier la gestion du réseau, d'optimiser les dépenses d'énergie et enfin de rendre le réseau plus évolutif (scalable).

### ☞ **Inconvénients**

- Il est possible ne pas avoir de CHs durant un round si les nombres aléatoires générés par tous les nœuds du réseau sont supérieurs à la probabilité  $P_i(t)$ .
- Les nœuds les plus éloignés du CH meurent rapidement par rapport aux plus proches.
- L'utilisation d'une communication à un seul saut au lieu d'une communication multi-sauts diminue l'énergie des nœuds.
- Quelques protocoles ne peuvent pas être appliqués à des applications temps-réel du fait qu'il engendre une longue latence.
- La rotation des CH permet de ne pas épuiser les batteries. Cependant, cette méthode n'est pas efficace pour les grandes structures de réseaux à cause de la surcharge d'annonces engendrées par le changement des CH, et qui réduit le gain d'énergie initial.
- Il n'est pas évident que les CHs soient uniformément distribués. Donc, il est possible que les CH puissent être concentrés dans une partie du réseau. Par conséquent, certains nœuds n'auront pas des CH dans leurs voisinages.
- Aucun mécanisme de sécurité n'est intégré. Ainsi, il est très vulnérable même aux simples attaques. Donc, un attaquant peut facilement monopoliser le réseau et provoque son dysfonctionnement.

## **Conclusion**

Clustering est une technique permettant de réduire la consommation d'énergie et d'assurer la stabilité dans les réseaux de capteurs sans fil. Dans ce chapitre, nous avons présenté une analyse détaillée des principales techniques de routage dans les réseaux de capteurs et les concepts sur lesquels se basent ces techniques. Au début, nous avons étudié plusieurs techniques de clustering conçues pour les réseaux de capteurs. Nous avons conclu que le clustering imite

l'architecture centralisée et tire profits de ses avantages dans les réseaux de petite ou moyenne taille. Il permet une réutilisation spatiale des fréquences radio pour minimiser les interférences et le surcoût du trafic de contrôle. Il est bien adapté aux réseaux de capteurs puisque ceux-ci disposent de faible mémoire pour stocker toute la topologie du réseau. Nous avons constaté également qu'il est nécessaire de bien choisir la ou les métrique(s) d'élection de cluster-head pour construire des clusters plus ou moins stables. Ces métriques doivent tenir compte de la capacité des noeuds pour jouer le rôle de cluster-head et de la topologie du réseau. Toutefois, la contrainte énergie est considérée comme une contrainte forte dans les réseaux de capteurs. De plus, pour assurer un équilibre de charge entre les noeuds, les clusters formés devront être homogènes en taille i.e. tous les clusters générés doivent contenir presque le même nombre de noeuds et rayon i.e. tous les membres d'un cluster doivent être à au plus k-sauts de leur cluster-head afin de limiter le trafic de contrôle induit par l'implémentation des fonctions de routage, découverte de services. Aussi le rôle du clusterhead devra être joué pour un intervalle de temps limité et le clustering devra être distribué.

A travers ce chapitre, nous avons pu voir les différentes approches utilisées dans les réseaux hiérarchiques telles que les techniques de clustering et les modes de communication adoptés par les noeuds du réseau. Nous avons constaté également qu'il est nécessaire de bien choisir les métriques d'élection des cluster-heads pour construire des clusters plus ou moins stables. Ainsi, en analysant les principaux protocoles de routage nous avons pu dresser les avantages et inconvénients de chaque approche. Cette étude nous a permis de tirer profit des différentes techniques utilisées dans les réseaux hiérarchiques pour proposer un nouvel algorithme de routage dont l'objectif principal est le prolongement de la durée de vie du réseau tout en assurant un bon fonctionnement du réseau.

Le concept de base ainsi que l'architecture de fonctionnement de notre protocole feront l'objet du chapitre suivant.

# Chapitre 4

## Proposition du Protocole HEBM

### 1. Introduction

Le regroupement des nœuds capteurs en clusters a été largement étudié par la communauté des chercheurs du domaine, en vue de lui concéder un niveau de maturité satisfaisant, en particulier lui assurer un degré d'évolutivité (scalabilité) appréciable. Plusieurs algorithmes de clustering ont été spécialement conçus pour les réseaux de capteurs. Ces techniques de clustering proposées varient largement selon les schémas de déploiement des nœuds, l'architecture du réseau, les caractéristiques des nœuds CH et le modèle de fonctionnement du réseau. Le clustering a de nombreux avantages, il peut réduire la bande passante de communication, car elle limite la portée des interactions inter-clusters aux CHs et évite l'échange redondant des messages entre les nœuds capteurs [22]. Un CH peut planifier des activités dans le cluster de sorte que les nœuds puissent passer en mode faible puissance et par conséquent réduire le taux de consommation d'énergie. Les capteurs peuvent être engagés dans un ordre round-robin et le temps de leur transmission et réception peuvent être déterminés de sorte que la redondance de la couverture puisse être limitée et la collision lors de l'accès au support soit évitée [61].

Plusieurs protocoles de routage n'adoptent pas le Duty-Cycle car la communication intra-cluster est à un seul saut, c'est-à-dire que chaque nœud communique directement (d'après TDMA schedule) avec son CH, qui est considéré actif pendant tout le round. Il présente également une mauvaise distribution des cluster-heads [60]. Cette disparité conduit à avoir plusieurs CHs situés dans une même partie du réseau alors que d'autres parties n'en possèdent pas. Cette situation induit un accroissement de la consommation d'énergie, puisque les membres de certains clusters seront très éloignés de leur CH. Un autre problème soulevé est la taille (nombres de capteurs dans le cluster) inégale des clusters. Ce problème conduit à l'épuisement de l'énergie des CHs des clusters chargés plus rapidement que les autres. Pour remédier à ces inconvénients, nous avons proposé le protocole (HEBM : Hierarchical Clustering Energy-Balancing Multipath routing protocol for Wireless Sensor networks) ayant pour objectif une répartition équitable des CHs dans le réseau, l'équilibrage de la charge des

CHs et l'auto-organisation des nœuds de façon distribuée. Pour assurer le premier objectif important c'est-à-dire une meilleure sélection des cluster-heads, nous proposons d'élire ces derniers en fonction de 4 paramètres. Cette élection va assurer une réduction importante du nombre de clusters solitaires et offrir un gain énergétique considérable.

### 2. Motivation

Il existe plusieurs objectifs assurés par notre protocole HEBM:

- ↳ Un algorithme de clustering devrait être complètement distribué parce qu'une méthode de contrôle centralisée n'est pas pratique pour les RCSFs à grande échelle.
- ↳ Les cluster-heads devraient être bien choisis pour une répartition optimale dans la zone de surveillance afin de rendre la consommation d'énergie bien équilibrée des membres d'un cluster.
- ↳ L'algorithme de clustering devrait être efficace sur le plan énergétique.
- ↳ Un algorithme de regroupement doit appuyer l'approche d'énergie hétérogène.
- ↳ Fournir une qualité de service qui reflète adéquatement la fiabilité du réseau et les délais de latence.
- ↳ L'algorithme a l'avantage de minimiser les messages de commande de routage et les messages de contrôles et par conséquent, le protocole peut fonctionner efficacement du point de vue énergétique.

Notre proposition consiste à réaliser un routage multi-sauts inter-cluster (entre les CHs sélectionnés) et intra-cluster (entre les membres d'un cluster), afin d'améliorer le processus de routage et l'équilibrage des énergies des capteurs. Ensuite, une simulation, sous NS2, du protocole proposé afin d'évaluer ses performances, et effectuer une comparaison avec les protocoles de référence DEEAC et FEMCHRP.

### 3. Présentation de HEBM

#### 3.1. Principe de HEBM

Afin de proposer une stratégie d'acheminement, permettant de prolonger la durée de vie du réseau en équilibrant l'énergie consommée de la manière la plus équitable possible, nous proposons un protocole de clustering avec un équilibrage de la charge. Celui-ci prend en considération la capacité énergétique, mémorielle et calculatoire des capteurs lors de la sélection des cluster-heads et de l'énergie résiduelle ainsi que la distance qui sépare un nœud adjacent du nœud émetteur lors du choix du nœud relai. Nous nous intéressons à une optimisation des contraintes selon les paramètres suivants : l'énergie, la localisation, le facteur d'échelle, les ressources limitées, la bande passante limitée, l'équilibrage de la charge et la

topologie dynamique. HEBM introduit un mécanisme d'équilibrage de la charge grâce à la réalisation d'une bonne distribution des cluster-heads basés sur le modèle réseau proposé et ce, afin d'améliorer les performances en termes de durée par la réduction du signal de transmission et les messages de contrôles.

L'approche proposée vise à remplir les objectifs suivants :

- ↪ Diminution de la consommation d'énergie globale de réseau.
- ↪ Équilibrage de la dissipation d'énergie entre les nœuds capteurs et comme conséquence directe l'extension de la durée de vie du réseau.
- ↪ Équilibrage de la charge des clusters doit être effectif par l'utilisation de deux contributions : Le cluster-head temporaire et le cluster-head final.
- ↪ Les Cluster-heads sont parfaitement déterminés par une proposition d'analyse mathématique.
- ↪ Les cluster-heads sont convenablement répartis sur la zone d'intérêt.
- ↪ Le Clustering est efficace dans la complexité des messages et du temps.
- ↪ Les cluster-heads et les membres de cluster doivent être entièrement connectés selon 4 métriques proposées.
- ↪ Diminution de la latence.
- ↪ Amélioration de la fiabilité des données reçues à la station de base.

### 3.2. Modèle de réseau du protocole HEBM

Dans la suite, nous considérons un RCSF homogène, où les nœuds sont uniformément dispersés dans une zone carrée, où la longueur des côtés est représentée par "M". Nous supposons que tous les nœuds peuvent communiquer avec la BS avec suffisamment d'énergie, et ils peuvent également utiliser différents niveaux de puissance pour les communications. Les nœuds et la BS (Station de Base) sont stationnaires (figure 4.1). Tous les nœuds peuvent communiquer avec leurs voisins situés dans la portée par une communication multi-sauts. Ceci est supposé que tous les nœuds soient synchronisés au moins une fois au début de chaque phase. Par souci de simplicité, on suppose que le canal de transmission sans fil est sécurisé et que les capteurs ont des horloges synchronisées. Le processus de fonctionnement du réseau est cyclique. Chaque cycle (round) contient deux phases, la première pour la structuration du réseau et la seconde pour la transmission des données captées qui seront regroupées et acheminées vers la station de base.

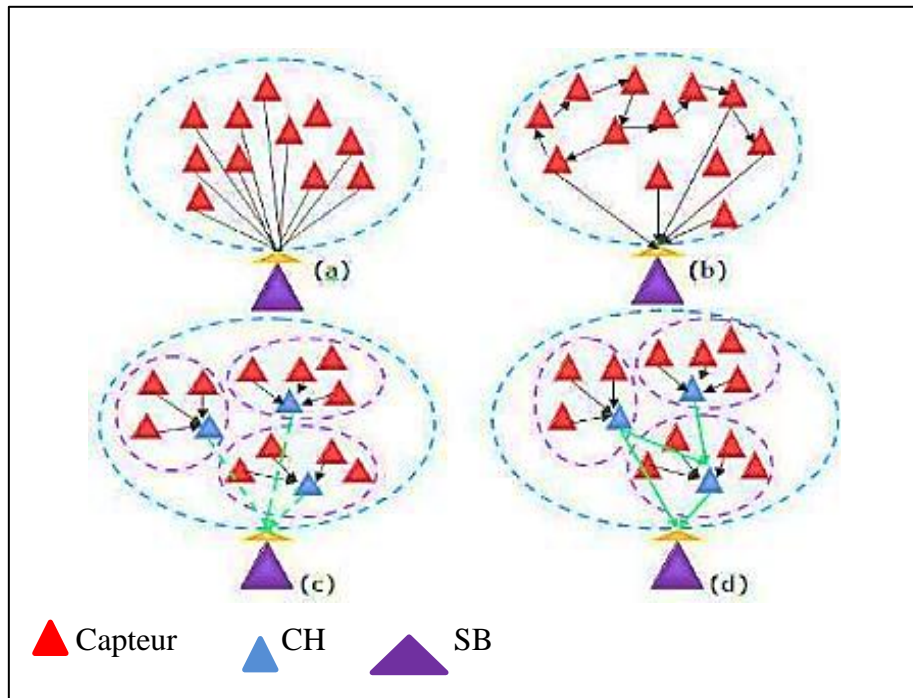


Figure 4.1 Modèles de réseau

### 3.3. Le modèle radio

Nous avons utilisé les équations suivantes [60] pour calculer la dissipation d'énergie pendant la communication. L'énergie dépensée pour la transmission des paquets de  $k$  bits sur la distance  $d$  est donnée par l'équation (4.1) :

$$E_{TX}(k, d) = kE_{elec} + kE_{amp}d^2 \quad \dots\dots\dots \text{Équation 4.1}$$

$E_{elec}$  est l'énergie nécessaire pour activer les circuits électroniques.  $E_{amp}$  est l'énergie requise pour l'amplification des signaux transmis à envoyer un bit dans un espace ouvert. La consommation d'énergie pour recevoir un paquet de  $K$  bits est calculée selon l'équation (4.2).

$$E_{RX}(K) = kE_{elec} \quad \dots\dots\dots \text{Équation 4.2}$$

L'énergie résiduelle d'un nœud, après la transmission d'un message de  $k$  bits à distance  $d$  du récepteur, est calculée en utilisant l'équation (4.3).

$$E_{ri} = E_{initial} - (E_{TX}(k, d) + E_{RX}(k)) \quad \dots\dots\dots \text{Équation 4.3}$$

Nous pouvons calculer l'énergie initiale totale du réseau en utilisant l'équation (4.4):

$$E_{initial-total} = NE_{initial} \quad \dots\dots\dots \text{Équation 4.4}$$

$E_{initial}$  : Energie initiale N: nombre des nœuds.

L'énergie moyenne restante  $E_{average}$  est obtenue par l'équation (4.5) :

$$E_{average} = \frac{\sum_{n=1}^n E_{residual}^{(i)}}{n} \quad \dots\dots\dots \text{Équation 4.5}$$

$E_{round}$  est l'énergie totale dissipée dans le réseau au cours d'un round, qui est donnée par l'équation (4.6):

$$E_{round} = \lambda [NE_{DA} + 2NE_{elect} + NE_{fs}d_{neigh}^2 + E_{amp}d_{CHtoBS}^4] \quad \dots\dots\dots \text{Équation 4.6}$$

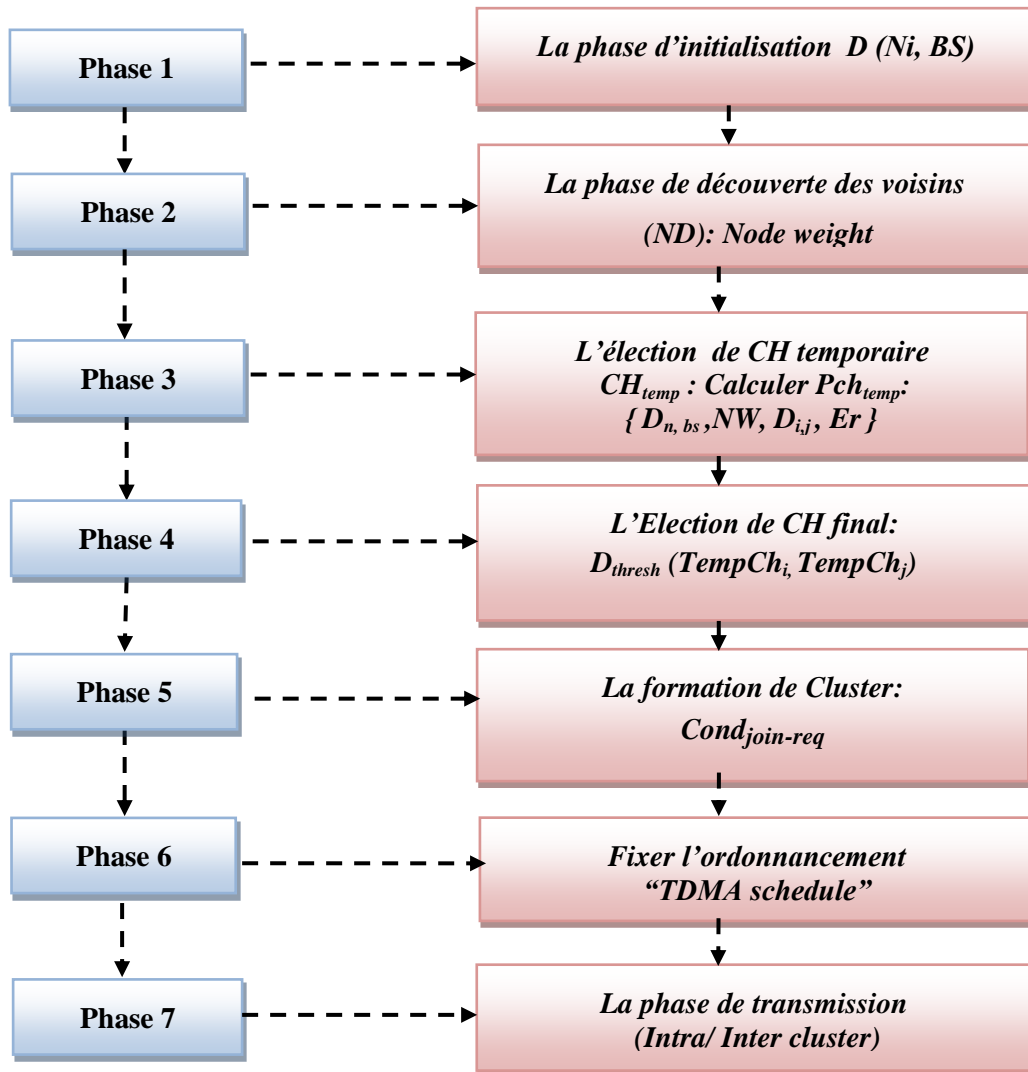
Où  $E_{DA}$  est le coût d'agrégation de données passé dans chaque nœud,  $d_{CH toBS}$  est la distance moyenne entre le Cluster-Head et la BS,  $d_{neigh}$  est la distance moyenne au nœud suivant dans la chaîne,  $\lambda$  est la taille totale des données transmises,  $E_{fs}$  et  $E_{amp}$  dépendent du modèle d'amplificateur utilisé.

#### **4. Phases de HEBM**

L'un des facteurs importants qui permettent d'améliorer la durée de vie d'un réseau de capteurs sans fil est la conception du réseau. Dans cette section, nous décrivons l'approche HEBM proposée. Ce protocole utilise un schéma de regroupement adaptatif. Un schéma de regroupement est appelé un schéma d'adaptation si au fil du temps, le nombre de clusters varie et les membres des cluster-heads évoluent. La BS est supposée avoir une énergie illimitée et une puissance importante de communication. Il est également supposé que la station de base est située à une position fixe, soit à l'intérieur ou hors du champ de captage [28]. L'algorithme du protocole HEBM se déroule en « rounds » qui représentent des intervalles de temps déterminés au préalable où chaque round est constitué de 07 phases. Les nœuds avec haute condition (Pch) peuvent agir en tant que CH (Cluster-Head) pour agréger et envoyer les données. Tous les nœuds doivent alternativement devenir CH, la sélection du CH est proposée selon de nombreux facteurs. Au niveau du HEBM, les facteurs suivants sont considérés: le nombre des voisins d'un nœud, l'énergie résiduelle, condition de distance entre les nœuds, condition de distance avec la SB.

Le protocole HEBM réalise une bonne répartition des clusters (problème non résolu dans de nombreux protocoles). Comme on le remarque sur la (figure 4.2), le protocole HEBM se déroule en « cycles » qui représentent le temps d'intervalles déterminés à l'avance. Chaque round se compose de plusieurs phases, la phase d'initialisation, la phase de découverte des voisins, la phase d'élection de CH temporaire, la phase d'élection de CH final, la phase de

formation de cluster, la phase de fixer le TDMA et la phase de transmission (inter-cluster et intra-cluster).



**Figure 4.2** Les phases de HEBM

### 4.1 Phase d'initialisation du HEBM

La phase d'initialisation, présentée dans la figure 4.3, consiste à diffuser un message d'annonce "BS-Msg" par la station de base, à tous les nœuds capteurs du réseau. Ainsi, chaque nœud peut calculer approximativement la distance qui le sépare de la station de base, reposant sur la force du signal reçu. Pendant cette phase chaque nœud calcule un paramètre qui est défini comme suit:  $D(N, BS)$ .

- L'indicateur de puissance du signal reçu peut être utilisé pour estimer la distance entre deux nœuds sur la base de la force du signal reçu par un autre nœud. Dans HEBM, un nœud émetteur envoie un signal avec une force déterminée qui s'affaiblit lorsque le signal se propage (voir Figure 4.4)

- Théoriquement, la puissance du signal est inversement proportionnelle à la distance et un modèle de propagation radioélectrique connu peut être utilisé pour convertir la force du signal en distance.

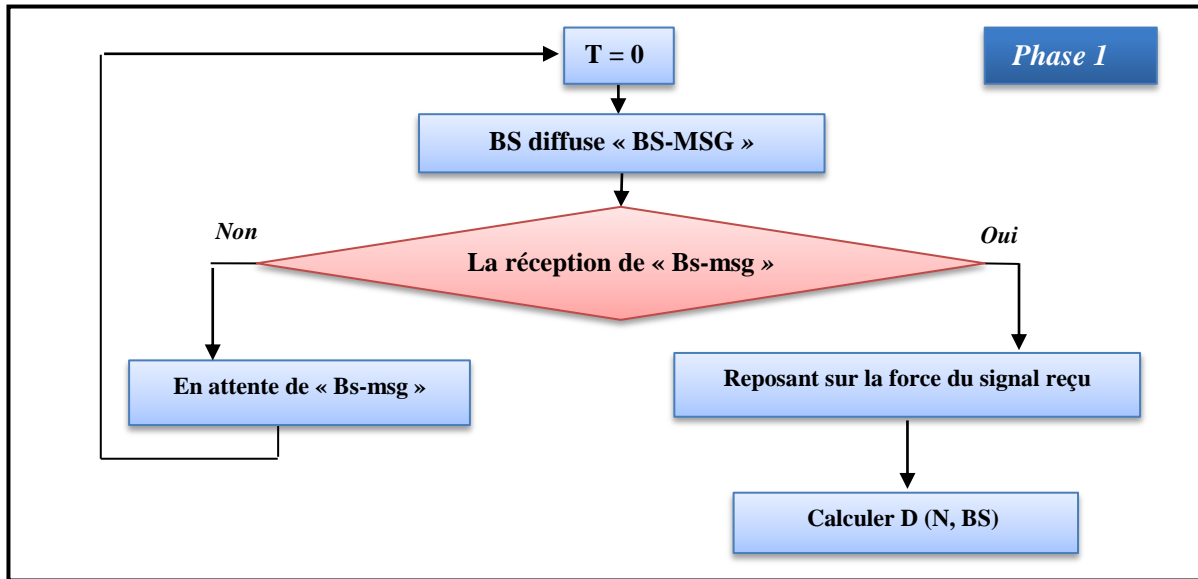


Figure 4.3 La distance  $D(N,BS)$

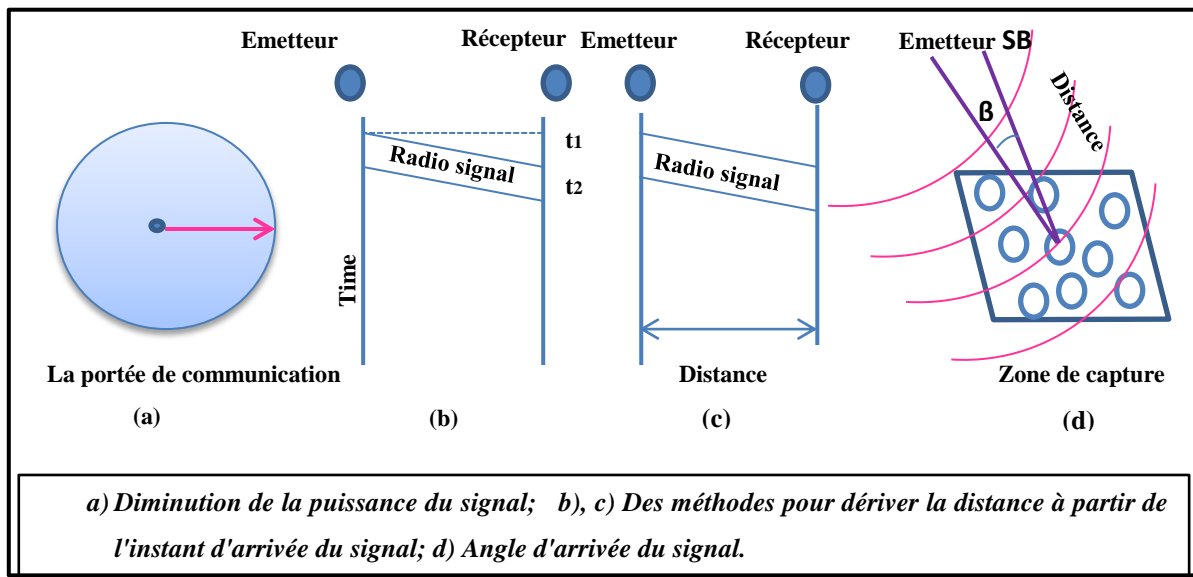


Figure 4.4 Dérivation de la distance à partir de l'instant d'arrivée du signal

Le RSSI (*Received Signal Strength Indicator*) est une sortie du récepteur du signal radio. Il permet de mesurer la puissance d'un signal reçu par un équipement. Sachant que la perte de puissance entre un émetteur et un récepteur est proportionnelle à la distance parcourue, le récepteur peut avoir une estimation de la distance le séparant de l'émetteur en se basant sur le RSSI et la puissance du signal émis.

### 4.2 Phase de découverte des voisins

Le processus de découverte de voisinage est réalisé par un msg « *Discov-neigh-msg* ». Ce dernier permet de construire et de maintenir des tables de voisinage grâce à un échange périodique de messages contenant nécessairement l'identifiant du nœud émetteur et éventuellement sa position. Ce processus dépend de plusieurs paramètres, par exemple la fréquence d'émission des paquets, la puissance de transmission, etc. On a introduit des mécanismes un peu plus complexes comme l'ordonnancement des activités et l'introduction de périodes de sommeil. L'idée est de mettre en veille périodiquement les nœuds afin d'économiser leur énergie. Ces mécanismes introduisent plusieurs nouveaux paramètres, comme la durée des périodes d'activité et de sommeil, qui peuvent avoir un impact considérable sur les performances et l'efficacité. En effet, ces paramètres peuvent avoir une influence non négligeable sur la durée du processus de découverte de voisinage, c'est à dire le temps nécessaire avant que chaque nœud ne découvre tous les autres nœuds présents dans son voisinage. Toutes ces problématiques doivent donc être considérées lors de la conception d'un protocole de découverte de voisinage.

Dans cette phase, chaque nœud, afin de calculer son « node-weight », il envoie un message «*Discov-neigh-msg*» (la force du signal dépend de la distance qui représente le rang ou le diamètre du cluster désiré), qui contient son identificateur. Tout nœud ayant reçu le message, émet immédiatement un «*Discov-neigh-msg* » de même type, alors chaque membre dispose de sa table de voisins, lui permettant de savoir son coût qui est la taille de cette dernière. Donc, chaque nœud envoie son identificateur avec son coût à ses voisins, qui font de même, à la fin de cette phase chaque nœud dispose d'une table qui contient l'identificateur et le coût de chacun de ses voisins. Cette procédure doit être exécutée à chaque round, ce qui génère un grand trafic à cause du nombre des messages circulant dans le réseau et produit une perte énergétique (voir figure 4.5 et 4.6)

Une solution proposée (qui minimise la surcharge des messages qui circulent dans le réseau et implique une optimisation de l'énergie des capteurs) consiste à ne pas découvrir le voisinage à chaque fois que le processus de clustering est déclenché, parce que les nœuds ne meurent pas de façon inattendue, et l'ensemble des voisins ne change pas très souvent. Au lieu de construire des nouvelles listes à chaque round, les nœuds mettent automatiquement à jour l'ensemble de leurs voisins en envoyant périodiquement des messages «*Discov-neigh-msg* ». On enregistre les listes remplies dans le premier round, et on fait la mise à jour de ces dernières au début de chaque round. À partir du deuxième round, chaque nœud dont l'énergie

a atteint un certain seuil (l'énergie minimal qui permet à un nœud de continuer le round courant mais pas le prochain), émet un message d'avertissement contenant son identificateur. Ce message indique que ce nœud ne sera pas fonctionnel au prochain round. Tout nœud ayant reçu ce message, supprime l'identificateur de l'émetteur de la liste des voisins, comme indiqué dans la figure 4.7, et il ne sera pas pris en considération lors du choix du ClusterHead dans la prochaine phase. En outre, HEBM prolonge la durée de vie du réseau, ce qui favorise sa stabilité.

```
# Distance D (Ni, Bs)
(1) La SB diffuse "BS-msg".
(2) Si Ni reçoit "BS-msg" Alors
(3) Ni Calcule D (Ni, Bs)
(4) Sinon
(5) Attendre  $t_{wait}$  (s) pour recevoir "BS-msg"
(6) FinSi
# Node weight.
(7)  $\forall i, N_i$  envoie "Disc-neigh-msg" in  $R_{thd}$ :
(8) Si  $N_j$  reçoit "Disc-neigh-msg" Alors
(9) Ajouter l'identificateur "id" et diffuser "Disc-neigh msg";
(10) Création de table des voisins "neigh-tab" ;
(11) Weight-node = la longueur de "neigh-tab" ;
(12) Sinon
(13) Attendre  $t_{wait}$  (s) pour recevoir "Disc-neigh-msg";
(14) Fin Si
```

**Figure 4.5** Pseudo Algorithme de la phase d'initialisation du protocole HEBM

L'objectif de cette phase est de recueillir autant d'informations possibles des autres nœuds dans le voisinage afin de fournir une bonne base pour la décision du choix des voisins. Au début, le découvreur transmet un message « Discov-neigh-msg », au moment où le découvreur transmet la première diffusion, un 'timer' « ts-discov » est démarré simultanément. pendant que la minuterie est en cours d'exécution, le nœud attend pour diffuser des messages reçus, un nœud recevant une notification ajoute la source comme voisin du voisin. Comme réaction à la réception d'une notification d'acquisition, le découvreur ajoute ainsi la source comme un voisin. Après avoir reçu toutes les notifications, le découvreur construit une table des voisins et donc la découverte du chemin est terminée. Et la découverte des voisins est complétée par la construction de la table finale, si elle pourrait être remplie avec des voisins ou restée vide.

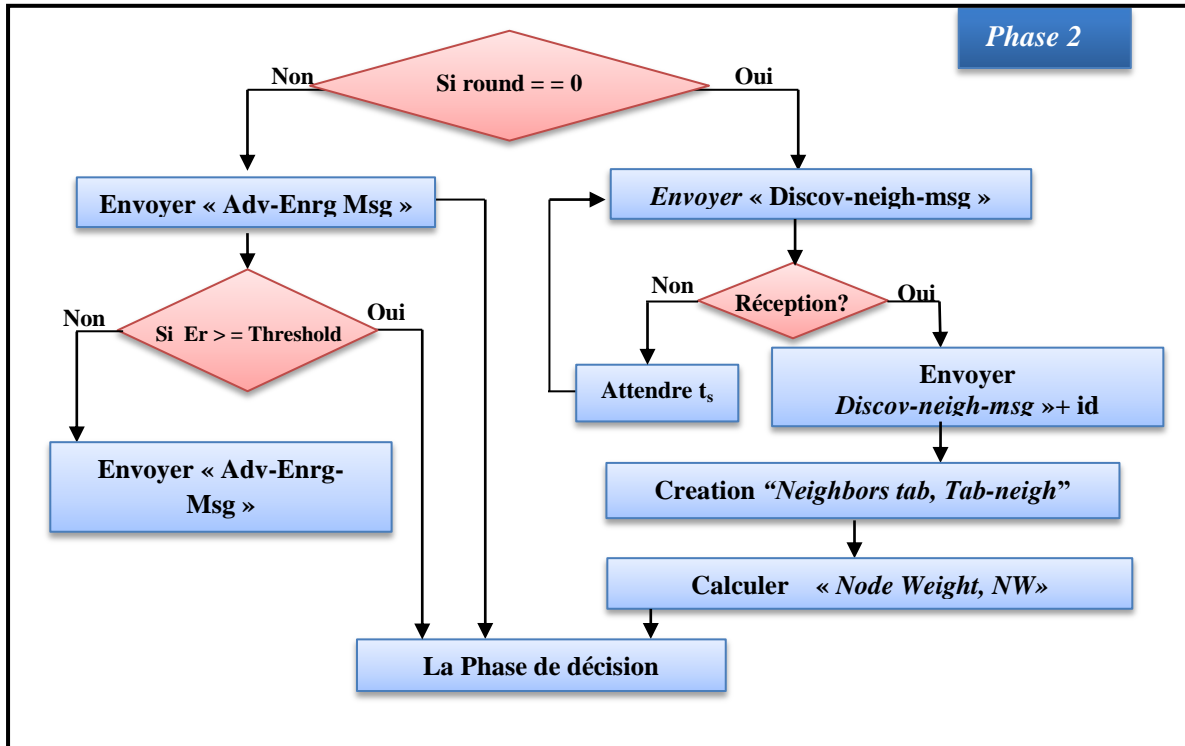


Figure 4.6 Découverte du voisin (Node Weight)

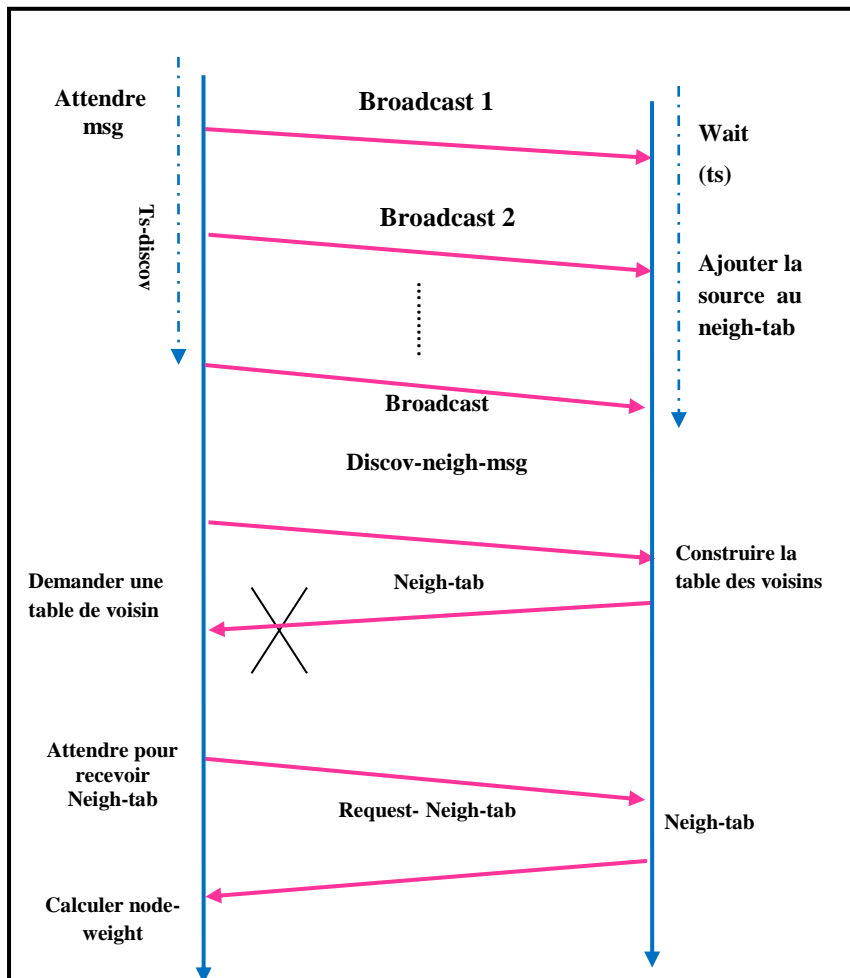


Figure 4.7 La création de table des voisins et le calcul 'node-weight'

### 4.2.1 Energie résiduelle

Elle est exprimée par la différence entre l'énergie courante et l'énergie instantanée consommée par un capteur. L'énergie d'un capteur est mise à jour par l'équation 4.7 suivante :

$$E_r = E_c - E_x \quad \dots\dots\dots \text{Équation 4.7}$$

$E_x = E_e$  si le capteur envoie un message.

$E_x = E_r$  si le capteur reçoit un message.

### 4.2.2 Distance entre les nœuds voisins

Initialement les nœuds sont déployés aléatoirement dans une zone d'intérêt et ils sont repérés par les coordonnées (x, y). Ces coordonnées nous permettent de calculer les distances euclidiennes entre les nœuds.

C'est l'affectation de deux valeurs aléatoires (x et y) à chaque capteur sur un plan à deux dimensions. Ces valeurs sont comprises entre 0 et la taille du terrain.

Soient les nœuds A et B ayant respectivement les coordonnées suivantes ( $x_a, y_a$ ) et ( $x_b, y_b$ ) alors la distance entre ses deux nœuds est calculée par l'équation (4.8).

$$dist(A, B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad \dots\dots\dots \text{Équation 4.8}$$

Dans HEBM, les nœuds ont la même portée de transmission et de réception (R). D'où si la distance euclidienne entre deux nœuds A et B est inférieure ou égale à R alors ces deux nœuds sont considérés comme voisins. Après la découverte des voisins par chaque nœud, alors chacun d'eux établit la table des distances de la manière suivante : si le nœud est voisin alors cette entrée de la table est initialisée par (A, B) sinon par la valeur infinie.

### 4.3 Phase d'élection de CH temporaire

La phase d'élection du cluster-head appelée aussi la phase Set-up, on utilise une combinaison de métriques (le degré de connectivité, l'énergie restante et la distance) pour choisir un nœud leader qui est le cluster-head temporaire.

Dans notre proposition, les nœuds concourent dans une compétition afin d'élire le candidat CHtemp de cluster ensuite après on utilise le paramètre de distance pour définir le cluster-head final. L'élection CH temporaire, final et la formation de cluster dans le protocole HEBM ont les objectifs suivants:

- Prolonger la durée de vie du réseau en répartissant la consommation d'énergie, minimiser la surcharge des messages de contrôles, le processus de regroupement est

dans un nombre d'itérations constant, et les cluster-heads sont bien distribués sur la surface de réseau.

- Minimiser l'énergie consommée par chaque nœud dans la phase de transmission et ainsi prolonger la durée de vie du réseau.
- Atteindre une bonne connectivité du réseau et la maintenir durant la durée de vie.
- Proposer une méthode de contrôle de puissance qui peut être intégrée dans différents protocoles de routage.
- Proposer une méthode d'élection des cluster-heads.

La condition d'un nœud pour être un CHtemp "Pch", est déterminée proportionnellement en utilisant une combinaison de quatre mesures:

- Distance entre le nœud et la station de base.
- Energie résiduelle du nœud et de ses voisins.
- La distance entre les nœuds voisins.
- Node-Weight: nombre de voisins de nœud.

Nous avons proposé les équations suivantes (*Equations 9-10-11-12*) pour élire le Cluster Head temporaire:

$$\beta_1(i, j) = 1 - \alpha_1 \left(1 - \frac{D_{Bs,i}}{D_{Bs,j}}\right) \quad \dots\dots\dots \text{Équation 4.9}$$

$D_{Bs,i}$ : la distance entre le nœud "i" et la station de base BS.

$D_{Bs,j}$ : distance entre le nœud voisin "j" et la Station de base BS.

$$\beta_2(i, j) = 1 - \alpha_2 \left(1 - \frac{NW_i}{NW_j}\right) \quad \dots\dots\dots \text{Équation 4.10}$$

$Nw_i$ : nombre de voisins du nœud «i».

$Nw_j$ : nombre de voisins du nœud voisin «j».

$$\beta_3(i, j) = 1 - \alpha_3 \left(1 - \frac{E_i}{E_j}\right) \quad \dots\dots\dots \text{Équation 4.11}$$

$E_r, i$ : Energie résiduelle du nœud «i».

$E_r, j$ : Energie résiduelle du nœud voisin «j».

$$Pch_{(i,j)} = \text{Max} \left[1 - \sum_{i,j=1}^n \beta_1(i, j), \beta_2(i, j), \beta_3(i, j)\right] \quad \dots\dots\dots \text{Équation 4.12}$$

$Pch(i, j)$ : une condition pour être un cluster-head temporaire «CHtemp».

$\alpha_1, \alpha_2, \alpha_3$ : coefficient constant «0» ou «1».

Chaque nœud de capteur ( $N_i$ ) du réseau calcule sa condition  $Pch(i, j)$  proposée, puis il diffuse un message, appelé *CH-ADV*, vers d'autres nœuds. Ce message inclut l'ID du nœud et la

valeur de la condition Pch. Dans le schéma de compétition proposé, le paramètre  $R_{comp}$  spécifie le rayon d'un cluster. Ce dernier est un paramètre fixe du système réseau, cette valeur devrait être raisonnable, c'est-à-dire qu'elle ne doit pas être ni trop longue ni trop petite pour ne pas surcharger le réseau. Comme indiqué sur la (Figure 4.8), le nœud  $N_i$  attend pendant  $t_{wait}$  et il reçoit le message Pch-msg de tous les voisins (voir Figure 12). Notons que le temps d'attente ne devrait pas être trop court car certains nœuds n'ont pas reçu le message Pch-msg, et il ne devrait pas être trop long car il augmente la complexité du temps. Ensuite, il compare sa valeur de condition Pch avec celle de ses voisins. Si le nœud a trouvé la valeur de sa condition Pch supérieure à la valeur Pch de tous ses voisins, il deviendra un cluster-head temporaire. Sinon, il envoie un message de jointure au voisin qui a la plus haute condition Pch pour devenir membre du Cluster.

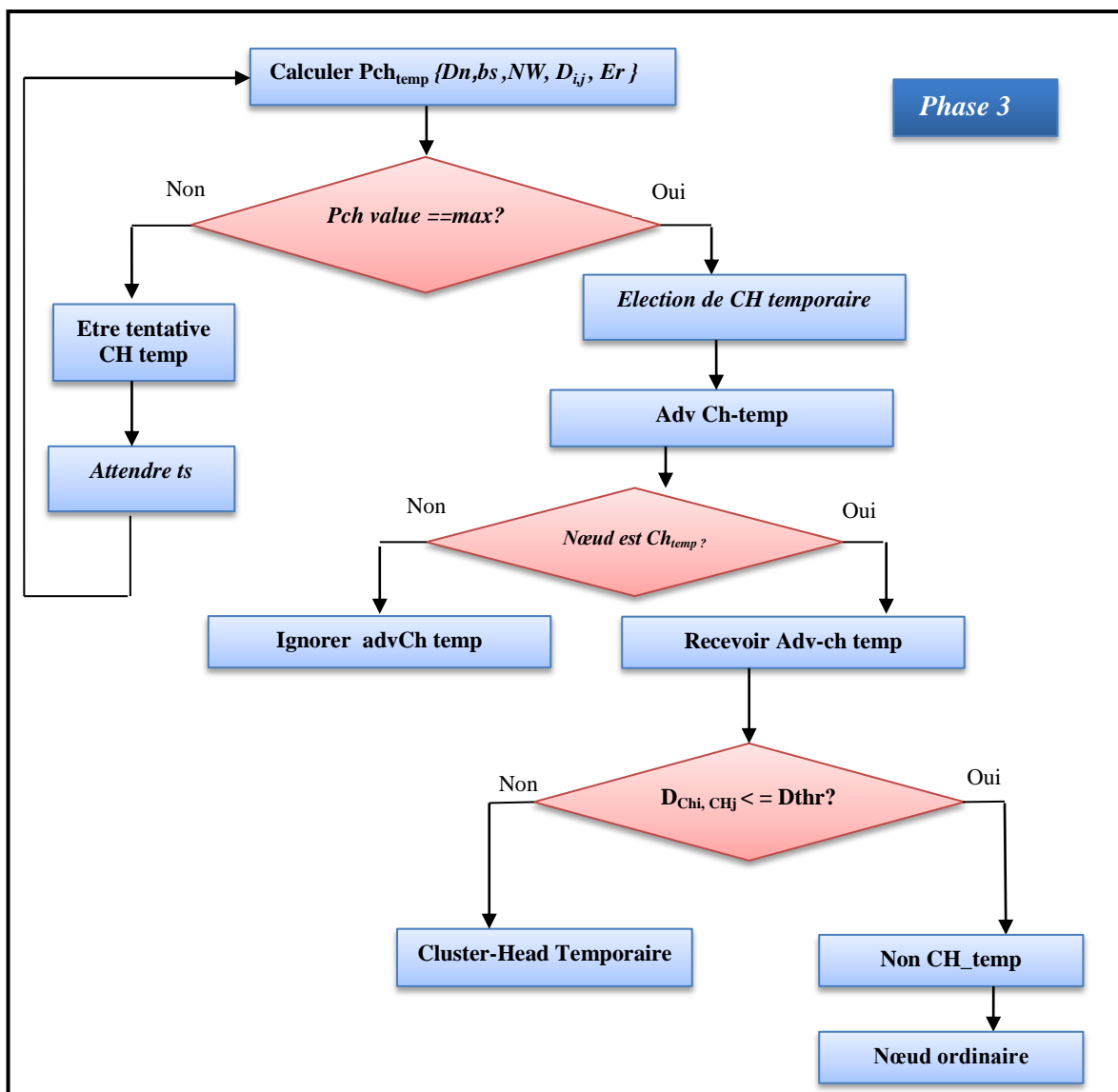


Figure 4.8 Election du clusterhead temporaire

#### 4.4 Phase d'élection de cluster-head final

Sur les figures (4.9 et 4.10), chaque  $CH_i$  temporaire doit diffuser un message  $CH-ADV$  contenant son  $ID_i$  de nœud et la probabilité  $Pch(i)$ . Lorsqu'un  $CH_j$  temporaire reçoit ce message, il calcule la distance entre l'émetteur  $CH_i$  et lui-même. Si cette distance est supérieure ou égale à une distance seuil,  $D_{thd}$ , il ignore le message, mais si la distance ( $d_{chi}, d_{chj}$ ) est inférieure à  $D_{thd}$  et si il trouve la valeur  $Pch(i)$  d'émetteur est supérieure à son propre  $Pch(j)$ , alors le récepteur  $CH_j$  temporaire devient un nœud ordinaire. Il envoie alors un message  $Join-msg$  vers le cluster-head émetteur  $CH_i$ . Si deux  $CH$  temporaires sont sur le même niveau, la distance entre eux est inférieure au seuil  $D_{thd}$  et ils ont le même  $Pch$ , alors le  $CH$  temporaire avec la valeur de  $Pch$  la plus élevée sera élu comme Cluster-head final (voir figures 4.11 et 4.12).

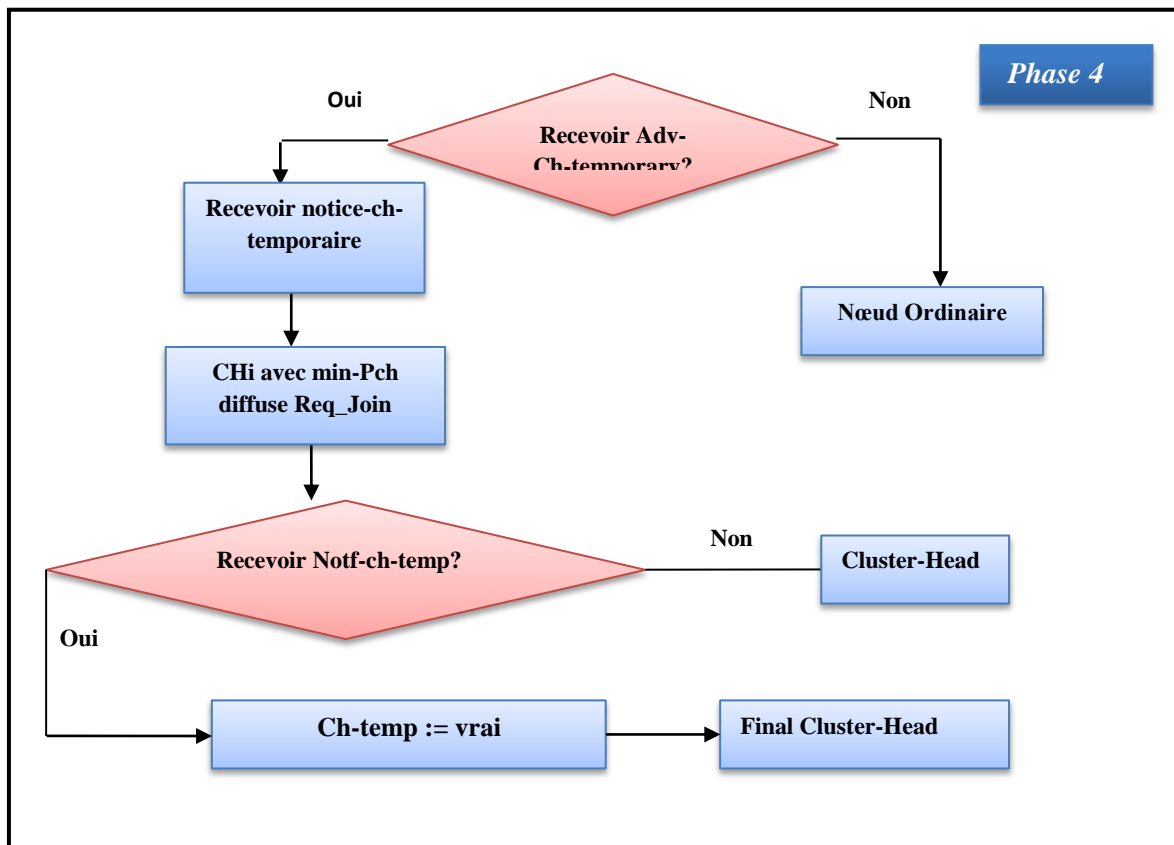


Figure 4.9 Election du cluster-head final

Le rôle de chaque  $CH$  est de réaliser les trois tâches suivantes. La première tâche est de recueillir périodiquement les données captées des nœuds du cluster et agréger les données dans le but d'éliminer les redondances des valeurs corrélées. La deuxième tâche du clusterhead est de générer un programme d'accès multiple par répartition dans le temps par lequel les nœuds de capteurs reçoivent un intervalle de temps pour la transmission des

données. La troisième consiste à transmettre les données agrégées au CH voisin ou directement à la station de base.

Ainsi, le rayon du cluster est déterminé par la puissance de transmission utilisée pour découvrir les voisins, qui est proportionnelle à la distance utilisée lors de l'envoi du message « Discov-neigh-msg ». Plus la puissance de transmission (distance) est élevée, plus le domaine de couverture d'un cluster augmente. La puissance de transmission détermine aussi le nombre de clusters : plus la distance de transmission est petite, plus le nombre de clusters augmente. Cette étape prend un temps, qui devrait être suffisamment long pour recevoir des messages de tout voisin dans le cluster. L'objectif principal de notre protocole HEBM est de prolonger la durée de vie du réseau. Pour cela, la sélection des chefs est principalement basée sur l'énergie résiduelle de chaque nœud. Pour accroître l'efficacité énergétique, trois autres paramètres secondaires sont considérés : le coût des voisins, la distance entre les voisins, la distance entre lui-même et la station de base.

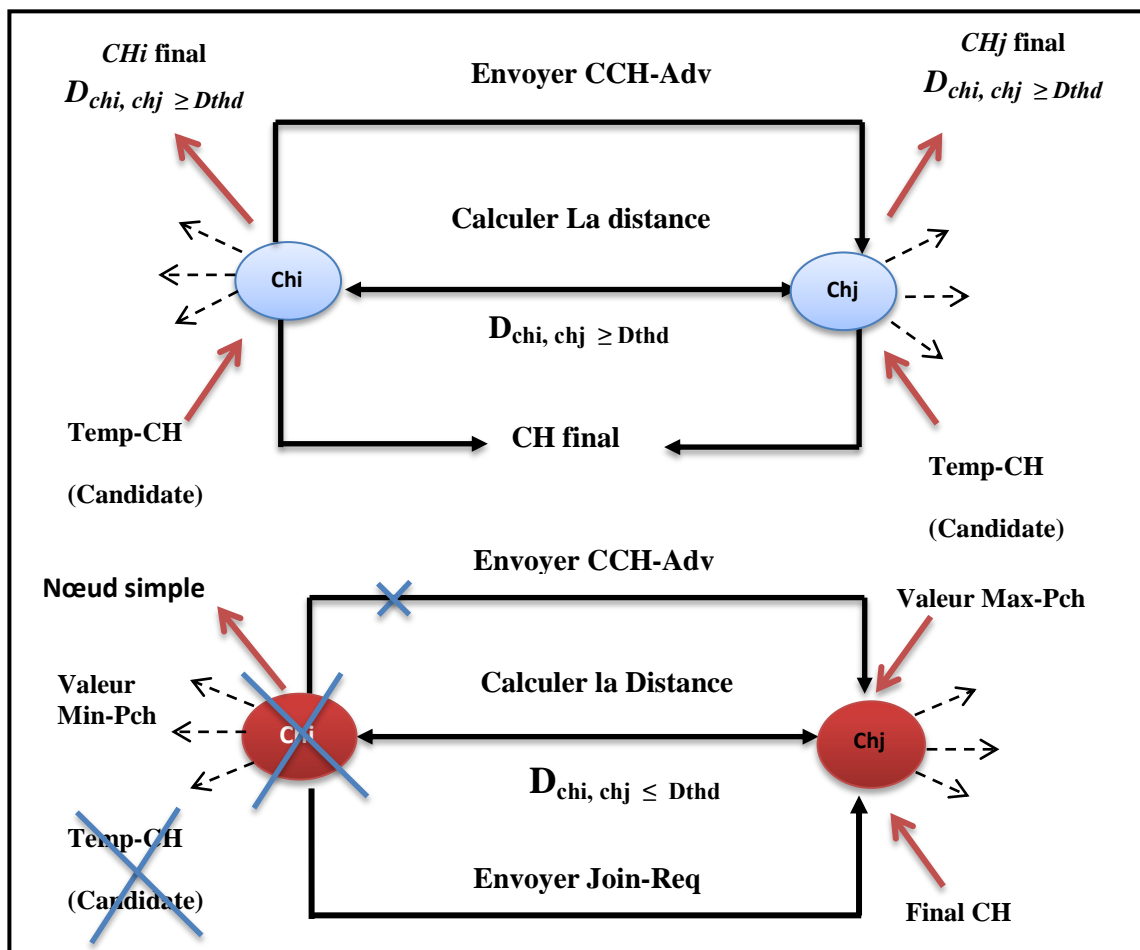


Figure 4.10 Différence entre Cluster-head temporaire et cluster head final

- (1)  $\forall i, j$  "Ni" envoie msg {WN, Er, Dbs,  $D_{i-j}$ };
- (2)  $\forall i$ , "Ni" calcule la condition  $P_{ch(i)}$ ;
- (3) **Evaluer** la condition Pch (i);
- (4) **Pour** chaque nœud voisin
- (5) Si  $\forall i, j; Pch(i) > Pch(j)$  Alors
- (6)  $Ni \leftarrow CHtemp(i)$ ;
- (7) **Sinon**
- (8)  $Nj \leftarrow CHtemp(j)$ ;
- (9) **Attendre**  $t_{wait}$  (s) pour CCH-ADV message
- (10) **Envoyer** a Join-Req message
- (11) **FinSi**

Figure 4.11 Pseudo algorithme de la phase de décision

- # La condition de distance  $D(CH_{temp(i)}, CHtemp(i))$
- (1) **Si** Ni est un CH temporaire Alors
  - (2) Attendre  $t_{wait}$  (s) pour recevoir CCH-ADV message
  - (3) **Si** Ni reçoit CCH-ADV Alors
  - (4) **Evaluer** le message reçu de Nj
  - (5) **Calculer** la distance D (CHtemp(i), CHtemp(i))
  - (6) **Si**  $\forall j, i \sqrt{(Xcch(i) - Xcch(j))^2 + (ycch(i) - ycch(j))^2} \geq Dth$  Alors
  - (7) **Envoyer** un message CH-ADV dans Rc
  - (8) **Sinon**
  - (9) **Envoyer** *Join-Req* message au cluster-head temporaire Max-pch
  - (10) **FinSi**

Figure 4.12 Pseudo algorithme de sélection d'un CH temporaire et finale selon la condition de distance

#### 4.5 Phase de formation des clusters

Dans cette phase, nous avons proposé une métrique pour la formation des clusters, nous avons pris en considération la valeur « Pch » et la « distance par rapport au SB ». Chaque nœud ordinaire choisit son cluster head et l'informe de sa décision par un message d'appartenance

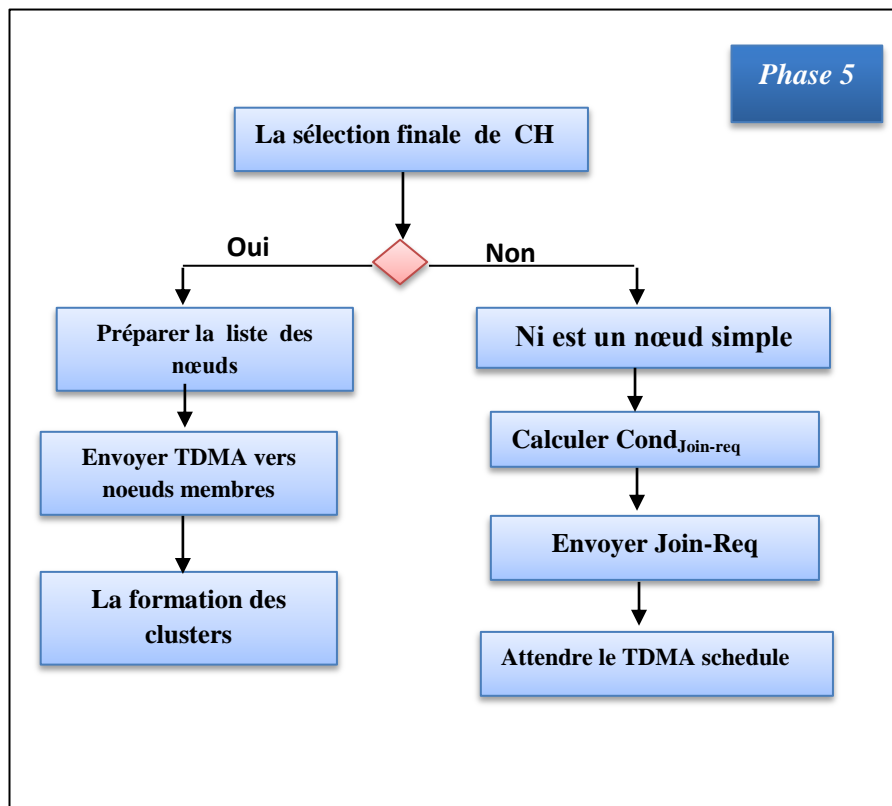
de type  $Join_{cond}(ch,j)$ . Les nœuds non CH doivent estimer  $Join_{cond}(i, CHj)$ , par l'équation proposée (équation 4.13), puis en envoyant un Join\_Msg à leur CH respectif.

$$Join_{cond}(ch,j) = \text{Max} \left( 1 - \frac{P_{chj}}{D_{i,ch} + D_{chj,bs}} \right) \quad \dots \dots \dots \text{Équation 4.13}$$

Où:  $D_{chj, bs}$  est la distance entre  $chj$  et la SB.  $D_{i, chj}$  est la distance entre le nœud  $i$  et  $chj$ .

$Join_{cond}(ch, J) \rightarrow \text{Max}$ ,  $P_{chj} \rightarrow \text{Max}$ , et  $D_{chj, bs}$  et  $D_{i, CHj} \rightarrow \text{Min}$ .

Cette équation permet au nœud de choisir le CH qui a le plus grand  $P_{ch}$  et qui est le plus proche de la BS. Cette proposition permet au HEBM d'optimiser l'utilisation des ressources pour économiser de temps et l'énergie, ce qui améliore également l'évolutivité: Comme le clustering aide à organiser les réseaux à grand échelle selon les exigences spécifiques de l'application, les tâches et les ressources nécessaires peuvent être ainsi distribuées et optimisées (voir figures 4.13 et 4.14). En se basant sur la méthode d'ordonnancement des tâches MAC-TDMA, le protocole met en œuvre la méthode d'accès au médium en assignant à chaque nœud ordinaire un intervalle temporel de communication. Chaque nœud peut activer la tâche de mise en veille pour optimiser son énergie. Les différents CHs réalisent l'agrégation et la fusion de l'ensemble des données perçues puis les transmettent vers SB.



**Figure 4.13** Phase de formation des clusters

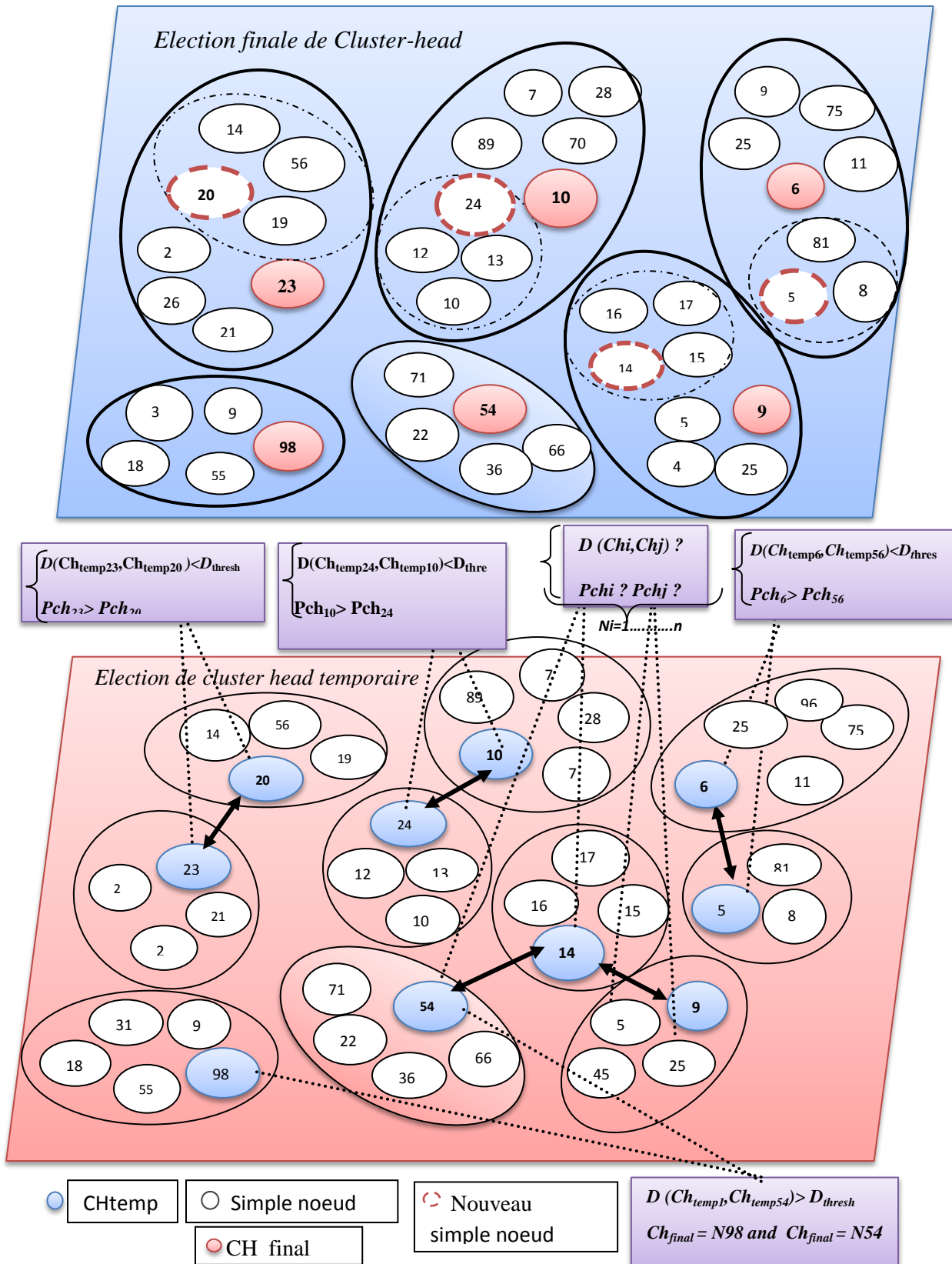


Figure 4.14 Re-clustering de HEBM

### 4.6 Phase d'ordonnancement

Concernant les CHs, chacun agit comme un centre de commande local pour coordonner les transmissions des données au sein de son cluster. Il crée un ordonnanceur (schedule) TDMA (Figures 4.15 et 4.16) et assigne à chaque nœud membre un slot de temps durant lequel, il peut transmettre ses données. L'ensemble des slots assignés aux nœuds d'un groupes est appelé frame. La durée de chaque frame diffère selon le nombre des membres du groupe.

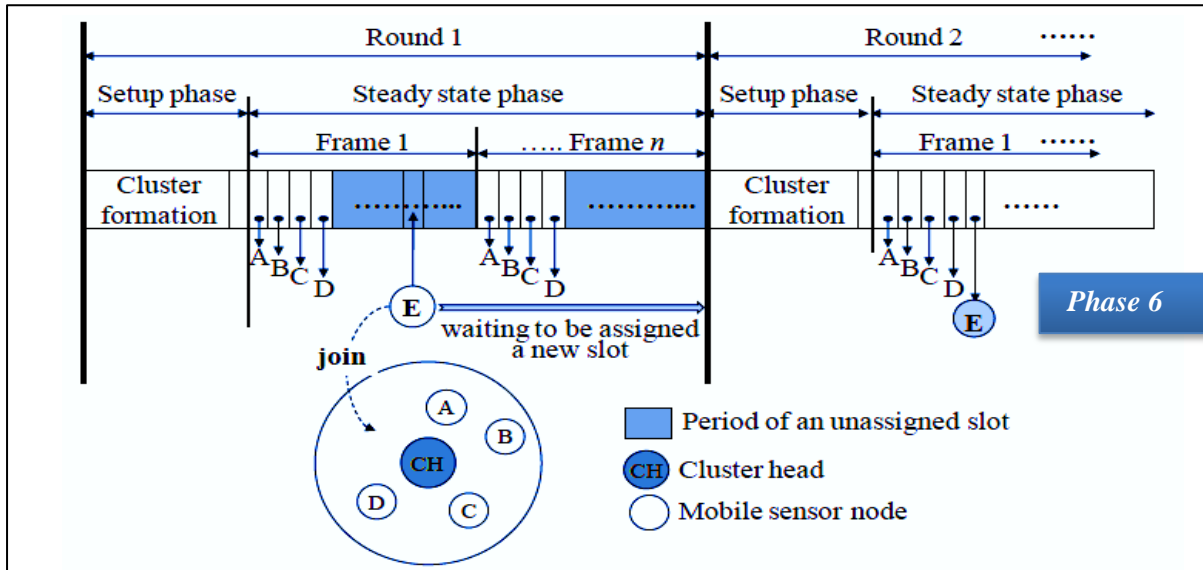


Figure 4.15 TDMA et les différents rounds de HEBM

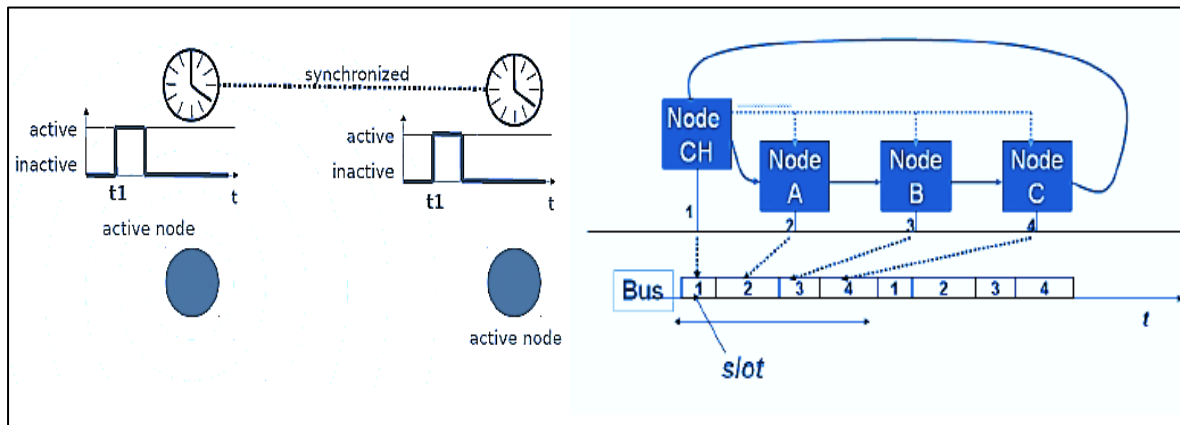


Figure 4.16 Accès multiple par répartition de temps du HEBM

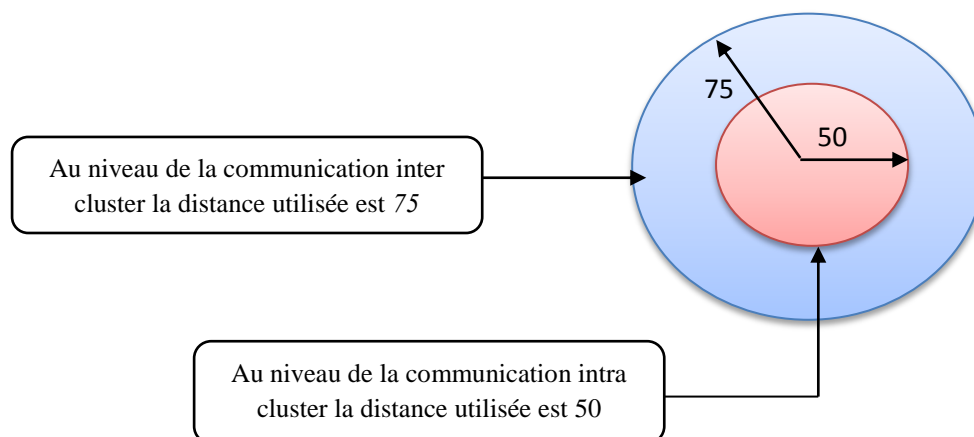
Cette phase est plus longue que la phase précédente, et permet la collecte de données captées. En utilisant l'ordonnanceur TDMA, les membres émettent leurs données captées pendant leurs propres slots. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leurs énergies. Ces données sont ensuite agrégées par les CHs qui les fusionnent, les compressent, et envoient les résultats finales à la SB.

### 5. Phase de transmission

La solution retenue le plus communément pour organiser un très grand RCSF est de regrouper les nœuds en clusters. Ce type d'organisation de communication basée sur un routage intra-cluster et inter-cluster permet de réduire le nombre de nœuds participant à des communications sur de longues distances. Chaque cluster de nœuds est identifié par un CH, permettant de coordonner les activités de son groupe comme le routage des données, leur agrégation, la synchronisation, etc. Les nœuds membres d'un cluster peuvent être actifs ou bien au contraire endormis (afin de préserver leur énergie). Sur détection d'un événement ou sur demande, les membres actifs transmettent leurs données (constituées des grandeurs physiques mesurées) au CH auquel ils sont associés. Les CHs forment ensuite la structure de niveau hiérarchique supérieur relayant ces données jusqu'à la station de base.

#### 5.1 Les communications inter-cluster

Le gain en performance est réalisé en éliminant le surcoût causé par la formation dynamique de cluster, ainsi que la réduction du nombre d'émission et de la réception en utilisant l'agrégation de données.



**Figure 4.17** La portée optimale de communication intra et inter cluster

Afin d'implémenter les communications inter-cluster, il faut garantir que les CHs peuvent communiquer les uns avec les autres. Comme montré sur figure 4.17, seuls les CHs traitent ce message (ADV I msg, I pour inter), qui doit être envoyé avec une puissance de transmission (distance égale à 75) supérieure à celle qui a été utilisée lors de la découverte des voisins intra-cluster (distance égale 50). A la réception de ce message, chaque CH construit une liste contenant tous ses voisins CHs. Ensuite la détermination de son niveau et ses voisins qui se trouvent au niveau supérieur du côté de la SB. Par la suite, il détermine le plus proche voisin CH (plus proche que la station de base) parmi les membres du niveau supérieur. Pour cela il

## Chapitre 4: le protocole HEBM proposé

va calculer la distance entre lui et la station de base, ensuite il détermine un point qui représente le centre d'un cercle avec un diamètre qui est la distance entre lui et la station de base. Tout nœud qui a la distance minimale entre lui et le centre du cercle et qui est inférieur à la distance entre le CH concerné, le centre du cercle sera choisi comme un relais vers la station de base. Dans la figure 4.18 et figure 4.19, le meilleur relais vers la station de base sera  $E$ , et le plus court chemin est  $\{A, E, BS\}$ , car  $E$  est le plus proche au centre du cercle formé par le diamètre qui représente la distance entre le nœud et la station de base.

- (1) **Préciser** le max CH - SB { **clusterheadChoices\_**}
- (2) **Déterminer** les rayons  $R_X$  de trois niveaux à partir du **maxCH**.
- (3) **Si**  $chID \in \text{niv-3}$  **alors**
- (4) **Chercher** ( $nextHopID_$ ) **Sinon**
- (5) **Chercher** un voisin ( $nextHopID_$ )  $\in \text{niv-2}$ . **FinSi**
- (6) **Si** ( $chID \in \text{niv-2}$ ) **alors**
- (7) **Si** ( $chID$  a un voisin proche  $\in \text{niv-2}$ ) **alors**
- (8) **Chercher** ( $nextHopID_$ )  $\in \text{niv-2}$ . **Sinon**
- (9) **Chercher** un voisin ( $nextHopID_$ ) dans le niv-1.
- (10) **Si** ( $chID \in \text{niv-1}$ ) **alors**
- (11) **Si** ( $chID$  a un voisin proche dans niv1\_) **alors**
- (12) **Chercher**  $nextHopID_ \in \text{nive1}$ . **Sinon**
- (13) Le voisin est le CH  $NextHopID_ : chID$ .
- (14) **FinSi**
- (15) **FinSi**
- (16) **Fin**

Figure 4.18 : Phase de transmission Inter-cluster

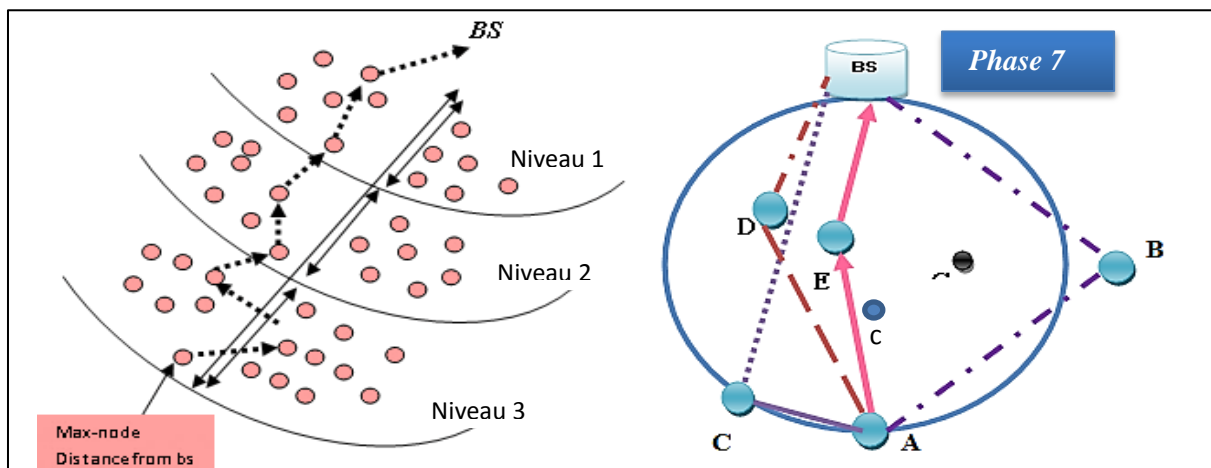


Figure 4.19 Communication multisauts avec le meilleur chemin

### 5.2 Communications intra-cluster

➤ Le routage multi sauts intra-cluster est presque le même que la communication inter-cluster, mais à deux différences très importantes, la première concerne le TDMA des membres de groupe, et la deuxième concerne les CHs qui jouent le même rôle que la BS (figure 4.19). En se basant sur la méthode TDMA, les nœuds ordinaires communiquent leurs données vers les CHs correspondants pendant un temps prédéfini. En dehors de l'intervalle de temps alloué pour la transmission, dans la figure 4.16 chaque nœud a la possibilité de se mettre en veille pour économiser ses ressources.

➤ Pour réaliser le Duty-Cycle dans HEBM, il faut synchroniser les temps de réveil et de sommeil pour chaque nœud dans le réseau. Chaque nœud de niveau 1 connaît les slots de réveil de ses prédécesseurs, et durant ces slots, il doit être à l'écoute pour pouvoir réceptionner les données du niveau 2, et le même processus se répète jusqu'au niveau 3. Pendant cette transmission, le CH reste endormi jusqu'à l'arrivée des slots d'émission de ses prédécesseurs qui sont les nœuds du niveau 1. Ainsi, le CH est à l'état actif (à l'écoute) seulement pendant les émissions des nœuds du niveau 1. Dans chaque cluster les premiers slots sont affectés au nœuds du niveau 3 vient après les slots des nœuds du niveau 2 et enfin ceux du niveau 1.

Pour réaliser cette synchronisation nous avons besoin des procédures *dutyCycleNodes* et *dutyCycleDesCH* représentées ci-dessous qui seront exécutées par les nœuds de niveau 1 et les CH respectivement (voir figure 4.20).

La procédure *DutyCycleDesCH* est comme suit :

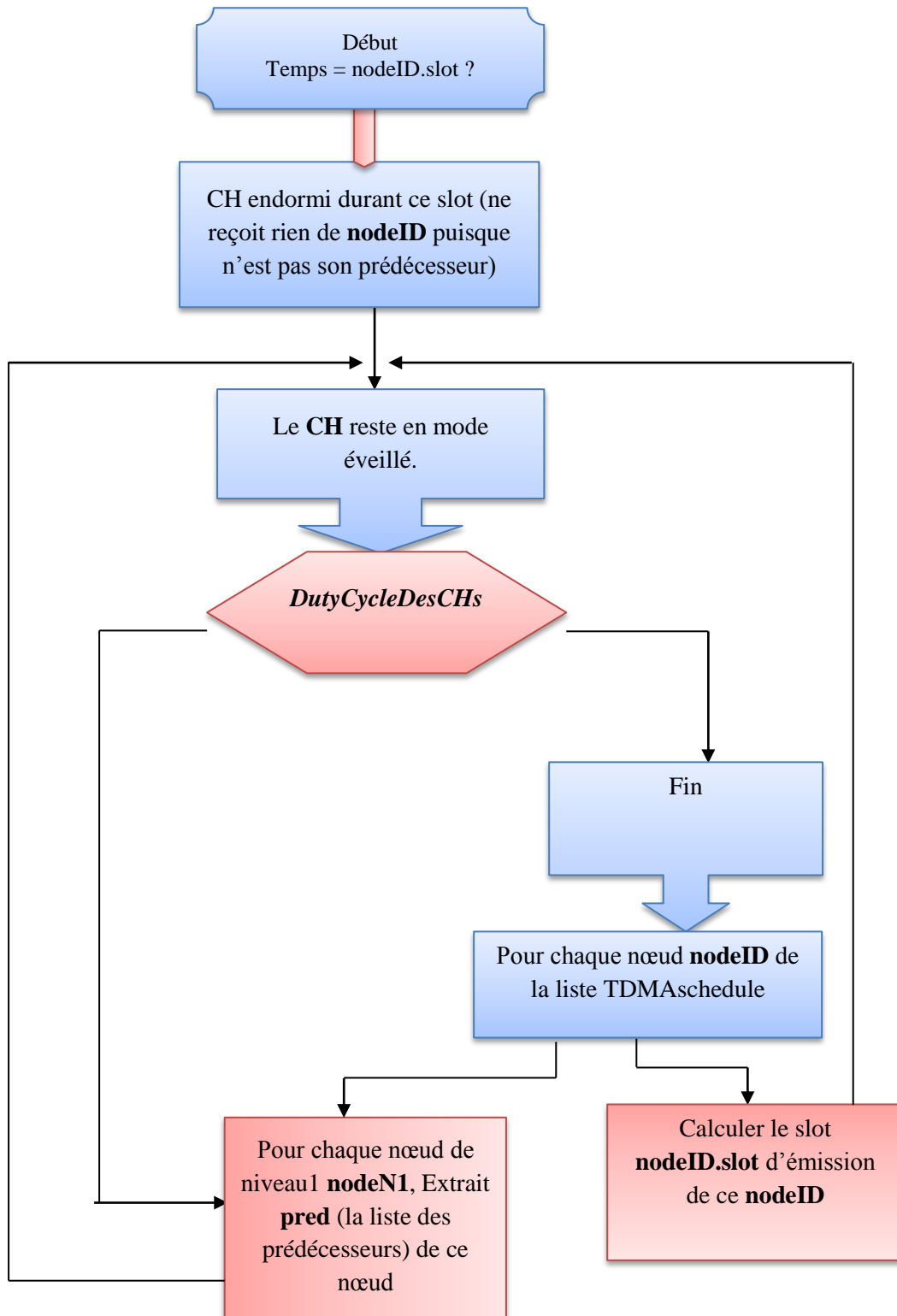


Figure 4.20 Procédure DutyCycle des CH

### 5.3 Le Duty-Cycle synchrone

Nous avons utilisé la technique Duty-cycle pour synchroniser la communication multi-sauts intra-cluster parceque le moyen le plus efficace pour conserver l'énergie est de mettre la radio

de l'émetteur en mode veille (low-power) à chaque fois que la communication n'est pas nécessaire. Idéalement, la radio doit être éteinte dès qu'il n'y a plus de données à envoyer et ou à recevoir, et devrait être prête dès qu'un nouveau paquet de données doit être envoyé ou reçu. Ainsi, les nœuds alternent entre périodes actives et sommeil en fonction de l'activité du réseau. Ce comportement est généralement dénommé Duty-cycling.

- Le Duty-cycle Est défini par la proportion de la période active sur la durée totale d'un cycle (période active + période inactive).

$$\text{Duty - cycle} = \frac{P_{\text{actif}}}{P_{\text{inactif}} + P_{\text{actif}}} \quad \text{..... Equation 4.14}$$

- Ainsi on doit le minimiser, et pour cela la période d'un capteur inactive doit être maximisée. Plus le Duty-Cycle est faible, moins le nœud consomme de l'énergie
- Chaque nœud de niveau inférieur envoie ses données à son nextHop de niveau supérieur. Ce nextHop doit être éveillé pendant le slot de transmission de ses prédécesseurs. Le CH reste endormi pendant l'émission des données des nœuds des niveaux 2 et 3.

### 5.3.1 Exemple de Synchronisation

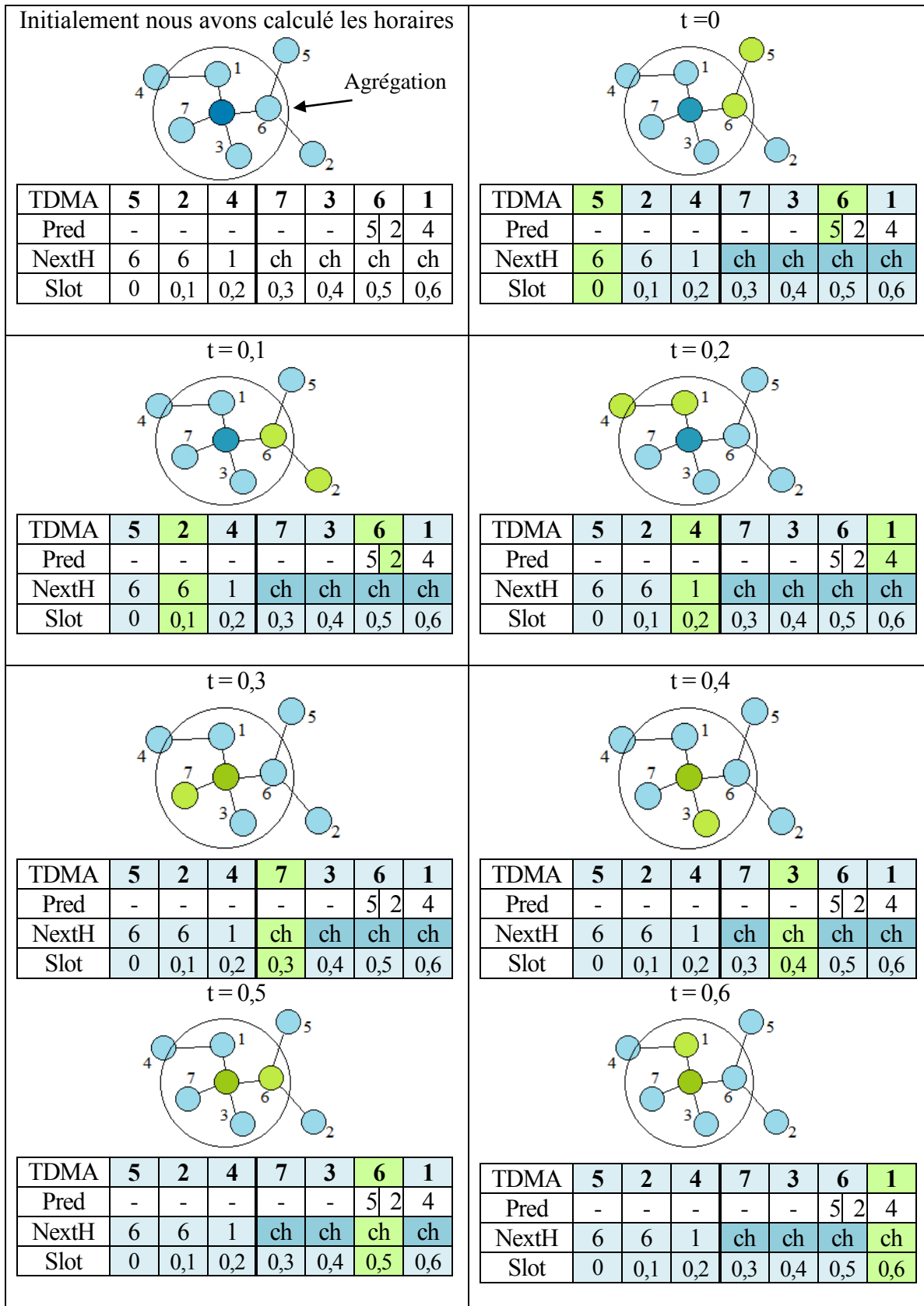
La figure 4.21 montre un exemple de synchronisation des nœuds numérotés de deux niveaux. Les nœuds 1, 3, 6 et 7 forment le niveau 1 et les nœuds 2, 4 et 5 appartiennent au niveau 2. Le CH est au milieu du cluster niveau 1. La durée d'un slot est supposée égale à 0,1 unité de temps. La couleur bleu représente l'état inactif d'un nœud et la couleur verte représente l'état actif.

TDMA : C'est la liste TDMAschedule du CH courant.

Pred : C'est la liste des prédécesseurs de chaque nœud.

Le CH est le prédécesseur des nœuds du niveau 1, le nœud 6 (niveau 1) est le prédécesseur des nœuds 5 et 2 (appartenant au niveau 2) et le nœud 1 est le prédécesseur de 4. Le nœud 6 doit être actif durant l'émission des données des nœuds 5 et 2 c'est-à-dire aux instants 0 et 0,1 ensuite il passe en mode inactif jusqu'à l'instant de son émission (à 0,5). Quant au CH, il reste en mode sommeil jusqu'à l'instant 0,3. Ce dernier reste actif seulement durant 0,4 unité de temps au lieu 0,7 unité de temps. Ce petit exemple nous indique qu'avec duty-cycle, la somme des temps actifs de tous les nœuds durant un frame (de 1,4 unité de temps) est 1,3 alors que sans duty-cycle ce temps est 1,4 (le CH reste actif durant l'émission de tous les nœuds). Pour chaque frame, le gain est :  $0,1 * \text{le coût d'écoute (pour un cluster)}$ .

## Chapitre 4: le protocole HEBM proposé



**Figure 4.21** : Exemple illustratif de la phase de synchronisation de notre proposition Duty-Cycle

### 6. Discussions sur l'efficacité du protocole HEBM

Outre les avantages apportés par HEBM et exploités pour le bon fonctionnement de notre protocole (l'efficacité énergétique, la fonction de coût pour le routage et l'acquittement implicite des paquets de données), HEBM apporte les avantages suivants :

#### ▪ **Auto organisation**

HEBM opère d'une manière complètement distribuée pour la formation des clusters. Contrairement à la plupart des approches distribuées, notre protocole ne fixe pas le nombre de cluster-heads désiré et n'utilise pas les probabilités pour leur élection. Les nœuds décident d'une manière autonome s'ils peuvent devenir cluster-heads ou pas et ceci selon les paramètres de la fonction de coût proposée.

#### ▪ **Fonction pondérée d'élection des cluster-heads**

Notre protocole prend en compte le niveau d'énergie des nœuds, leur degré de connectivité, la distance par rapport la station de base et leur emplacement par rapport à leurs voisins. En effet, la fonction proposée pour la formation des clusters évite d'avoir recours à des probabilités. La fonction est pondérée selon l'importance des métriques.

#### ▪ **Duty-Cycle synchrone**

Comme les nœuds capteurs effectuent des tâches en coopération, ils doivent coordonner leurs dates de sommeil et de réveil. Un algorithme d'ordonnancement Sommeil/Réveil accompagne donc tout plan de Duty-cycling. Il s'agit généralement d'un algorithme distribué reposant sur les périodes auxquelles des nœuds décident de passer d'un état actif et un état sommeil ou vice versa. Il permet aux nœuds voisins d'être actifs en même temp, ce qui rend possible l'échange de paquets, même si les nœuds ont un faible Duty-cycle.

#### ▪ **Routage multi-saut inter-cluster**

Dans les protocoles hiérarchiques qui se basent sur un routage à un seul saut des cluster-heads vers la station de base, les CHs s'épuisent rapidement car ils peuvent être éloignés de la station de base. Dans notre approche, un schéma de routage multi-sauts est adopté, où les chemins optimaux pour l'acheminement des paquets de données sont choisis selon une fonction de coût minimisant la dissipation d'énergie.

#### ▪ **Agrégation des données**

Les cluster-heads compressent et agrègent les données envoyés par leurs membres, puis acheminent les paquets résultant de l'agrégation vers la station de base. Ceci permet de réduire le nombre de messages de données transmis, d'optimiser la consommation d'énergie et de rendre le réseau plus scalable, par contre on peut citer l'inconvénient suivant :

### ▪ **Gestion de la panne des cluster-heads**

Comme perspective pour éviter le problème de panne de cluster-head du protocole HEBM. La désignation d'un adjoint par chaque cluster head sert à assurer le routage des données même en cas de panne des CHs. Les membres peuvent envoyer leurs données vers l'adjoint et ce dernier peut participer au routage des données.

## **7. Conclusion**

Nous avons vu que les protocoles de routage hiérarchiques apportent de nombreux avantages en termes de consommation d'énergie et de prolongement de la durée de vie du réseau. Nous avons donc opté pour la conception d'un protocole de routage adapté aux topologies hiérarchiques. Ce chapitre a pour objectif d'étudier la proposition de protocole HEBM. Ce protocole avec les 07 phases, a introduit le mécanisme d'équilibrage de la charge, grâce à la réalisation d'une bonne distribution des cluster-heads dans le réseau et améliore les performances en termes de durée de vie du réseau par la réduction du signal de transmission et l'utilisation de duty-cycle. De plus, la méthode distribuée de clustering proposée permet d'économiser l'énergie des nœuds en se basant sur des métriques telles que l'énergie et le degré de connectivité des nœuds et la condition de distance et la portée de transmission afin de s'assurer que les cluster-heads élus soient capables d'accomplir leurs tâches efficacement. Notre contribution est également adaptée aux changements de topologie et maintient le réseau à jour. Le chapitre suivant traite l'implémentation de notre protocole au niveau de l'environnement de simulation Ns2 sous Ubuntu 10.10.

## **Chapitre 5**

# **Simulation et évaluation du protocole HEBM**

### **1. Introduction**

Obtenir de meilleures performances avec les réseaux de capteurs résulte d'une auto-organisation efficace de ces réseaux. Ainsi, la proposition d'une technique d'auto-organisation et de distribution des clusters s'avère nécessaire pour que ces réseaux puissent accomplir leurs missions sans que leurs performances soient dégradées.

Une implémentation correcte de protocole de routage et d'applications pour les réseaux de capteurs nécessite de passer par une phase très importante : la phase de la simulation.

Afin d'évaluer les performances du protocole HEBM, des simulations à événements discrets sont réalisées et ce, dans le but de montrer l'apport de l'approche proposée par rapport aux approches déjà existantes à travers une comparaison de notre proposition HEBM avec les protocoles DEEAC et FEMCHRP. Nous prenons ainsi en compte plusieurs métriques pour l'évaluation des performances de notre contribution. Nous schématisons les résultats obtenus à l'aide de graphes que nous exposons et interprétons. Nous avons pu montrer à travers les résultats de simulation obtenus que les objectifs de notre protocole aient été atteints. En effet, l'efficacité énergétique de HEBM a été démontrée grâce à la politique de regroupement utilisée et à la formation de clusters distribués.

### **2. Métriques d'évaluation de notre contribution**

#### **2.1 Energie consommée**

La première étape dans la conception du système énergétique de capteurs consiste à analyser les caractéristiques de consommation d'énergie d'un nœud de capteur sans fil. Cette analyse systématique de l'énergie d'un capteur est extrêmement importante pour identifier les problèmes dans le système énergétique pour permettre une optimisation efficace.

#### **2.2 Durée de vie**

Chaque capteur doit gérer efficacement son énergie afin de garder le réseau en état opérationnel. Le facteur le plus contrôlable dans la durée de vie des systèmes de capteurs sans

fil est l'énergie du module radio. En fait, la consommation de cette énergie peut être réduite en diminuant la puissance du signal de transmission etc.

- Energie moyenne résiduelle des capteurs en fonction du temps.
- Energie consommée au fil du temps.
- La durée de vie de réseau pour 100 nœuds.
- La durée de vie de réseau pour 200 nœuds.
- Pourcentage des noeuds vivants pour chaque protocole.
- Le 1er et le dernier nœud ayant épuisé son énergie au fil du temps de simulation.
- La relation entre la position de la BS et la durée de vie.
- Le nombre de clusters par rapport au nombre de nœuds.
- Le nombre de clusters par rapport à portée de transmission.
- La distribution des cluster-heads du protocole HEBM.
- La moyenne de latence des paquets par rapport au nombre de nœuds.
- Le nombre des paquets transmis avec différentes tailles au fil de temps.

### **3. Environnement de simulation**

#### **3.1 Objectifs de simulation**

La simulation permet de tester à moindre coût les nouveaux protocoles et d'anticiper les problèmes qui pourraient se poser dans le futur afin d'implémenter la technologie la mieux adaptée aux besoins. NS2 est un simulateur à événements discrets disponible gratuitement sur le site <http://www.isi.edu/nsnam/>. Il permet à l'utilisateur de définir un réseau et de simuler des communications entre les noeuds de ce réseau. NS2 utilise le langage OTCL (Object Tools Command Language), dérivé objet de TCL. À travers ce langage, l'utilisateur décrit les conditions de la simulation : topologie du réseau, caractéristiques des liens physiques, protocoles utilisés, communications etc. La simulation doit d'abord être saisie sous forme de fichier texte que NS2 utilise pour produire un fichier trace contenant les résultats. NS2 est fourni avec différents utilitaires dont des générateurs aléatoires et un programme de visualisation : NAM.

#### **3.2 Choix de simulateurs**

Nous avons choisi le simulateur le plus répandu dans le domaine des réseaux qui est NS2. NS2 est destiné à simuler les protocoles de communication des réseaux filaires, sans fil et satellitaires tel que TCP, IP, le routage, le multicast et les protocoles de la couche application. L'utilisation du simulateur est gratuite et son développement suit une approche orientée objet.

### 3.3 Présentation du simulateur NS2

NS2 est un simulateur à événements discrets orienté objet. Il est écrit en C++ avec une interface textuelle (ou shell) qui utilise le langage OTcl. L'OTcl est une extension objet du langage de commande Tcl. Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Pour reprendre la terminologie objet, il sert à définir les classes. Quant au langage OTcl, il fournit un moyen flexible et puissant de contrôle de la simulation comme le déclenchement des événements, la configuration du réseau, la collecte de statistiques, etc. (figure 5.1) L'application NS2 se compose de deux éléments fonctionnels: un interpréteur et un moteur de simulation. Au moyen de l'interpréteur, l'utilisateur est capable de créer le modèle de simulation, ce qui revient à assembler les différents composants nécessaires à l'étude.

### 3.4 Comment peut-on simuler le protocole HEBM dans NS2 ?

La figure qui suit montre les principales étapes à suivre pour une simulation sous NS2.

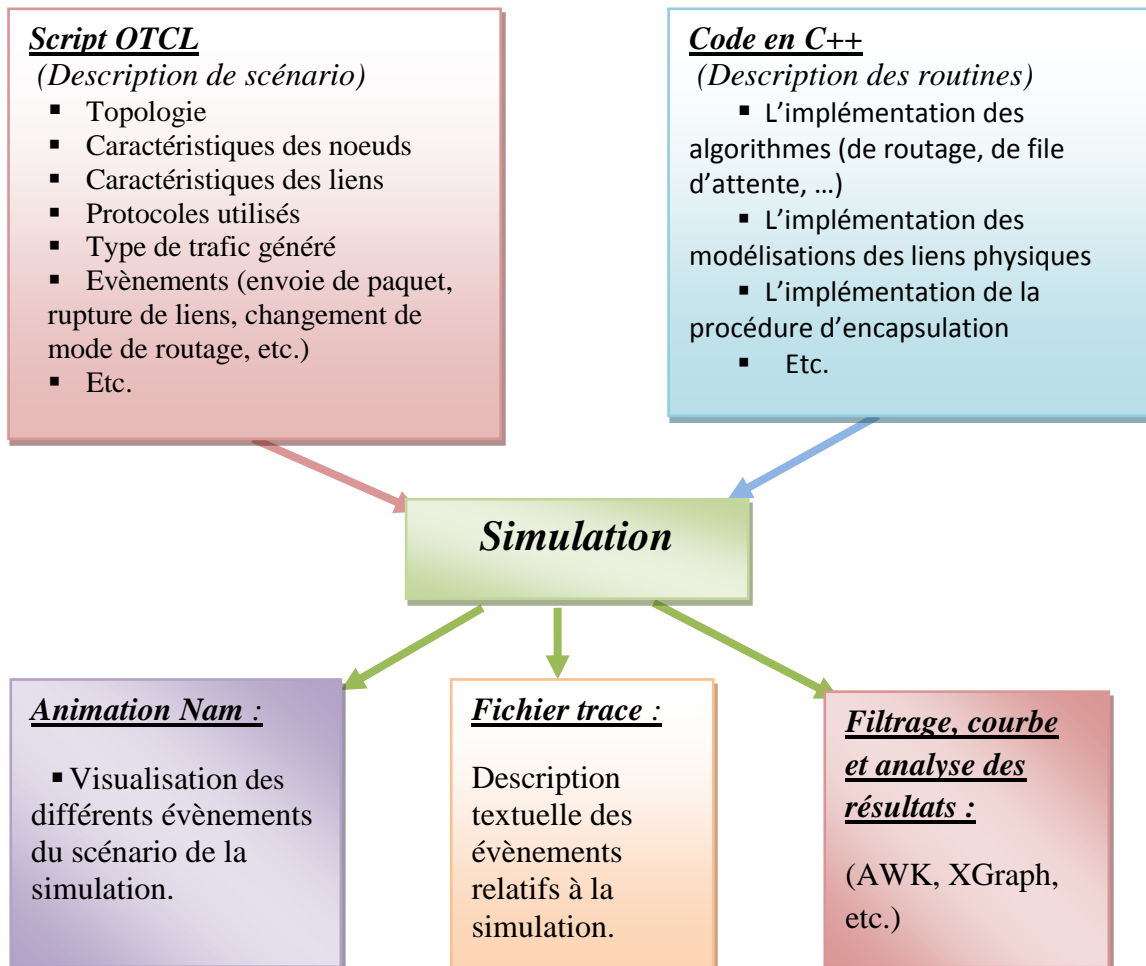


Figure 5.1 Les étapes de simulation d'un protocole sous NS2

### 3.5 Avantages, limites et difficultés de NS2

- Comme tout outil de simulation, NS2 permet l'étude de l'existant, la conception, la validation et l'évaluation de performances et ceci dans le domaine des protocoles et mécanismes réseau.
- C'est un simulateur extrêmement flexible. NS2 permet l'étude des cas difficiles à reproduire dans la réalité. Il offre la souplesse de pouvoir varier aisément une multitude de paramètres du réseau (ce qui n'est pas le cas si on simule sur un réseau réel).
- D'un autre côté, NS2 souffre des inconvénients des simulateurs tels que : les performances du simulateur sont assez limitées et ne semblent pas adaptées pour des simulations de réseaux importants, la simplification et l'abstraction du monde réel, la difficulté de tester certains aspects, et la puissance de calcul requise qui croit exponentiellement avec la complexification du système simulé.

### 4. Evaluation des Performances

Dans cette section, nous évaluons les performances de l'HEBM proposé par plusieurs expériences de simulation. Dans un premier temps, la configuration de la simulation est expliquée et les résultats sont présentés. Une comparaison entre les résultats de la simulation des algorithmes HEBM, DEEAC et FEMCHRP est réalisée. Nous utilisons deux scénarios pour les simulations. Dans le premier scénario, 100 noeuds sont uniformément et aléatoirement dispersés dans un champ de taille 200mx200m. Pour étudier l'effet d'échelle sur la performance de HEBM, un deuxième scénario a été introduit utilisant 200 noeuds dispersés uniformément. On suppose que la station de base est située au centre du champ. Les autres paramètres de la simulation sont résumés dans le tableau 5.1.

Paramètres	Valeur
La surface	200m X200m
Taille du paquet de données	4000 bits
Taille du paquet de contrôle	512 bits
Nombre de noeuds capteurs	100/200/.../800
Énergie initiale	2 Joule
Emplacement de la station de base	(50,50)
$E_{elec}$	50nj/bit

Tableau 5.1 Les paramètres de simulation

#### 4.1 Energie résiduelle

Vu les limitations en énergie des capteurs, il est primordial de réduire l'énergie consommée à tous les niveaux et ce afin de permettre une plus grande longévité au réseau. On constate sur la figure 5.2, que l'énergie moyenne résiduelle de trois protocoles étudiés diminue avec l'augmentation de temps de simulation. Ceci est dû au nombre de messages qui transitent dans le réseau. Il est à noter que ce résultat reste très raisonnable avec HEBM contrairement aux autres protocoles FEMCHRP et DEEAC. En effet, le gain en énergie de HEBM est très avantageux relativement aux deux protocoles FEMCHRP et DEEAC. Les simulations effectuées montrent que l'énergie moyenne résiduelle au niveau du protocole HEBM est supérieure à l'énergie moyenne restante au niveau de FEMCHRP et DEEAC. Ceci est dû notamment à la différence du mode de fonctionnement des trois protocoles. HEBM (pour  $t=400$  s,  $E_r$  du réseau=300 j), DEEAC (pour  $t=400$ s ,  $E_r$  du réseau=270 j), FEMCHRP (pour  $t=400$ s,  $E_r$  du réseau = 150 j) même avec le temps de 1600s l'énergie consommée avec le protocole HEBM est 3 fois moins importante que celle consommée avec les deux autres protocoles. Ceci prouve l'apport de la technique d'équilibrage de charge adoptée et le duty cycle utilisé. Donc le protocole HEBM équilibre la charge d'énergie grâce à la bonne distribution des cluster-heads, ce qui assure l'amélioration de la quantité de l'énergie résiduelle et par conséquent une longévité de réseau.

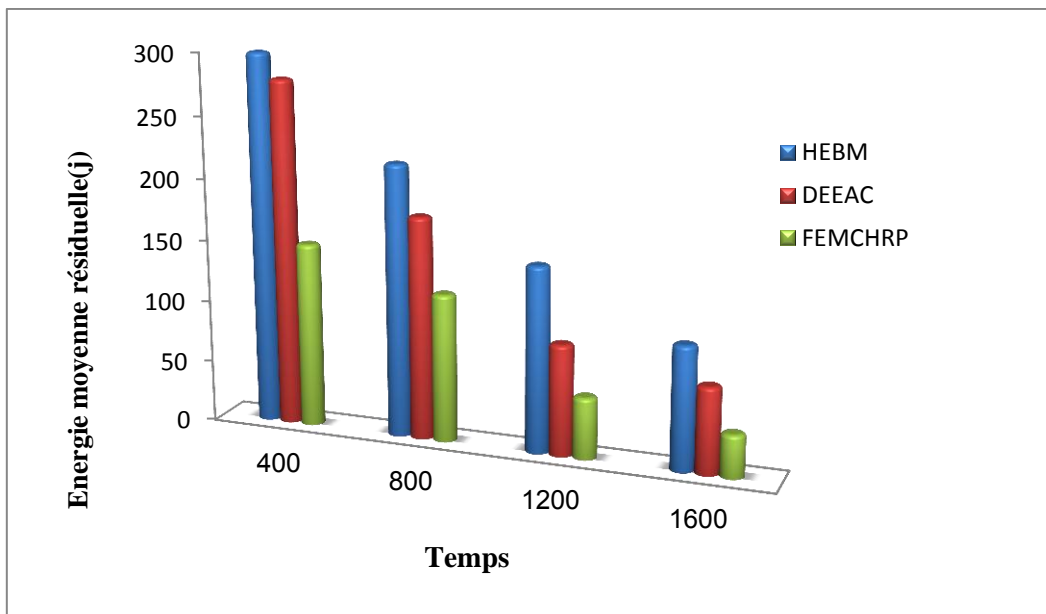


Figure 5.2 Energie moyenne résiduelle des capteurs en fonction du temps.

#### 4.2 L'énergie consommée

Le graphe de la Figure 5.3 représente l'énergie consommée en fonction du temps de simulation ; nous pouvons voir la faible consommation énergétique du protocole HEBM par rapport aux protocoles FEMCHRP et DEEAC: la raison c'est que dans le protocole HEBM la sélection de ClusterHead est déterminée selon des métriques hybrides: énergie résiduelle, la distance entre le nœud et BS, le degré de nœud (taux de connectivité), le nombre des voisins, la distance entre les nœuds. Le gain en énergie de HEBM est très avantageux relativement aux deux autres protocoles DEEAC et FEMCHRP. Pour ( $t=500$  s,  $HEBM-E_{cons}=400$  j,  $DEEAC-E_{cons}=720$  j,  $FEMCHRP-E_{cons}=1100$ ). La raison principale de ce résultat est le nombre approprié de clusterhead et la répartition des clusters dans le réseau. HEBM utilise une distribution des ClusterHeads sur tout le réseau total avec une sélection très spécifique des cluster-heads selon des formules mathématiques proposées, avec une communication multi-sauts Inter-cluster et intra-cluster.

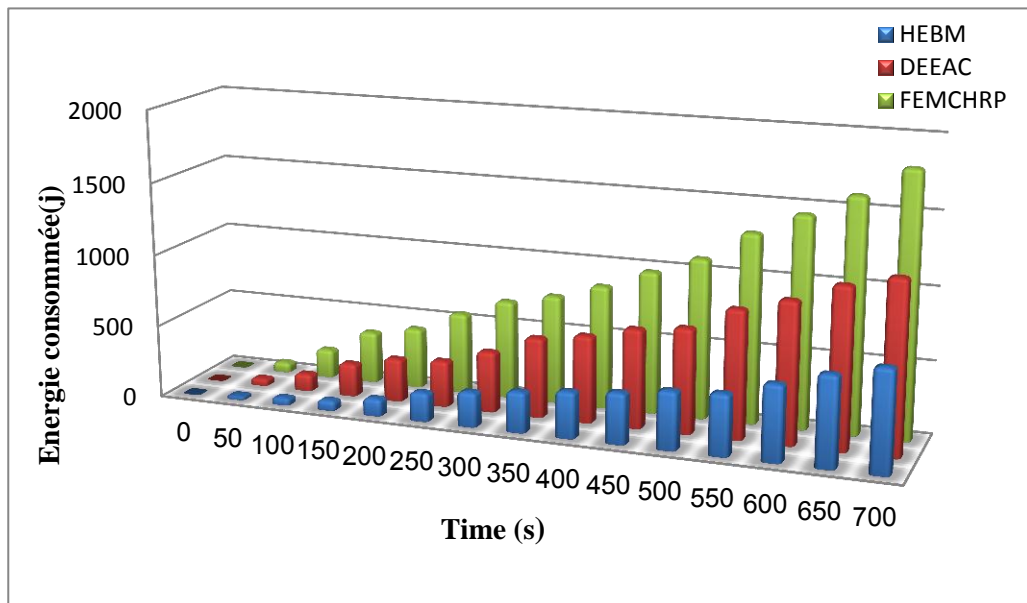


Figure 5.3 L'énergie consommée au fil de temps

#### 4.3 Durée de vie de HEBM

5. La durée de vie du réseau est définie comme étant le temps qui s'écoule ou le nombre de cycles de captage, depuis le déploiement des nœuds, c'est-à-dire le démarrage de l'activité du réseau jusqu'au moment où le réseau devient non-fonctionnel [10]. Le réseau est considéré comme non-fonctionnel selon les spécificités de l'application pour laquelle il a

été déployé. Pour certaines applications critiques, le réseau est considéré comme non-fonctionnel dès la mort du premier nœud dans le réseau (si absence de redondance). Un nœud est considéré comme mort dès l'épuisement de son énergie ( $E_{disp}=0$ ). Dans ce cas, le nœud ne peut plus émettre ni réceptionner de messages. Pour d'autres applications, le réseau est considéré comme non-fonctionnel à partir de la mort d'un certain pourcentage de nœuds. Le réseau peut également être considéré comme non-fonctionnel dès la perte de la connexité totale (perte de couverture) c'est-à-dire à partir du moment où la BS n'est plus atteignable depuis n'importe quel nœud du réseau.

6. Nous évaluons la durée de vie du réseau dans chacun des trois scénarios de routage (HEBM, FEMCHRP, DEEAC) en mesurant d'abord le temps qui s'écoule et le nombre de cycles de captage depuis le déploiement des nœuds jusqu'à ce que le premier nœud disparaisse.
7. Ensuite, sans refaire de re-clustering, nous évaluons la durée de vie maximale du réseau jusqu'à ce qu'à la perte de connexité totale. L'objectif est d'évaluer le nombre de nœuds  $i$  dont l'épuisement total de la batterie conduit à une perte de connexité totale.
8. Ainsi, nous avons mesuré l'énergie résiduelle des nœuds capteurs toutes les 20 secondes pendant toute la durée de la simulation, afin de calculer le nombre total des nœuds vivants. De plus, nous avons calculé le pourcentage de nœuds morts, au fur et à mesure de l'avancement de la simulation. Les figures (5.4 et 5.5) présentent les résultats que nous avons obtenus. Les résultats obtenus représentent le nombre total de nœuds (100 et 200 dispositifs) qui restent en vie pendant le temps de simulation. Les résultats montrent que HEBM fonctionne mieux que DEEAC et le FEMCHRP. En DEEAC (200 nœuds), le premier nœud perd l'énergie après (210 rounds). Pour le protocole FEMCHRP, après 800 rounds, presque tous les nœuds ont perdus leur énergie. Mais pour le protocole HEBM, le premier nœud épuise son énergie après 310 rounds. Ainsi, HEBM fonctionne mieux pour une densité de capteurs de 100 et 200 nœuds. Le protocole HEBM améliore considérablement la consommation moyenne d'énergie dans le réseau par rapport au protocole FEMCHRP. Par contre, en HEBM, FEMCHRP on envoie les données agrégées directement à la station de base (BS). Cette amélioration peut être justifiée par le mécanisme appliqué par HEBM qui essaye d'acheminer les paquets via le chemin optimal en termes de consommation d'énergie. Lorsque la durée du round est longue, un nœud tombe en panne plus rapidement vu le nombre important de messages de contrôle échangés qui surchargent considérablement le réseau. En effet, les phases d'initialisation

et de clustering qui induisent un nombre important de messages de contrôle vont se faire à chaque nouveau round impliquant une consommation d'énergie supplémentaire. Cependant, en choisissant une durée de round faible, l'élection des cluster-heads est renouvelée plus souvent évitant ainsi d'épuiser la réserve d'énergie des anciens CHs. Le graphe de la figure 5.6 représente le pourcentage des nœuds vivants dans le réseau. On peut conclure que HEBM augmente la valeur d'énergie résiduelle et il prolonge la durée de vie du réseau, car nous utilisons une proposition d'élection d'un cluster head adaptatif. Les cluster-heads sont élus en fonction de l'énergie résiduelle de chaque nœud et l'énergie moyenne de chaque cluster. HEBM améliore la durée de vie du réseau de 55% à 75% par rapport au protocole FEMCHRP et de 40% à 50% par rapport au protocole DEEAC.

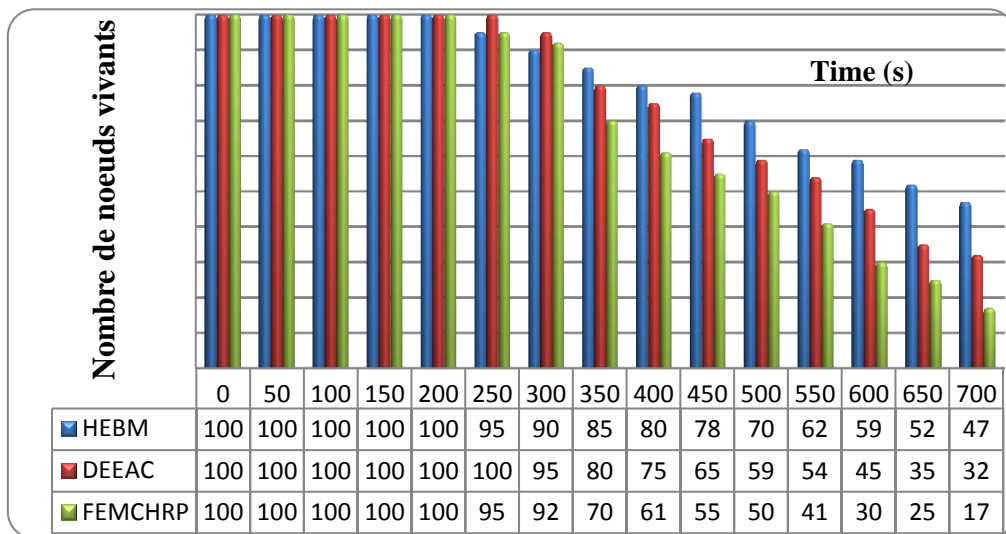


Figure 5.4 Durée de vie du réseau pour 100 nœuds

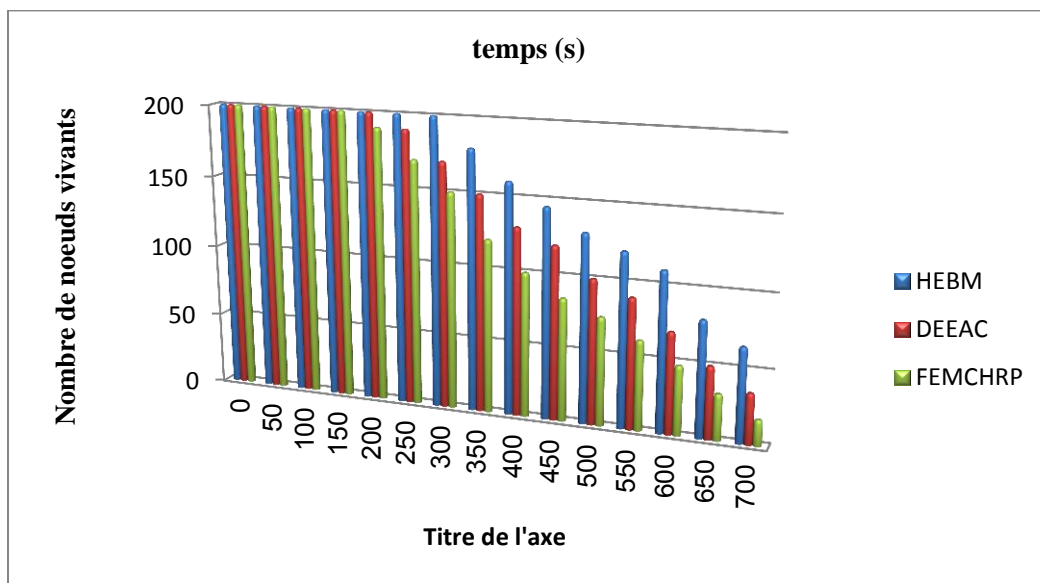


Figure 5.5 Durée de vie du réseau pour 200 nœuds

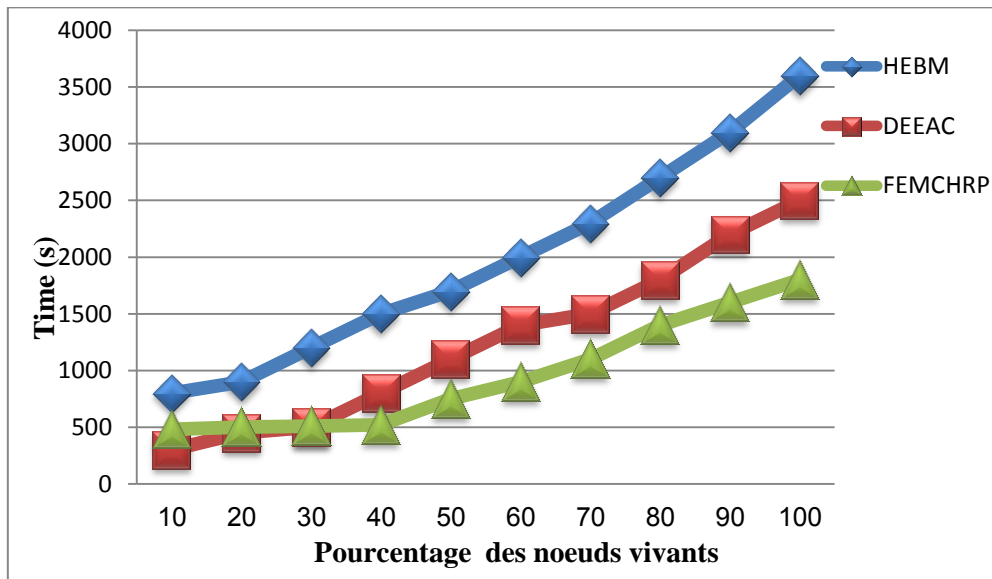


Figure 5.6 Pourcentage de nœuds vivants pour chaque protocole.

### 8.1 Instant de déconnexion du 1<sup>er</sup> et du dernier nœud

On peut observer sur la Figure 5.7, que la durée de vie du réseau obtenue par HEBM est plus grande que DEEAC. Pour 500 nœuds, le dernier nœud meurt dans DEEAC après 11350 rounds et dans HEBM il meurt après 6140 tours. Ainsi, la période de stabilité de HEBM est plus grande que dans DEEAC. Cette amélioration est due au fait que le protocole HEBM élit les CHs avec différents paramètres. En outre, dans cette approche, l'équilibrage de la charge dans le réseau est exécuté correctement et une communication inter-cluster et intra-cluster en utilisant le duty cycle pour les CHs. Ce qui donne un temps plus long entre le début des opérations jusqu'à ce que le moment où le premier nœud meurt. Nous avons utilisé la technique Duty-cycle pour synchroniser la communication multi-sauts intra-cluster parce que le moyen le plus efficace pour conserver l'énergie est de mettre la radio de l'émetteur en mode veille (low-power) à chaque fois que la communication n'est pas nécessaire. Cela est dû qu'au moment du routage des données, le protocole HEBM équilibre la charge d'énergie grâce à la bonne distribution des CHs, ce qui améliore la durée de vie du réseau et assure une consommation minimale d'énergie par rapport aux protocoles FEMCHRP et DEEAC. De même, la présence de la topologie hiérarchique globale réduit les longues transmissions depuis les nœuds leaders jusqu'à la SB.

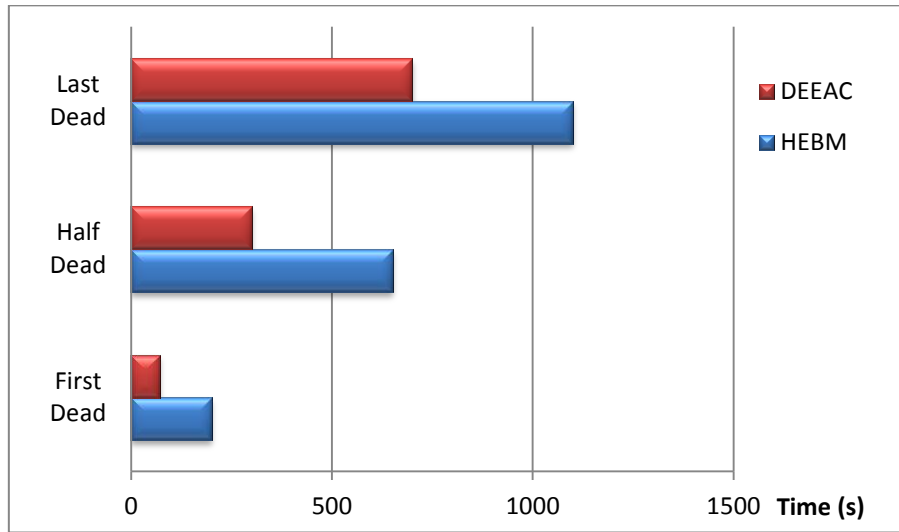


Figure 5.7 Mort du 1<sup>er</sup> et du dernier nœud

## 8.2 Relation entre la position de la BS et la durée de vie de réseau

La figure 5.8 présente la durée de vie du réseau de HEBM, DEEAC et FEMCHRP avec différentes positions de la SB. A cause de l'augmentation de la distance entre la SB et le réseau, la consommation d'énergie des nœuds qui communiquent directement avec la BS augmente remarquablement. La performance du protocole HEBM est l'utilisation de la communication multi-sauts pour communiquer avec la SB. Et puisque la distance entre chaque nœud et la SB est différente, la consommation d'énergie pour chaque nœud est différente aussi. Les capteurs proches de la SB vont plus rapidement manquer d'énergie car ils seront plus sollicités pour relayer les messages des autres. La solution consiste à hiérarchiser les échanges en divisant la zone d'observation en clusters. On améliore ainsi la qualité du lien radio en utilisant la communication multi-sauts intra et inter cluster avec une technique duty-cycle. Ainsi, avec cette communication hiérarchique et multi-sauts le réseau reste fonctionnel malgré la station de base soit éloignée. L'éloignement de la SB et la durée de vie de réseau sont inversement proportionnels. Comme le montre la figure 5.8, si la distance qui sépare la SB et le réseau est POS-BS=500 alors la durée de vie du réseau dans FEMCHRP est 700s, dans DEEAC elle est 850s et dans HEBM elle atteint 1150s. La durée de vie du protocole HEBM reste plus importante par rapport au DEEAC et FEMCHRP. On conclut que la durée de vie du réseau HEBM et DEEAC est plus longue que celle de FEMCHRP.

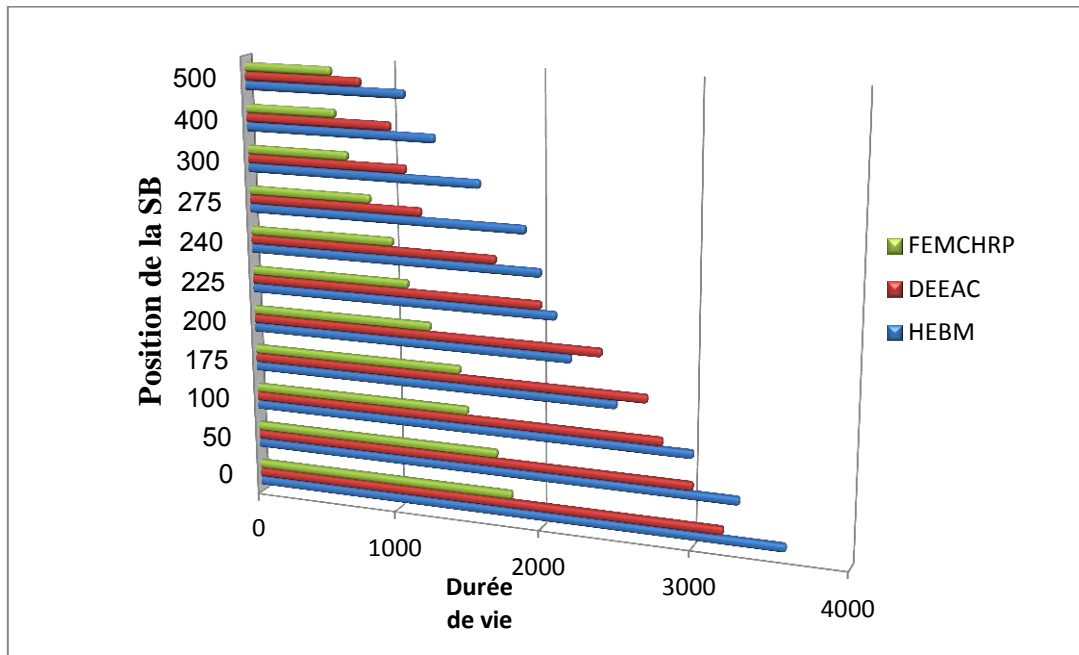


Figure 5.8 Relation entre la position de la BS et la durée de vie

### 8.3 Nombre des clusters par rapport au nombre de nœuds

La figure 5.9 montre le nombre moyen de clusters obtenus dans le réseau par rapport au nombre de nœuds. Le protocole HEBM fournit environ 32,81% moins de clusters que le FEMCHRP et 45,73% moins que DEEAC. Au niveau du protocole HEBM, nous avons utilisé l'équilibrage de la charge, ce résultat montre que notre algorithme de clustering crée un nombre de clusters légèrement inférieur par rapport aux protocoles FEMCHRP et DEEAC. Nous expliquons ce phénomène par la résolution supérieure de la densité par rapport au degré, ce qui permet de créer des clusters plus optimaux par l'utilisation des 2 techniques (Clusterhead temporaire et final) et l'évitement de la sélection des chefs proches à l'aide de la condition de distance. Pour élire le cluster-head final, si le nombre de nœuds est égal à 100, le nombre moyen de clusters pour les trois protocoles est 8, 10, 12 dans les protocoles HEBM, DEEAC et FEMCHRP respectivement.

Cette solution a pour objectif d'élire un nombre optimal et distribué des chefs ce qui permet finalement d'obtenir des clusters équilibrés et distribués sur la surface de réseau.

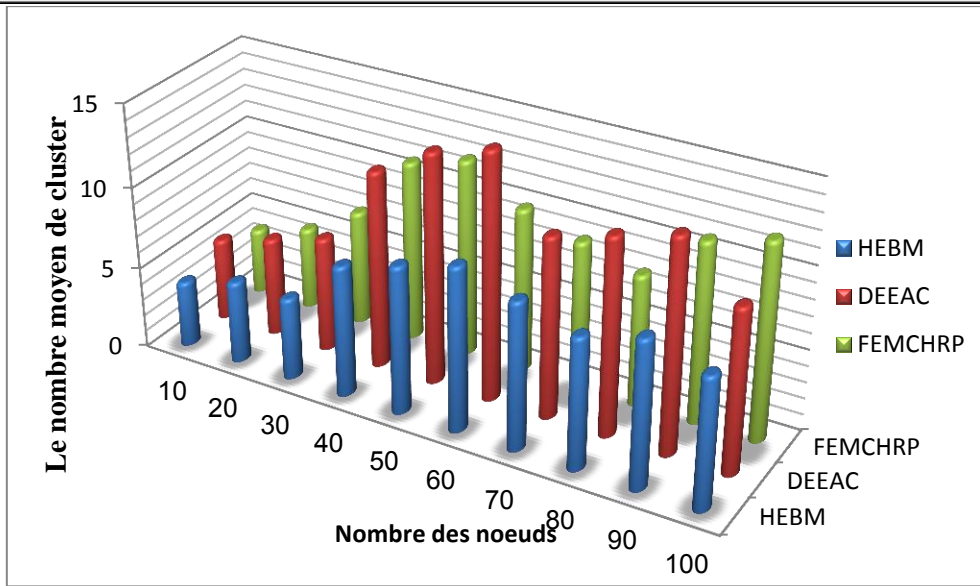


Figure 5.9 Nombre de clusters par rapport au nombre de nœuds

### 8.4 Nombre de clusters par rapport à la portée de transmission

La Figure 5.10 illustre la formation des clusters du protocole HEBM par rapport à la portée de transmission. Ce graphe montre que lorsque la portée de transmission augmente, le nombre moyen de Clusters diminue. Ainsi, pour un réseau composé de  $n=800$  nœuds et une portée de transmission  $Tr=150$ , le nombre de clusters est  $N.cluster = 4$ . Tandis que pour  $n=600$  et  $Tr=110$ ,  $N.cluster = 8$  ; pour  $n=200$ ,  $Tr=50$ ,  $N.cluster = 12$ . La raison est qu'un cluster avec une large portée de transmission couvrira une plus grande surface. D'où l'adaptation du protocole HEBM réalisée en fonction du nombre de nœuds et la portée de transmission des capteurs.

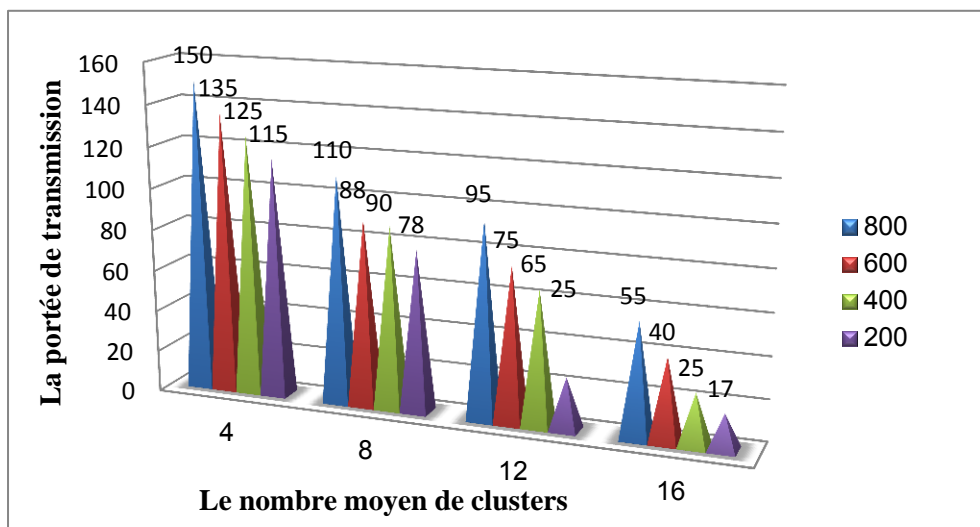


Figure 5.10 Nombre de clusters par rapport à la portée de transmission

### 8.5 Distribution des cluster-heads du protocole HEBM

La figure 5.11 montre la répartition géographique des cluster-Heads dans protocole HEBM, sur un réseau (200 m X 200 m). L'algorithme de regroupement HEBM est complètement distribué parce qu'une méthode de contrôle centralisée n'est pas pratique pour le passage à l'échelle d'un réseau de capteurs (Pour  $n=100$ ,  $n=200$ ,  $n=300$ ,  $n=400$ ). Les Cluster-Heads dans HEBM sont bien répartis sur la zone de surveillance afin d'avoir toute partie du réseau couverte par un CH et ainsi minimiser la consommation d'énergie des capteurs lors de la communication avec son CH. L'équilibrage de la charge est réalisé par la spécification d'un seuil prédéfini sur le nombre de Cluster-heads, ce nombre de chefs peut couvrir idéalement la surface de déploiement des noeuds. Cela garantit qu'aucun CH ne soit surchargé à tout moment. Dans notre algorithme, l'équilibrage de la charge signifie que chaque cluster-head peut gérer approximativement le même nombre de noeuds en même temps.

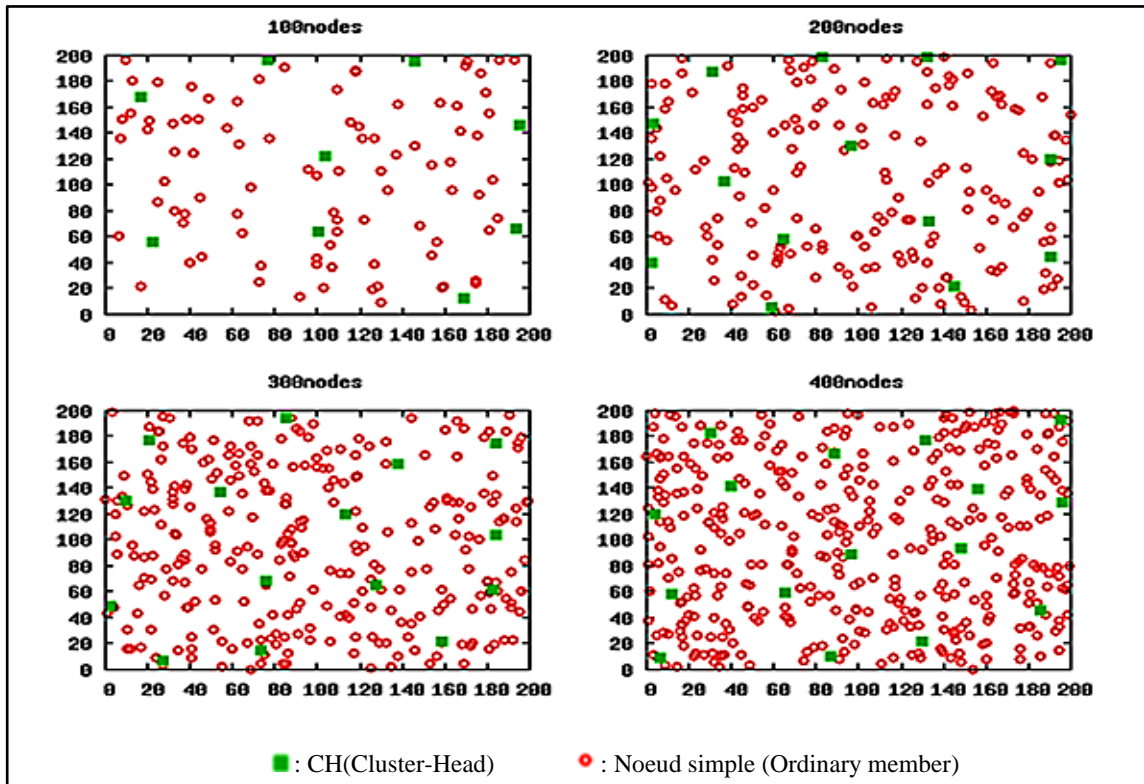


Figure 5.11 Distribution des cluster-heads dans le protocole HEBM

### 8.6 Latence moyenne des paquets par rapport au nombre des noeuds

La figure 5.12 illustre la latence des paquets sur le nombre de noeuds. La Latence est le délai entre l'émission par la source et la dernière réception du message de diffusion.

C'est un critère déterminant dans les réseaux de capteurs car il permet à l'utilisateur d'intervenir rapidement à l'occurrence d'événement. C'est le temps nécessaire pour transmettre un paquet de données à la BS. Certains dispositifs comme les capteurs n'ont pas besoin d'une largeur de bande très élevée, mais plutôt d'un temps de latence faible ainsi qu'une consommation d'énergie très basse, pour la durée de vie de batterie et un grand nombre de dispositifs d'où la nécessité de concevoir d'autres normes sans fils capables de répondre à ces exigences.

Les résultats de simulation montrent que le protocole HEBM offre des meilleurs résultats en termes de réduction de la latence contrairement aux autres protocoles comme le DEEAC et le FEMCHRP. La latence maximale de protocole HEBM par rapport au nombre des nœuds est 20% contrairement au protocole DEEAC qui est 40%, et 55 % pour le protocole FEMCHRP. Cela est justifié par l'utilisation d'antennes d'activation radio, qui permettent aux nœuds capteurs d'être prêts à tout événement de transmission ou de réception de données. Comme les nœuds capteurs effectuent des tâches en coopération, ils doivent coordonner leurs temps de sommeil et de réveil. Un algorithme d'ordonnancement Sommeil/Réveil accompagne donc le Duty-cycling. Il s'agit généralement d'un algorithme distribué reposant sur les périodes auxquelles des nœuds décident de passer d'un état à un autre (l'état actif et l'état sommeil). Il permet aux nœuds voisins (émetteur et récepteur) d'être actifs en même temps, ce qui rend possible l'échange de paquets avec un faible Duty-cycle (ils dorment la plupart du temps).

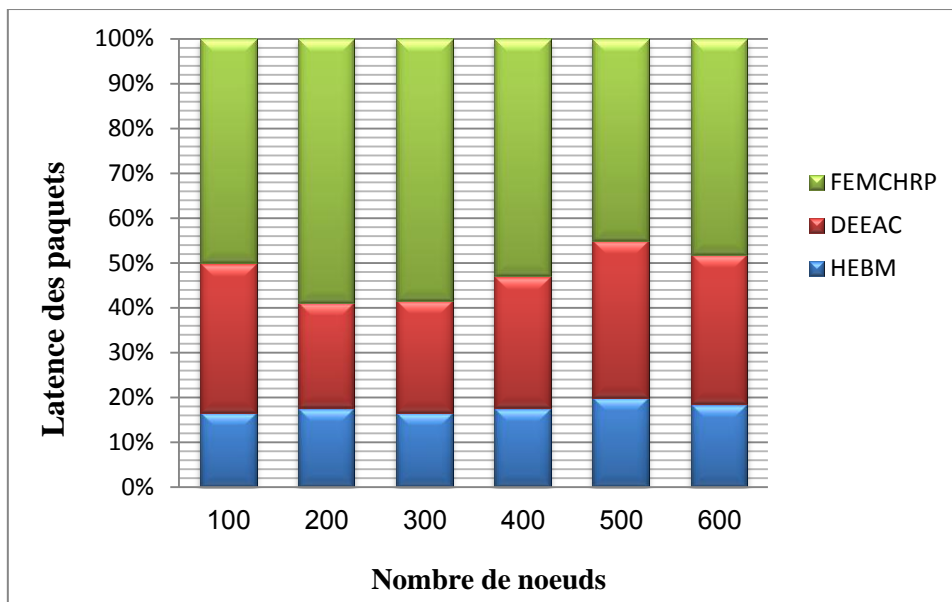


Figure 5.12 Latence moyenne de paquets par rapport au nombre de nœuds

### 8.7 Nombre de paquets transmis avec différentes tailles

Dans la Figure 5.13, le nombre des paquets transmis avec différentes tailles au fil du temps ce qui représente : nombre de paquets, taille du paquet, Temps = fonction du nombre de paquets,

cette figure illustre dans le protocole HEBM le nombre de paquets envoyé avec les tailles suivantes (4 paquets/s, 16 paquets /s, 32 paquets /s).

(Pour la taille de 4 paquets, pendant 512s, le nombre de paquets transmis = 6 paquets)

(Pour la taille de 16 paquets, pendant 512s, le nombre de paquets transmis = 4 paquets)

(Pour la taille de 32 paquets, pendant 512s, le nombre de paquets transmis = 2 paquets)

Donc on peut conclure d'après ces résultats que 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 élevée ni trop faible. En effet, si elle est petite, le nombre de paquets de contrôle (acquiescement) généré augmente l'overhead. Dans le cas contraire, une grande puissance de transmission est nécessaire.

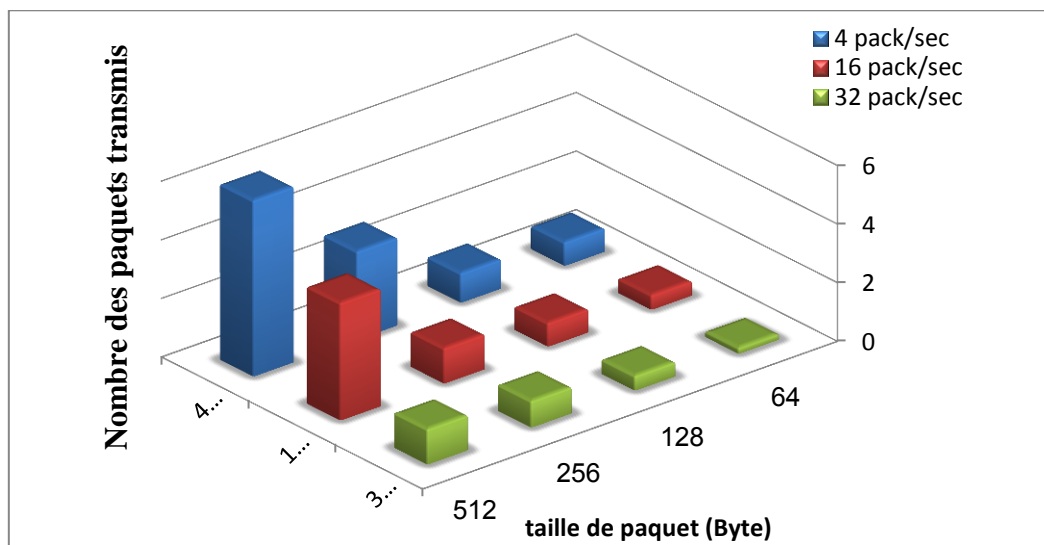


Figure 5.13 Nombre de paquets transmis avec différentes tailles

### 8.8 Nombre de paquets délivrés à la BS

La figure 5.14 représente le nombre total de données reçues à la Station de Base en fonction du temps (les paquets envoyés, reçus et perdus). Le protocole HEBM proposé avec l'auto-adaptation des clusters et chefs montre que le nombre de données reçues par la station de base est plus élevé que le protocole DEEAC. Cela est dû à la durée de vie du réseau de capteurs qui est plus longue avec HEBM qu'avec DEEAC. La performance de HEBM est plus grande lorsque le nombre CH est variable. Par exemple à l'instant  $t=1500s$ , le nombre de paquets est égal à  $6 \times 10^4$  dans le protocole HEBM alors que dans DEEAC le nombre de paquets est  $3 \times 10^4$ . Donc le protocole HEBM a une amélioration de 50 %. Cet avantage offert par HEBM est dû au mécanisme utilisé qui procure une grande possibilité d'envoi des données vers la station de base. Nous avons utilisé l'équation suivante (5.1) pour calculer la probabilité des paquets envoyés.

$$Pqt - recv = \frac{nb\_succ\_packet}{tot\_nb\_send\_packet} \dots\dots\dots \text{Équation 5.1}$$

nb\_succ\_packet : Nombre de paquets reçus avec succès par la station de base.

tot\_nb\_send\_packet : Nombre total de paquets envoyés par chacun des capteurs.

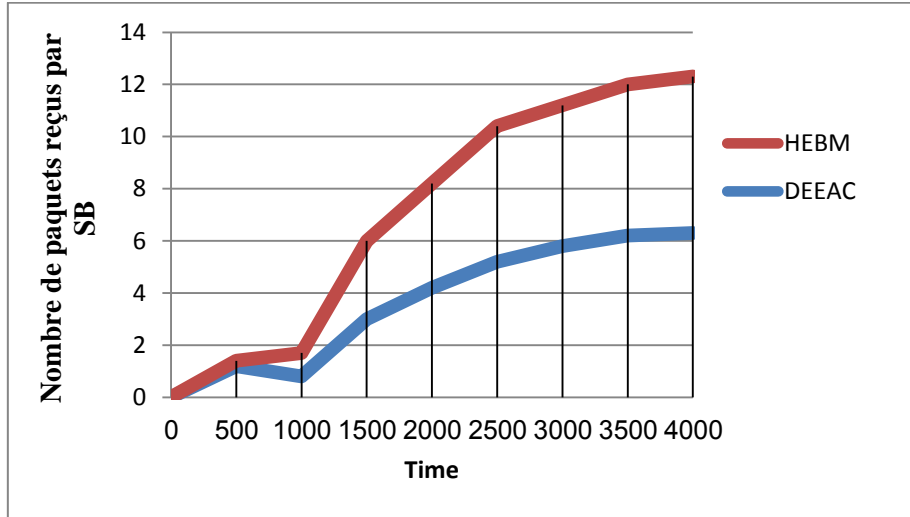


Figure 5.14 Nombre de paquets délivrés à la BS

Protocol / Characteristics	FEMCHPR (Fuzzy Based Energy Efficient Multiple Cluster Head Selection Routing Protocol)	DEEAC (Distributive Energy Efficient Adaptive Clustering protocol)	HEBM (Proposed protocol) ( Hierarchical Energy-Balancing Multipath routing protocol for wireless sensor networks )
Efficacité énergétique	Bonne	Bonne	Excellente
Stabilité des clusters	Moyenne	Elevée	Excellente
L'équilibrage de charge	Mauvais	Bon	Excellent
Délai de livraison	Moyen	Moyen	Petit
Évolutivité	Faible	Faible	Bonne
Fiabilité	Bonne	Bonne	Elevée
Complexité des algorithmes	Bonne	Moyenne	Bonne
Structure de routage	Plat/ hiérarchique	Plat/ hiérarchique	Plat/ hiérarchique
Sécurité	Non	Non	Non (les futures travaux)

Tableau 5.2 Etude comparative des trois protocoles (FEMCHRP, DEEAC and HEBM)

Comme le montre le tableau 5.2, notre proposition améliore les facteurs les plus importants comme l'efficacité énergétique, L'équilibrage de charge et le passage à l'échelle.

### **9. Conclusion**

Dans ce dernier chapitre, nous avons présenté l'environnement de simulation NS2, son principe de fonctionnement ainsi que la plateforme de simulation utilisée. Après avoir implémenté notre protocole HEBM nous avons implémenté aussi les deux autres protocoles FEMCHRP et DEEAC afin de pouvoir les comparer et montrer les performances de notre protocole. L'étude comparative entre les trois protocoles est faite sur différentes métriques à savoir 1) Energie moyenne résiduelle des capteurs, 2) l'énergie consommée, 3) la durée de vie du réseau (pour 100 nœuds et pour 200 nœuds), 4) le pourcentage des nœuds vivants pour chaque protocole, 5) l'instant de l'épuisement d'énergie du 1er et du dernier nœud au fil de temps de simulation, 6) la relation entre la position de la BS et la durée de vie, 7) le nombre de clusters par rapport au nombre de nœuds, 8) le nombre de cluster par rapport à la portée de transmission, 9) la distribution des cluster-heads, 10) la latence moyenne des paquets par rapport au nombre de nœuds, 11) le nombre de paquets transmis avec différentes tailles au fil de temps. Les résultats obtenus ont montré que le protocole HEBM est plus performant par rapport aux deux autres.

## Conclusion Générale

Les réseaux de capteurs ont plusieurs contraintes et principalement l'énergie. Ce paramètre permet d'étendre la durée de vie du réseau. Pour cela on essaye d'économiser cette énergie durant les états inactifs.

Le regroupement des nœuds capteurs en groupes est basé sur certains attributs dépendant de leurs emplacements et la valeur de l'énergie résiduelle. Le regroupement est une approche bien connue pour faire face à la densité des nœuds et la conservation efficace de l'énergie dans le réseau. La métrique de la sélection de la tête de groupe Inter-cluster (et intra-cluster) de communication dans plusieurs propositions par l'organisation des chefs de groupes dans une meilleure répartition de l'énergie. L'agrégation des données a été présentée comme un paradigme essentiel pour le routage sans fil dans les réseaux de capteurs. L'idée est de combiner les données provenant de différentes sources pour éliminer la redondance, minimiser le nombre de transmissions et ainsi économiser l'énergie.

Dans cette thèse nous avons proposé un protocole de routage hiérarchique et distribué pour les réseaux de capteurs sans fils. HEBM utilise un schéma de regroupement adaptatif, un schéma de regroupement est appelé un schéma d'adaptation si au fil du temps, le nombre de groupes varie et les membres des nœuds évolue. Le protocole HEBM réalise une bonne répartition des clusters (problème non résolu avec des nombreux protocoles). Le mode de fonctionnement du protocole HEBM se compose de sept phases. La phase d'initialisation durant laquelle chaque nœud calcule un paramètre :  $D(N, BS)$ . La phase de découverte des voisins pour calculer et gérer le nombre des voisins d'un nœud capteur. La phase d'élection de CH temporaire qui permettra d'élire un clusterhead temporaire calculé à partir de quatre paramètres principaux : distance entre le nœud et la station de base, l'énergie résiduelle du nœud et de ses voisins, la distance entre les nœuds voisins, node-weight ce qui signifie le degré d'un nœud ou le nombre de voisins d'un nœud. Ensuite la phase d'élection de clusterhead parmi deux clusterheads temporaires et voisins en utilisant la condition de distance seuil pour choisir entre eux le clusterhead final. Cette technique nous permettra de réaliser un équilibrage de la charge et une distribution idéale sur la surface de captage. Après l'auto-designation des clusterheads on va passer à la phase de formation de groupe, dans cette phase les nœuds non CH doivent estimer *Join-cond* ( $i, CH_j$ ), puis rejoindre les clusters. HEBM opère d'une manière complètement distribuée pour la formation des clusters. Contrairement à

---

---

## *Conclusion Générale (2)*

---

---

la plupart des approches distribuées, notre protocole ne fixe pas le nombre de cluster-heads désiré et n'utilise pas les probabilités pour leur élection. Les noeuds décident d'une manière autonome s'ils peuvent devenir cluster-heads ou pas et ceci selon les paramètres de la fonction de coût proposée. En utilisant l'ordonnanceur TDMA, les membres émettent leurs données captées pendant leurs propres slots. Cela leur permet d'éteindre leurs interfaces de communication en dehors de leurs slots afin d'économiser leurs énergie. Ensuite une communication multisauts intra et inter clusters est réalisé pour obtenir une agrégation de données. Notre contribution a introduit le mécanisme d'équilibrage de la charge grâce à la réalisation d'une bonne distribution des cluster-heads dans le modèle de réseaux et pour améliorer les performances en termes de durée par la réduction du signal de transmission et les messages de contrôle. Le protocole proposé est un protocole efficace en consommation d'énergie et garanti une meilleure distribution des chefs dans le réseau.

Comme perspectives envisageables pour améliorer les performances de notre protocole nous proposons de :

- Gérer la panne des noeuds membres par l'introduction d'un mécanisme permettant la détection de la panne des membres par leur cluster-head.
- Gérer la mobilité des cluster-heads et des noeuds membres en permettant aux noeuds de se déplacer sans perturber le déroulement des opérations du protocole telles que la mise à jour des clusters lors de la migration d'un membre d'un cluster vers un autre. Examiner le cas de plusieurs stations de base et le cas d'une station de base mobile.
- D'autres méthodes moins coûteuses peuvent être envisagées telles que les méthodes de triangulation et la méthode des centroïdes qui donnent une estimation de la position des noeuds.
- Utiliser les algorithmes génétiques pour trouver le chemin le plus court qui relie les clusters head et la station de base.
- Utiliser les réseaux de neurones artificiels pour former les clusters.

## *Nos contributions*

**Auteurs :** Gherbi Chirihane, Aliouat Zibouda et Benmohammed Mohamed

### • **Les revues scientifiques**

**1. Journal of Energy : Elsevier - Science Direct- (2016)**

**Title:** “An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks” Vol. 114, Pages 647-662

**Impact Factor:** IF= 4.292 - (ISI) Thomson Scientific Citation Index.

**2. Journal of Sensor Review – Emerald- (2017)**

**Title:** “A Survey on Clustering Routing Protocols in Wireless Sensor Networks” Vol. 37, Issue 1.

**Impact factor:** IF=**0.898** - (ISI) Thomson Scientific Citation Index, Scopus

**3. Journal of Pervasive Computing and Communications – Emerald- (2016)**

**Title:** “Using adaptive clustering scheme with load balancing to enhance energy efficiency and reliability in delay tolerant with QoS in large-scale mobile wireless sensor networks”, Vol. 12 Issue: 3, pp.352 - 374

Indexé Isi Thomson, Scopus.

**4. Journal of High Performance Computing and Networking – Inderscience (2017)**

**Title:** “Comparative analysis of hierarchical cluster protocols for wireless sensor networks”, “In press”

Inderscience – Scopus (Elsevier)

**5. Journal of Procedia From Awict 2015**

**Title:** “A Load-Balancing and self-adaptation clustering for lifetime prolonging in large scale wireless sensor networks”.

Proceeding AWICT October 5-7, 2015 Sousse- Tunisia

### • **Conférences internationales**

- ✓ IEEE 12 th International Symposium on programming and system ISPS 28-30 april 2015, “ *Distributed Energy Efficient Adaptive clustering protocol for large scale with gathering data in wireless sensor network*”.

## *Nos contributions (2)*

- ✓ International Conference on Embedded Systems in Telecommunications and Instrumentation ICESTI'14 Annaba, Algeria 27-29, 2014 “*DEACP a novel approach with adaptive clustering in WSN*”
- ✓ International Conference CISC2014 December 09-10, 2014 Jijel, “*Distributed Energy Efficient with data gathering in WSN*”.
- ✓ The International Conference on Advanced Wireless, Information, and Communication Technologies AWICT october5-7, 2015 Sousse- Tunisia “*A Load-Balancing and self-adaptation clustering for lifetime prolonging in large scale wireless sensor networks*”.

## *Références Bibliographiques*

- [1] Yi, D. and Yang, H. (2016), "HEER- A Delay-aware and Energy Efficient Routing Protocol for Wireless Sensor Networks" .Journal of Computer Networks. Vol. 104, pp 155-173.
- [2] Linghe, k., Qiao, X., Xue, L., Xiao, L., Xiao, G., Guihai, C. and Min-You, Wu. (2016), "ICP: Instantaneous clustering protocol for wireless sensor networks", Journal of Computer Networks, Vol .101, pp. 144 –157.
- [3] Pengfei, Z., Gaoxi, X. and Hwee, P. (2013), " Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors". Journal of Computer Networks. Vol. 57 No. 14, pp 2689–2704.
- [4] Nooshin, N., Zurina, M., Shamala, S. and Mohamad, A. (2015), "An Energy Aware Distributed Clustering Algorithm Using Fuzzy Logic for Wireless Sensor Networks with Non-uniform Node Distribution". International journal of wireless personal communications. Vol. 84 No.1, pp 395-419.
- [5] Sohail, J., Abid, A. , Muhammad, I. , Shehzad, K. and Kashif, S. (2015), "Energy Efficient Strategy for Throughput Improvement in Wireless Sensor Networks". Journal of Sensors. Vol.15 No.2, pp 2473-2495.
- [6] Khalid, H., Kamalrulnizam, A., Abdul, H. and Tasneem, D. (2016), "Adaptive energy aware cluster-based routing protocol for wireless sensor networks" Journal of Wireless Networks. pp. 1-14
- [7] Hireen, K., Deva, S. and Rajib, M. (2016), "A Hierarchical and Role Based Secure Routing Protocol for Mobile Wireless Sensor Networks" .Journal of Wireless Personal Communications. pp. 1-37.
- [8] Xiaohui, l., Mohamed, E., Hamdy, K., ElMinir, A. and Riad, M. (2016), "A Genetic Algorithm-Based Dynamic Clustering Method Towards Improved WSN Longevity" Journal of Network and Systems Management . pp 1-26.
- [9] Elham, R., Amir , A. (2015) , " Multi-hop Routing Algorithm Using Steiner Points for Reducing Energy Consumption in Wireless Sensor Networks". Journal of Wireless Personal Communications. Vol. 86 No. 3, pp. 1557-1570.
- [10] Singh, A., S., Awadhesh, K. (2015), "Mobility and Energy Conscious Clustering Protocol for Wireless Networks", In: Suresh,C.(Ed.) Advances in Intelligent Systems and Computing :Proceedings of the International Congress on Information and Communication Technology . Springer Singapore, pp 365-374.
- [11] Srikanth, J. and Prasanta, K. (2016) ," A grid based clustering and routing algorithm for solving hot spot problem in wireless sensor networks" Journal of wireless Networks. Vol 22 No 6, pp 1901-1916.
- [12] Juliana, R., Maheswari, U. (2016), "An Energy Efficient Cluster Head Selection Technique Using Network Trust and Swarm Intelligence", Journal of Wireless Personal Communications. Vol. 89, No. 2, pp. 351-364.

## *Références Bibliographiques*

- [13] Subha, C. and Malarkkan, S. (2016), “H-CERP: Energy Efficient Scheme for Industrial WSN Applications” *Journal of Wireless Personal Communications*. Vol. 18 No.4, pp 1-14.
- [14] Guiloufi, A. B. F., N. and Abdennace, K. (2016), “An Energy-Efficient Unequal Clustering Algorithm Using ‘Sierpinski Triangle’ for WSNs”, *Journal of Wireless Personal Communications*. Vol. 88 No. 3, pp. 449-465.
- [15] Sivakumar, B. and Sowmya, B. (2016), “ An Energy Efficient Clustering with Delay Reduction in Data Gathering (EE-CDRDG) Using Mobile Sensor Node”. *Journal of Wireless Personal Communications*. pp 1-14.
- [16] Nitin, G., Mayank, D. and Anil, K. (2016), “ Energy Efficient Architecture for Intra and Inter Cluster Communication for Underwater Wireless Sensor Networks” . *Journal of Wireless Personal Communications*, Vol. 89 No.2, pp. 687-707.
- [17] Akila, S. I., Venkatesan, R. (2016), “A Cognitive Multi-hop Clustering Approach for Wireless Sensor Networks”, *Journal of Wireless Personal Communications*. doi: 10.1007/s11277-016-3200-5.
- [18] Alaei, M., Barcelo, J. (2010), “Node clustering based on overlapping FOVs for wireless multimedia sensor networks”. *Proceedings of the IEEE wireless communication and networking, Sydney, Australia*. doi: 10.1109/WCNC.2010.5506615.
- [19] Bai, F., Mu, H. and Sun, J. (2009), “Power-efficient zoning clustering algorithm for wireless sensor networks”. *Proceedings of the IEEE Information Engineering and computer science, China*, pp 1-4.
- [20] Boubiche, D. and Bilami, A. (2011), “HEEP: Hybrid Energy Efficiency Protocol based on chain clustering”, *Journal of Sensor Networks*. Vol. 10 No. 1/2, pp. 25–35.
- [21] Braginsky, D., Estrin, D. and Rumor, I. (2012), “Routing Algorithm for Sensor Networks”. *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta*, pp. 22–31.
- [22] Gherbi, C., Aliouat, Z. and Benohammed, M. (2016), “An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks”. *Journal of Energy*. Vol. 114, pp 647–662.
- [23] Enamu, H. , Noriko, M. and Norihiko, Y. (2009), “Context-aware cluster-based hierarchical protocol for Wireless Sensor Networks”. *International Journal of Ad Hoc and Ubiquitous Computing*. Vol. 4 No. 6, pp. 227-240.
- [24] Gherbi, C., Aliouat, Z. and Benmohammed, M. (2015), “ Distributed energy efficient adaptive clustering protocol with data gathering for large scale wireless sensor networks”. *Proceedings of 12th IEEE International Symposium on Programming and Systems, Algiers* .pp 128-136.

## *Références Bibliographiques*

- [25] Hemavathi, N., Sudha, S. (2016), "A Novel Regression Based Clustering Technique for Wireless Sensor Networks". *Journal of Wireless Personal Communications*. Vol. 88 No. 4, pp. 985-1013.
- [26] Jan, N., Marek, W. and Maciej, N. (2012), "Communication Activity in Wireless Sensor Networks Based on Migrated Base Stations", *Electrical and Electronic Engineering*, Vol. 8 No. 3, pp.120-131.
- [27] Kameshwaran, K., Malarvizhi, k. (2014), "Survey on Clustering Techniques in Data Mining". *International Journal of Computer Science and Information Technologies* .Vol. 5 No. 2, pp .0975-9646.
- [28] Li, C., Zhang, H., Hao, X. and Li, J. (2011), "A survey on routing protocols for large-scale wireless sensor networks", *Journal of Sensors*, Vol. 11, No. 4, pp. 3498 –3526.
- [29] Nisha, k., Puneet, J. (2015), "A Survey of Clustering Techniques and Algorithms" *Second International Conference on Computing for sustainable Global Development (INDAICom)*, pp. 304-307.
- [30]Nitin, k., Ghanshyam, C. and Ajay, K. (2016), "Effet of multipath fading model on T-ANT clustering protocol for WSN" *Journal of wireless Networks*. Vol. 21 No. 4, pp. 1155-1162.
- [31]Robert, B. (2010),"Wireless sensors: a review of technologies, products and applications", *Journal of Sensor Review*, Vol. 30, No. 4, pp.285 – 289.
- [32] Schurgers, C. and Srivastava, M. (2011), "Energy Efficient Routing in Wireless Sensor Networks". In *Proceedings of Military Communications Conference on Communications for Network-Centric Operations: Creating the Information Force*, McLean, VA, USA. pp. 357–361.
- [33] Shah, R. and Rabaey, J. (2010), " Energy Aware Routing for Low Energy Ad Hoc Sensor Networks". In *Proceedings of the Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, USA, pp. 350–355.
- [34] Taewook, K., Jangkyu, Y. and Hoseung, L. (2010), "A Clustering Method for Energy Efficient Routing in Wireless Sensor Networks". *Proceedings of the 6th WSEAS, Int Conf on Electronics, Hardware, wireless and optical communications*. USA. pp 65-74.
- [35] Victor, S., Firooz, B. and Jonathan, R. (2016), "A Cognitive Self-Organising Clustering Algorithm for Urban Scenarios" *Journal of Wireless Personal Communications*. pp 1–36
- [36] M. Ilyas and I. Mahgoub. ( 2005), "Handbook of sensor networks Compact wireless and wired Sensing Systems", ISBN 08493196864. CRC PRESS LLS, USA.
- [37] R. Kravets and P. Krishnan. (2000), "Application-driven Power Management for Mobile Communication", In *Wireless Networks*, Volume 6, No 4, pages 263–277.
- [38] LEHSAINI Mohamed. "Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique ". Thèse de Doctorat, 2009.

## *Références Bibliographiques*

- [40] Hurni Philipp and Torsten Braun (2009), “Calibrating wireless sensor network simulation models with real-world experiments” . In Proceedings of the 8th International IFIP-TC6 Networking Conference (NETWORKING’09) Springer. pp14, 36.
- [41] Subramaniam1 Shamala, Mohamed Mohamad Afendee. (2015) “An energy aware distributed clustering algorithm using fuzzy logic for wireless sensor networks with non-uniform node distribution”. Int J Wirel Personal Commun. Vol 84, No, 1 pp 395-419.
- [42] Konga Linghe, Xiang b,c Qiao.( 2016)” ICP: instantaneous clustering protocol for wireless sensor networks. Computer Network ,Vol 57, pp 101-144.
- [43] Chirihane Gherbi, Zibouda Aliouat , Mohamed Benmohammed (2016) “An adaptive clustering approach to dynamic load balancing and energy efficiency in wireless sensor networks” Journal of Energy , Vol 114, pp 647-662.
- [44] Zhang a Pengfei, Xiao a Gaoxi, Hwee-Pink. (2013) “Clustering algorithms for maximizing the lifetime of wireless sensor networks with energy-harvesting sensors”. Computer Network ;Vol 57,No14, pp 2689e704.
- [45] Nisha, Kaur Puneet Jai. ( 2015) “A survey of clustering techniques and algorithms”. IEEE 11-13 March:304e7. ISBN 978-9-3805-4415-1.
- [46] Ataul Bari, Shamsul Wazed, Arunita Jaekel, Subir Bandyopadhyay. (2009). “A genetic algorithm based approach for energy efficient routing in two tiered sensor networks” .Journal of Ad-Hoc and Networks7, pp 65-76, DOI:10.1016/j.adhoc.2008.04.003
- [47] Chia-Yen Shih ,Jenks, S.F. (2007) “A Dynamic Cluster Formation Algorithm for Collaborative Information Processing in Wireless Sensor Networks. International Conference Sensor Networks and on Intelligent Sensors, Information Processing. DOI: 10.1109/ISSNIP.2007.4496828
- [48] Attea Baraa A, Khalil Enan A. (2012) “A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks. Appl Soft Comput. Vol 12, No 7, pp 195-207.
- [49] Jan Nikodem, Marek Woda and Maciej Nikodem.(2012) ”Spatial Communication Activity in Wireless Sensor Networks Based on Migrated Base Stations” . Engineering and Electrical and Electronic Engineering. "Wireless Sensor Networks - Technology and Protocols", book edited by Mohammad A. Matin, ISBN 978-953-51-0735-4, DOI: 10.5772/50078.
- [50] Ming Liu, Jiannong Cao, Guihai Chen and Xiaomin Wang (2012). “An Energy-Aware Routing Protocol in Wireless Sensor Networks Sensors”.Journal of Ad-Hoc and Networks. DOI: 10.3390/s90100445.

## *Références Bibliographiques*

- [51] Pedro José Marròn , Stamatis Karnouskos, Daniel Minder and Aníbal Ollero (2010). “State of the Art in Cooperating Objects Research”. Journal of State of the Art in Cooperating Objects Research. 16 December 2010. pp 19- 124 doi : 10.1007/978-3-642-16946-5\_3
- [52] Sajjanhar. U, P. Mitra, (2007). “Distributive Energy Efficient Adaptive Clustering Protocol for Wireless Sensor Networks”, Proceedings of the International Conference on Mobile Data Management, pp. 326 – 330 DOI:10.1109/MDM.2007.69
- [53] Chirihane Gherbi , Zibouda Aliouat, mohammed benmohammed , (2016), "Using adaptive clustering scheme with load balancing to Enhance Energy Efficiency and Reliability in Delay Tolerant with QoS in large scale Wireless Sensor networks", International Journal of Pervasive Computing and Communications, Vol. 12 Iss 3 pp. – pp. 352 – 374.
- [54] Sohel Rana, Ali Newaz Bahar, Nazrul Islam, Johirul Islam (2015) “Fuzzy Based Energy Efficient Multiple Cluster Head Selection Routing Protocol for Wireless Sensor Networks” , International journal of Computer Network and Information Security. DOI: 10.5815/ijcnis.2015.04.07
- [55] Thippeswamy M, Reshma S and Shaila K . (2014).”EDOCR: Energy density on demand cluster routing in wireless sensor networks”. International journal of computer and communication vol 6. DOI:10.512/ijcnc.2014.6115
- [56] Xing Shao, Cui-Xiang Wang, and Yuan Rao. (2015). “Network Coding Aware QoS Routing for Wireless Sensor Network”. Journal of Communications Vol. 10, No. 1. DOI:10.12720/jcm.10.1.24.32.
- [57] Awad M, Abuhasan A. (2016) “A smart clustering based approach to dynamic bandwidth allocation in wireless networks”. Int J Comput Netw Commun, Vol 8, no1, pp 73-86.
- [58] Kowsalya PK, Harikumar R. (2015)” Performance analysis of adaptive routing structure for wireless sensor network based on load balancing”. Wirel Personal Commun. vol 8,pp1-13
- [59] Niu Wenjia, Lei Jun, Tong Endong, Li Gang, Chang Liang, Shi Zhongzhi, et al.(2014)” Context-aware service ranking in wireless sensor networks. J Netw Syst Manag . vol 22, no 1, pp 50-74.
- [60] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. (2000) “Energy-Efficient Communication Protocol for Wireless Microsensor Networks”. InProceedings of the 33rd Annual Hawaii International Conference on System Sciences.

## *Références Bibliographiques*

- [61] Gherbi, C. Aliouat, Z. and Benmohammed, M. (2017), " A Survey on Clustering Routing Protocols in Wireless Sensor Networks ", Sensor Review, Vol. 37 ,No 1 pp. 12-25.
- [62] Mohamed Younis, Moustafa Youssef, and Khaled Arisha. (2002) "Energy-aware routing in cluster-based sensor networks". In Proceedings of 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer, USA pp 129–136.
- [63] Wei Li, Christos G. Cassandras, and Michael Clune.(2006) "Model-Based Design of a Dynamic Voltage Scaling ler Based on Online Gradient Estimation Using SimEvents". In Proceedings of the 45th IEEE Conference on Decision , San Diego, CA, USA. pp 113-129.
- [64] Krisztián Flautner, Steve Reinhardt, Trevor Mudge. (2009) "Automatic performance setting for dynamic voltage scaling. Journal of Wireless Networks" , Kluwer Academic Publishers Hingham, MA, USA, Volume 8, Issue 5, pp 507 – 520.
- [65] Udit Sajjanhar Pabitra Mitra.(2007) "Distributive Energy Efficient Adaptive Clustering Protocol for Wireless Sensor Networks" the IEEE International Conference on Mobile Data Management.October , San Diego, USA. pp 96-118
- [66] Arjun P. Athreya and Patrick Tagu. (2011). "Towards Secure Multi-path Routing for Wireless Mobile Ad-Hoc Networks: A Cross-layer Strategy". 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp.146-148. DOI:10.1109/SAHCN.2011.5984886
- [67] Aravind Iyer, Sunil S. Kulkarni, Vivek Mhatre, Catherine P. Rosenberg (2008). "A taxonomy- based approach to design of large-scale sensor networks". Journal of Wireless Sensor Networks and Applications Signals and Communication Technology Springer, pp 3-33.





# Annexe-1

---


## Méthode d'intégration d'un protocole sous ns2




### 1. L'installation de NS2.34 et L'intégration de LEACH






- L'installation de NS2.34 sous Ubuntu 10.10

-  Copier le package d'installation « ns-allinone2.34 » dans votre dossier personnel.
-  Décompresser le fichier ns-allinone-2.34.tar.gz
-  Extraire le fichier dans le même répertoire.
-  Lancer la mise à jour de « gcc-4.3 » avec la commande:

```
sudo apt-get install build-essential autoconf automake libxmu-dev
```

-  Modifier le fichier « makefile.in » comme suit:

-  Accéder à l'emplacement /ns-allinone-2.34/otcl-1.13
-  Ouvrir le fichier makefile.in.
-  Modifier CC= @CC@ à : CC= gcc-4.3

-  Ouvrir un terminal et aller au répertoire : /ns-allinone2.34/ ns2.34.
-  Ensuite taper la commande : « ./install »
-  Puis taper « ./validate »
-  Ouvrir le fichier *bashrc* par la commande « gedit ~/.bashrc » dans un terminal.
-  Ensuite ajouter à la fin de fichier les lignes suivantes :

**#LD\_LIBRARY\_PATH**

OTCL\_LIB=/home/administrateur/ns-allinone-2.34/otcl-1.13

NS2\_LIB=/home/administrateur/ns-allinone-2.34/lib

X11\_LIB=/usr/X11R6/lib

USR\_LOCAL\_LIB=/usr/local/lib

export

LD\_LIBRARY\_PATH:\$OTCL\_LIB:\$NS\_LIB:\$X11\_LIB:\$USR\_LOCAL\_LIB

**#TCL\_LIBRARY**

TCL\_LIB=/home/administrateur/ns-allinone-2.34/tcl8.4.18/library

USR\_LIB=/usr/lib

export TCL\_LIBRARY = TCL\_LIB:\$USR\_LIB


**#PATH**

XGRAPH=/home/administrateur/ns-allinone-2.34/bin:/home/administrateur/ns-allinone-2.34/tcl8.4.18/unix:/home/administrateur/ns-allinone-2.34/tk8.4.18/unix

NS=/home/administrateur/ns-allinone-2.34/nam-1.13/

PATH=\$PATH:\$XGRAPH:\$NS:\$NAM

 Sauvegarder le fichier *bashrc*.

 Tapez dans un terminal : le mot '*ns*', il doit afficher le symbole '%'. Dans ce cas le ns a été bien installé.

 Installer **xgraph** à l'aide de la commande: **sudo apt-get install xgraph**

### • Intégration de LEACH sous ns2.34

Après l'installation ns2.34 sous Ubuntu 10.10, l'étape suivante consiste à intégrer le protocole LEACH par les instructions suivantes :

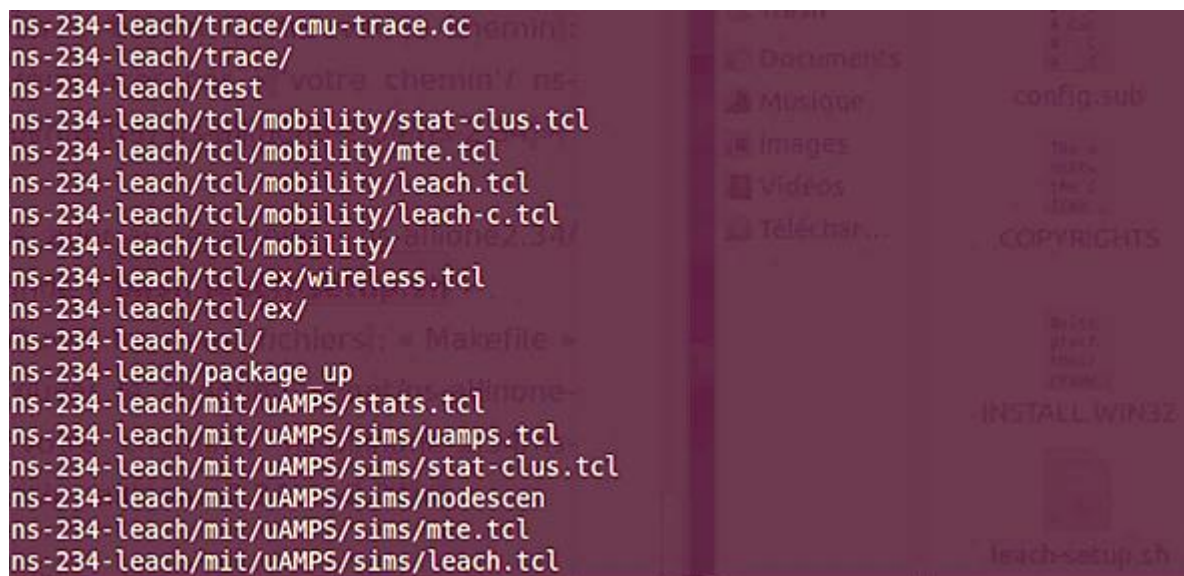
- Télécharger le paquet « **ns-234-leach.tar.gz** »
- Télécharger le fichier « **leach-setup.sh** »

- Placer le package « ns-234-leach » et le fichier « leach-setup.sh » du protocole LEACH dans le chemin suivant : « home/ 'votre chemin' / ns-allinone-2.34/ ns-2.34 » ;

Ex : « Home/gherbi/ -allinone-2.34/ns-2.34 »

1. Dans le fichier " Leach-setup.sh ",on besoin de trouver le chemin : « /opt/ns-allinone-2.34 »; et le remplacer par « 'votre chemin' » ex : « Home/gherbi/ ns-allinone-2.34/ns-2.34 »

2. Dans un terminal de commandes, aller au répertoire « ns-allione2.34/ ns2.34 » ensuite taper la commande « **bash leach-setup.sh** ».



```
ns-234-leach/trace/cmu-trace.cc
ns-234-leach/trace/
ns-234-leach/test
ns-234-leach/tcl/mobility/stat-clus.tcl
ns-234-leach/tcl/mobility/mte.tcl
ns-234-leach/tcl/mobility/leach.tcl
ns-234-leach/tcl/mobility/leach-c.tcl
ns-234-leach/tcl/mobility/
ns-234-leach/tcl/ex/wireless.tcl
ns-234-leach/tcl/ex/
ns-234-leach/tcl/
ns-234-leach/package up
ns-234-leach/mit/uAMPS/stats.tcl
ns-234-leach/mit/uAMPS/sims/uamps.tcl
ns-234-leach/mit/uAMPS/sims/stat-clus.tcl
ns-234-leach/mit/uAMPS/sims/nodescen
ns-234-leach/mit/uAMPS/sims/mte.tcl
ns-234-leach/mit/uAMPS/sims/leach.tcl
```

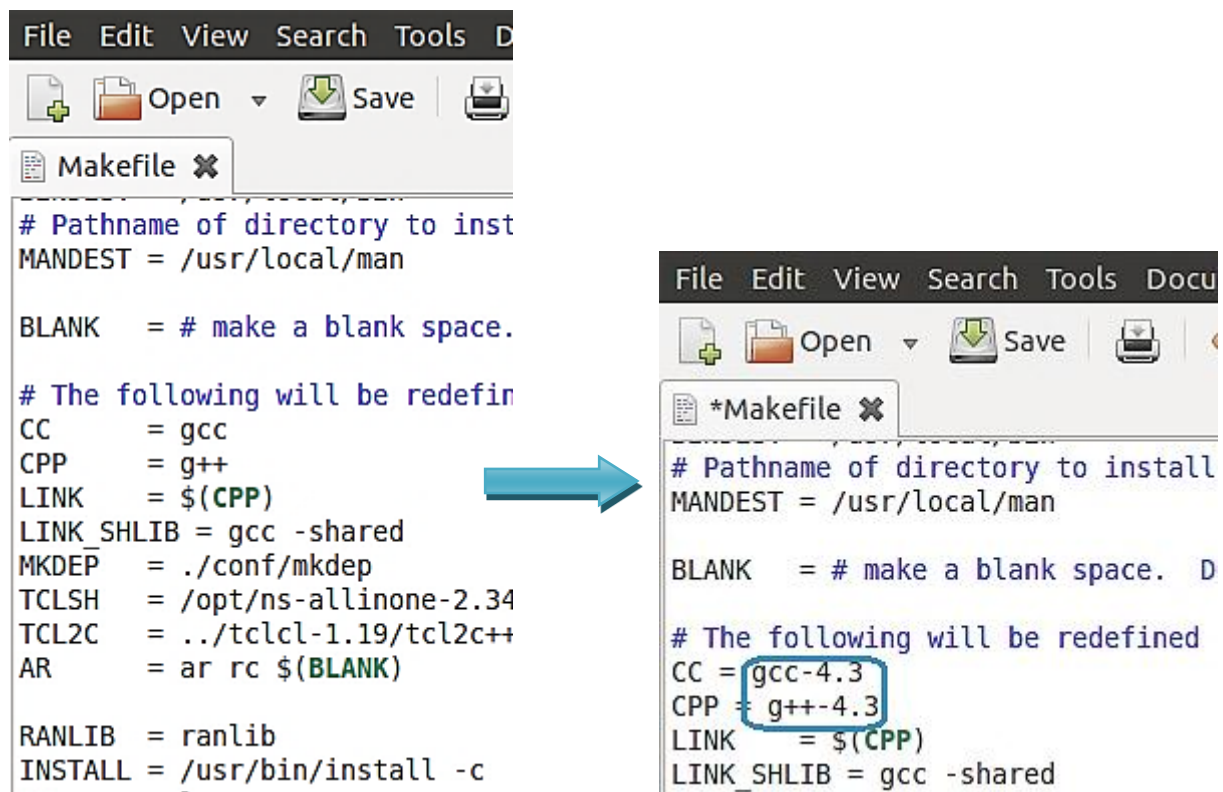
3. Dans le répertoire « ns-2.34», Ouvrir les deux fichiers : « Makefile » et « Makefile.in » ; il doit trouver le chemin : « /opt/ns-allinone-2.34 »;et le remplacer par « 'votre chemin' / ns-allinone-2.34/ns-2.34 » ex : « Home/gherbi/ ns-allinone-2.34/ns-2.34 »

Modifier les deux fichiers comme suit :

CC = gcc-4.3

CPP = g++-4.3

Et faire la recherche sur le chemin « /opt/ », puis vous devez le remplacer par votre chemin.



4. Exécuter les commande suivante : « `./configure` »

```
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking how to run the C preprocessor... gcc -E
checking for grep that handles long lines and -e... /bin/grep
```

5. Taper la commande « **make clean** » : `:/opt/ns-allinone-2.34/ns-2.34:$make clean`.

L'exécution de la commande « **make clean** » affiche les erreurs suivantes :

```

gherbi@gherbi-HP-PC: ~/ns-allinone-2.34/ns-2.34
File Edit View Search Terminal Help
AR -DNDEBUG -DLINUX_TCP_HEADER -DUSE_SHM -DHAVE_LIBTCLCL -DHAVE_TCLCL_H -DHAVE_
LIBOTCL1_13 -DHAVE_OTCL_H -DHAVE_LIBTK8_4 -DHAVE_TK_H -DHAVE_LIBTCL8_4 -DHAVE_TC
LINT_H -DHAVE_TCL_H -DHAVE_CONFIG_H -DNS_DIFFUSION -DSMAC_NO_SYNC -DCPP_NAMESPA
CE=std -DUSE_SINGLE_ADDRESS_SPACE -Drng_test -DMIT_uAMPS -I. -I. -I/home/gherbi
/ns-allinone-2.34/tclcl-1.19 -I/home/gherbi/ns-allinone-2.34/otcl -I/home/gherbi
/ns-allinone-2.34/include -I/home/gherbi/ns-allinone-2.34/include -I/home/gherbi
/ns-allinone-2.34/include -I/usr/include/pcap -I./tcp -I./sctp -I./common -I./li
nk -I./queue -I./adc -I./apps -I./mac -I./mobile -I./trace -I./routing -I./tools
-I./classifier -I./mcast -I./diffusion3/lib/main -I./diffusion3/lib -I./diffusi
on3/lib/nr -I./diffusion3/ns -I./diffusion3/filter_core -I./asim/ -I./qs -I./dif
fserve -I./satellite -I./wpan -I./mit/rca -I./mit/uAMPS -o trace/cmu-trace.o trac
e/cmu-trace.cc
trace/cmu-trace.cc: In member function 'void CMUTrace::format_rca(Packet*, int)'
:
trace/cmu-trace.cc:1531: error: 'ADV_CHAR' was not declared in this scope
trace/cmu-trace.cc:1535: error: 'REQ_CHAR' was not declared in this scope
trace/cmu-trace.cc:1539: error: 'DATA_CHAR' was not declared in this scope
make: *** [trace/cmu-trace.o] Error 1
gherbi@gherbi-HP-PC:~/ns-allinone-2.34/ns-2.34$

```

On a corrigé la première erreur :

```

trace/cmu-trace.cc: In member function 'void CMUTrace::format_rca(Packet*, int)'
:

```

Comme suit : Par l'insertion de fonction « **format\_rca** » dans le fichier **cmu-trace.h** dans le répertoire trace de ns2.34

```

File Edit View Search Tools Documents Help
Open Save Undo
*cmu-trace.h
void format_smac(Packet *p, int offset);
void format_ip(Packet *p, int offset);

void format_arp(Packet *p, int offset);
void format_hdlc(Packet *p, int offset);
void format_dsr(Packet *p, int offset);
void format_msg(Packet *p, int offset);
void format_tcp(Packet *p, int offset);
void format_sctp(Packet *p, int offset);
void format_rtp(Packet *p, int offset);
void format_tora(Packet *p, int offset);
void format_imep(Packet *p, int offset);
void format_aodv(Packet *p, int offset);
void format_aomdv(Packet *p, int offset);
void format_rca(Packet *p, int offset);

// This holds all the tracers added at run-time
static PacketTracer *pktTrc_;

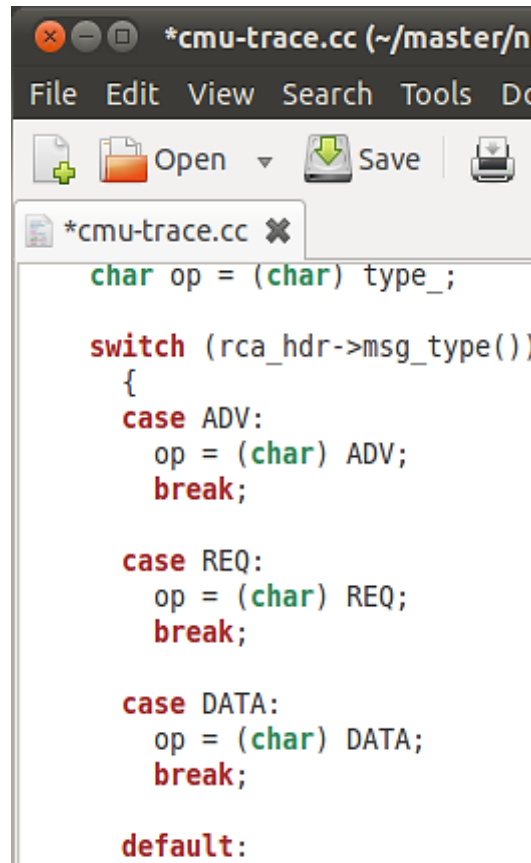
};

```

On a corrigé la deuxième erreur :

```
trace/cmu-trace.cc:1531: error: 'ADV_CHAR' was not declared in this scope
trace/cmu-trace.cc:1535: error: 'REQ_CHAR' was not declared in this scope
trace/cmu-trace.cc:1539: error: 'DATA_CHAR' was not declared in this scope
```

Comme suit : par une modification de la méthode de déclaration des variables dans le fichier « **cmu-trace.cc** » dans le répertoire « **trace** » de ns2.34.



```
char op = (char) type_;

switch (rca_hdr->msg_type())
{
case ADV:
    op = (char) ADV;
    break;

case REQ:
    op = (char) REQ;
    break;

case DATA:
    op = (char) DATA;
    break;

default:
```

- Après l'exécution de la commande **make clean** sans aucune erreur, afin d'effacer l'ancien fichier de compilation.
  - Et enfin recompiler de nouveau NS2 afin de prendre en considération l'ajout du nouveau protocole de routage LEACH en tapant la commande « **make** » dans le terminal de commandes. `./opt/ns-allinone-2.34/ns-2.34$make`.
  - Si l'exécution de la commande précédente est réussie alors on peut tester LEACH par l'exécution de la commande : « `./test` » : `./opt/ns-allinone-2.34/ns-2.34$./test`.
- ✓ **Le problème posé c'est** : après qu'on a tapé la commande « `./test` », On remarque que l'exécution ne marche pas. **La solution** : Il faut obliger d'ajouter les quatre lignes suivantes dans le fichier `bashrc`.

Ouvrir le fichier **bashrc** par la commande « `gedit ~/.bashrc` » et on ajoute :

```
RCA_LIB=/home/ 'votre chemin '/ns-allinone-2.34/ns-2.34/mit/rca  
uAMPS_LIB=/home/ 'votre chemin '/ns-allinone-2.34/ns-2.34/mit/uAMPS  
export RCA_LIBRARY=$RCA_LIB
```

- ✓ Ré-exécutez la commande : make puis la commande. /test
- ✓ Vérifier le fichier « leach.err », s'il n'y a pas d'erreurs, ouvrir le fichier « leach.out » et vous voyez les résultats.

# Annexe-2

## Code d'implémentation de chaque phase du protocole HEBM (Code source des fonctions principales)

### 1. La phase d'initialisation

```
Application/HEBM instproc HEBMfunction {} {
global ns_ opt
$self instvar now_
  $ns_ at [expr $now_ + $random_accessh] "$self discov_neigh_msg"
  $ns_ at [expr $now_ + $random_accessh] "$self Energie"
  $ns_ at [expr $now_ + $random_accessh] "$self Dist-bs"
  $ns_ at [expr $now_ + $opt(ra_adv)* $opt(nn_)+$random_accessh] "$self decideClusterHeadtemp"
  #puts "appel a discov_neigh_msg"
  $ns_ at [expr $now_ + $opt(ra_adv)* $opt(nn_) + $opt(ra_adv)* $opt(nn_)] "$self decideClusterHead"
}
```

### 2. La phase de découverte des voisins « Neighbor Discovery (NW) »

```
Application/HEBM instproc discov_neigh_msg {} {
  global ns_ opt ADV_h MAC_BROADCAST LINK_BROADCAST BYTES_ID
  $self instvar currentCH_ code_ now_ random_access

  set ID [$self nodeID]
  puts "discov_neigh_msg "
  set mac_dst $MAC_BROADCAST
  set link_dst $LINK_BROADCAST
  set msg [list $ID]
  set datasize [expr $BYTES_ID * [llength $msg]]
  $Send beacons opt(max dist) meters so all nodes can hear.
  $self send $mac_dst $link_dst $ADV_h $msg $datasize $opt(max_dist) $code_
  puts " $ID a envoye un discov_neigh_msg "
}
```

```
Application/HEBM instproc recv_discov_neigh_msg {msg} {
  global ns_
  $self instvar listneigh_ discov_neigh_
  set discov_neigh_ [lindex $msg 0]
  set nodeID [$self nodeID]
  pp "$nodeID rcvd $discov_neigh_ from $nodeID at [$ns_ now]"
  set listdiscov_neigh_ [lappend listdiscov_neigh_ $msg]
  puts " $nodeID a recu un discov_neigh_ = $discov_neigh_ "
}
```

### 3. La phase d'élection de Cluster head temporaire

```

Application/HEBM instproc decideClusterHeadtemp {} {

    global ns_ opt ADV_h MAC_BROADCAST LINK_BROADCAST BYTES_ID send_dst bs send_thr
    $self instvar currentCH_ code_ ID_ listneigh_ listdst_ listenergie_ cost dist E   dst dist
    set ID [$self nodeID]
    #pp "Cluster Head $currentCH_ broadcasting ADV at time [$ns_ now]"
    set mac_dst $MAC_BROADCAST
    set link_dst $LINK_BROADCAST
    set dist [nodeToBSDist [$self node] $bs]

    foreach element $listneigh_ {
    set neigh [lindex $element 0]
    }
    set neigh $neigh

    foreach element $listenergie_ {
    set Enrg [lindex $element 0]
    }
    set Enrg $Enrg

    foreach element $listdst_ {
    set dst [lindex $element 0]
    }
    set dst $dst

```

```

# Calculate the condition CHtemp
set nn $opt(nn_)
for {set id 0} {$id < [expr $opt(nn)-1]} {incr id} {
    set thresh2 [expr (1- ($cost / $neigh))]
    set thresh3 [expr double (1- ($dist / $dst))]
    set thresh1 [expr (1- ($E / $Enrg))]

    set thresh [expr (1- ($thresh1 + $thresh2 + $thresh3 ))]
}
set msg [list [list $ID $thresh]]
set datasize [expr $BYTES_ID * [llength $msg]]
# Send beacons opt(max dist) meters so all nodes can hear.
$self send $mac_dst $link_dst $send_thr $msg $datasize $opt(max_dist) $code
}

```

### 3. La phase d'élection de Cluster head final

```

Application/HEBM instproc decideClusterHead {} {

    global chan ns_ opt node_ E

    $self instvar next_change_time_ round_ clusterNodes_ listthresh_
    $self instvar now_ TDMAschedule_ beginningE_ alive_
    $self instvar myADVnum_ CHheard_ listneigh_ E

    set CHheard_ 0
    [$self mac] set CHheard_ $CHheard_
    set myADVnum_ 0

    set ISalive [[[$self node] set netif_(0)] set alive_]
    if {$alive_ == 1 && $ISalive == 0} {
        puts "Node [$self nodeID] is DEAD!!!! Energy = [[[$self getER] query]"
        $chan removeif [[[$self node] set netif_(0)]
        set alive_ 0
        set opt(nn_) [expr $opt(nn_) - 1]
    }
}

```

```

    if {$round_ == 0} {
        $self hasnotbeenClusterHead
    }

    if [[{$self hasbeenClusterHead?}] {
        set thresh 0
    }

    set clusterNodes_ ""
    set TDMAschedule_ ""

set dal 0
set min_thr 100000
    foreach element $listthresh_ {
        set chID [lindex $element 0]
        set thresh [lindex $element 1]
        if {$thresh < $min_thr} {
            set min_thr $thresh
            set currentCH_ $chID
            set numCodesAvail [expr 2 * $opt(spreading) - 1]

        }
    }

    set min_thr $min_thr
    puts " $min_thr "
    set currentCH_ $chID

```

```

    foreach element $listthresh_ {
        set chID [lindex $element 0]
        set thresh [lindex $element 1]

        if { $thresh <= $min_thr} {
            set bol 1

            if { $bol == 1 } {
                $self setClusterHead
                $ns_ at [expr [$ns_ now] + $random_accessh] "$self advertiseClusterHead"
                puts "$nodeID is final cluster head at time [$ns_ now] CAR J'AI LE Max Chtemp"
            }

            } else {

                $self unsetClusterHead
            }

        }

    incr round_
    set next_change_time_ [expr $now_ + $opt(ch_change)]
    $ns_ at $next_change_time_ "$self debut"
    $ns_ at [expr [$ns_ now] + $opt(ra_adv_total)] "$self findBestCluster "
}

```

```

Application/HEBM instproc rcvthr {msg} {
    global ns_
    $self instvar clusterChoices_ clusterDist_ thresh listthresh_
    set ID [lindex $msg 0]
    set thresh [lindex $msg 1]
    set nodeID [$self nodeID]
    pp "$nodeID rcvd thresh from $nodeID at [$ns_ now]"
    set listthresh_ [lappend listthresh_ $msg]
    puts " $nodeID $ID $listthresh_ "
}

```

## 4. La formation des clusters

```

Application/HEBM instproc informClusterHead {} {
    global ns_ opt JOIN_REQ MAC_BROADCAST BYTES_ID
    $self instvar currentCH_ dist_ code_

    set nodeID [$self nodeID]
    set chID $currentCH_
    pp "$nodeID: sending Join-REQ to $chID (dist = $dist_) at time [$ns_ now]"
    set mac_dst $MAC_BROADCAST
    set link_dst $chID
    set msg [list $nodeID]
    set spreading_factor $opt(spreading)
    set datasize [expr $spreading_factor * $BYTES_ID * [llength $msg]]
    $self WakeUp
    set join_req [expr ($thresh/($dist-$dst))]
    # NOTE!!!! Join-Req message sent with enough power so all nodes in
    # the network can hear the message. This avoids the hidden terminal
    # problem.
    $self send $mac_dst $link_dst $JOIN_REQ $msg $datasize $opt(max_dist) $code_
}

```

## 5. Création des TDMA

```

Application/HEBM instproc createSchedule {} {
    global ns_ opt ADV_SCH MAC_BROADCAST BYTES_ID
    $self instvar clusterNodes_ TDMAschedule_
    $self instvar dist_ code_ now_ beginningE_
    set numNodes [llength $clusterNodes_]
    set chID [$self nodeID]
    if {$numNodes == 0} {
        set xmitOrder ""
        puts "Warning! There are no nodes in this cluster ($chID)!"
        $self SendMyDataToBS
    } else {
        # Set the TDMA schedule and send it to all nodes in the cluster.
        set xmitOrder $clusterNodes_
        set msg [list $xmitOrder]
        set spreading_factor $opt(spreading)
        set datasize [expr $spreading_factor * $BYTES_ID * [llength $xmitOrder]]
        pp "$chID sending TDMA schedule: $xmitOrder at time [$ns_ now]"
        pp "Packet size is $datasize."
        set mac_dst $MAC_BROADCAST
        set link_dst $chID
        $self send $mac_dst $link_dst $ADV_SCH $msg $datasize $dist_ $code_
    }
    set TDMAschedule_ $xmitOrder
    set outf [open $opt(dirname)/TDMAschedule.$now_.txt a]
    puts $outf "$chID\t$TDMAschedule_"
    close $outf
    set outf [open $opt(dirname)/startup.energy a]
    puts $outf "[ $ns_ now ]\t$chID\t[expr $beginningE_ - [[self getER] query]] "
    close $outf
}

```

```

Application/HEBM instproc recvADV_SCH {order} {

  global ns_opt
  $self instvar xmitTime_ next_change_time_ now_
  $self instvar beginningE_ frame_time_ end_frm_time_

  set nodeID [$self nodeID]
  set ind [lsearch [join $order] $nodeID]
  set outf [open $opt(dirname)/startup.energy a]
  puts $outf "[Sns_ now]\t$nodeID\t[expr $beginningE_ - [[self getER] query]]"
  close $outf
  if {$ind < 0} {
    puts "Warning!!!! $nodeID does not have a transmit time!"
    puts "Must send data directly to BS."
    set outf [open $opt(dirname)/TDMAschedule.$now_.txt a]
    puts -nonewline $outf "$nodeID\t"
    close $outf
    $self SendMyDataToBS
    return
  }
  # Determine time for a single TDMA frame. Each node sends data once
  # per frame in the specified slot.
  set frame_time_ [expr [expr 5 + [llength [join $order]]] * $opt(ss_slot_time)]
  set xmitTime_ [expr $opt(ss_slot_time) * $ind]
  set end_frm_time_ [expr $frame_time_ - $xmitTime_]
  set xmitat [expr [Sns_ now] + $xmitTime_]
  pp "$nodeID scheduled to transmit at $xmitat. It is now [Sns_ now]."
  if {[expr $xmitat + $end_frm_time_] < \
      [expr $next_change_time_ - 10 * $opt(ss_slot_time)]} {
    Sns_ at $xmitat "$self sendData"
  }
  $self GoToSleep
}

```

## 6. Transmission Multisauts Intra-cluster et Inter-cluster

```

Application/HEBM instproc TrackNode {} {

  global node_opt ch ns_node_ now_ bs
  $self instvar nextHopID_ dist_ currentCH_ cluster_
  $self instvar now_ upstream_ clusterNodes_ clusterH_

  [$self mac] set node_num_ [$self nodeID]

  set nodeID [$self nodeID]
  set x [$node_($nodeID) set X_]
  set y [$node_($nodeID) set Y_]

  set chID $currentCH_
  set chx [$node_($chID) set X_]
  set chy [$node_($chID) set Y_]

  set cx [expr ($x + $chx) / 2]
  set cy [expr ($y + $chy) / 2]
  set dist_ch [dist $x $y $chx $chy]
  set r [expr $dist_ch / 2]

  set minx $chx
  set miny $chy
  set minID $chID
  set mind $dist_ch
  set CHID $clusterH_

  foreach element $cluster_ {
    set max [lindex $element 0]
    if { $CHID == $chID } {
      set maxd $max
      set R2 $maxd
      set R1 [expr $maxd - (1 * $maxd / 2)]
    }
  }
}

```

```

if { $dist_ch > 0 && $dist_ch <= $R1 } {
  set nextHopID_ $chID
  set mind $dist_ch
  puts "-----+"
  puts "N1 / node : [$self nodeID] next hop neighbor is CH $nextHopID_ (dist = $mind)"
  if { $minID < $opt(bsID) } {
    [$node_($minID) set rca_app_] addUpStreamNeighbor $nodeID
  }
}
#-----
if { $dist_ch > $R1 && $dist_ch <= $R2 } {
  for {set i 0} {$i < $opt(nn_)} {incr i} {
    set i $i
    set nx [$node_($i) set X_]
    set ny [$node_($i) set Y_]
    set incircle [dist $nx $ny $cx $cy]
    if {$incircle < $r && ($i != $nodeID) && ($chID == $chID)} {
      set distToCH [dist $nx $ny $chx $chy]
      if { $distToCH > 0 && $distToCH <= $R1 } {
        set d [dist $x $y $nx $ny]
        if {$d < $mind} {
          set mind $d
          set minx $nx
          set miny $ny
          set minID $i
        }
      }
    }
  }
}
set nextHopID_ $minID
set dist_ $mind
puts "N2 / node: [$self nodeID] next hop neighbor is $nextHopID_ (dist = $mind / CH = $chID)"

```

```

if { $minID < $opt(bsID) && $minID != $chID } {
  [$node_($minID) set rca_app_] addUpStreamNeighbor $nodeID
}

$self sendData
set startup_energy [expr $opt(nn_) * 1e-9]
puts "\tstartup_energy = $startup_energy"
[$self getER] remove $startup_energy
}
#-----
$self check Alive
set clusterH_ ""
}
}
}

```

```

Application/HEBM instproc TrackCH {} {

  global node_ opt bs ns_ node_ node_
  $self instvar nextHopID_ dist_ currentCH_ clusterheadChoices_
  $self instvar upstream_ niveau3_ niveau2_ niveau1_

  $self setCode $opt(bsCode)
  set maxd 0
  set bsx [lindex $bs 0]
  set bsy [lindex $bs 1]
  foreach element $clusterheadChoices_ {
    set ind [lsearch $clusterheadChoices_ $element]
    set chx [$node_($element) set X_]
    set chy [$node_($element) set Y_]
    set d [dist $chx $chy $bsx $bsy]
    if {$d > $maxd} {
      set maxd $d
    }
  }
  set dist_ $maxd
#-----
  set r3 $maxd
  set r2 [expr $maxd - (1 * $maxd/30)]
  set r1 [expr $maxd - (10 * $maxd/30)]

```

```

set CHID $currentCH_
set chx [$node_($CHID) set X_]
set chy [$node_($CHID) set Y_]
set cx [expr ($chx + $bsx) / 2]
set cy [expr ($chy + $bsy) / 2]
set dtobs [nodeToBSDist $node_($CHID) $bs]
set r [expr $dtobs / 2]

```

```

} else {
    foreach element $niveau2_ {
        set chID [lindex $element 0]
        set ind [lsearch $niveau2_ $element]
        set nx [$node_($chID) set X_]
        set ny [$node_($chID) set Y_]
        set incircle [dist $nx $ny $cx $cy]
        if {$incircle < $r && ($CHID != $chID)} {
            set d [dist $chx $chy $nx $ny]
            if {$d < $mind} {
                set mind $d
                set minx $nx
                set miny $ny
                set minID $chID
            }
        }
    }
}

```

```

#-----
if { $dtobs > $r1 && $dtobs <= $r2 } {
    foreach element $niveau1_ {
        set chID [lindex $element 0]
        set ind [lsearch $niveau1_ $element]
        set nx [$node_($chID) set X_]
        set ny [$node_($chID) set Y_]
        set incircle [dist $nx $ny $cx $cy]
        if {$incircle < $r && ($CHID != $chID)} {
            set d [dist $chx $chy $nx $ny]
            if {$d < $mind} {
                set mind $d
            }
        }
    }
}

```

```

    set minx $nx
    set miny $ny
    set minID $chID
}
}
}
#-----
set nextHopID_ $minID
set dist_ $mind

puts "*****"
puts "CH : [$self nodeID] next hop neighbor is $nextHopID_ (dist = $mind)"

if {$minID < $opt(bsID)} {
    [$node_($minID) set rca_app_] addUpStreamNeighbor $CHID
}
puts "*****"
#-----
set leng [llength $upstream_]

if { $leng == 0 } {
    $self sendDataNextHop
    # Finding next hop neighbor costs 1 nJ/node (guess) to compute distances.
    set startup_energy [expr $opt(nn_) * 1e-9]
    pp "\tstartup_energy = $startup_energy"
    [$self getER] remove $startup_energy
}

```

```

Application/HEBM instproc MaxNode {} {
    global node_opt ns_node_node_Jeton MAC_BROADCAST BYTES_ID now_
    $self instvar currentCH_ dist_ code_ clusterH_ dist_ clusterNodes_ now_ nextHopID_

    set chID $currentCH_
    set nodeID [$self nodeID]
    set maxd 0
    set chx [$node_($chID) set X_]
    set chy [$node_($chID) set Y_]

    set numNodes [[length $clusterNodes_]]
    if { $numNodes == "" } {
        set listeNodes ""
        puts "Warning! There are no nodes in this cluster ($chID)!"
        $self sendDataToBS
    } else {

    foreach element $clusterNodes_ {
        set ind [lsearch $clusterNodes_ $element]
        set nx [$node_($element) set X_]
        set ny [$node_($element) set Y_]
        set d [dist $chx $chy $nx $ny]
        if {$d > $maxd} {
            set maxd $d
        }
    }
    set dist $maxd

    set mac_dst $MAC_BROADCAST
    set link_dst $chID
    set msg [list $maxd]
    set spreading_factor $opt(spreading)
    set datasize [expr $spreading_factor * $BYTES_ID * [length $msg]]

```

```

    set chx [$node_($chID) set X_]
    set chy [$node_($chID) set Y_]

    set numNodes [[length $clusterNodes_]]
    if { $numNodes == "" } {
        set listeNodes ""
        puts "Warning! There are no nodes in this cluster ($chID)!"
        $self sendDataToBS
    } else {

    foreach element $clusterNodes_ {
        set ind [lsearch $clusterNodes_ $element]
        set nx [$node_($element) set X_]
        set ny [$node_($element) set Y_]
        set d [dist $chx $chy $nx $ny]
        if {$d > $maxd} {
            set maxd $d
        }
    }
    set dist $maxd

    set mac_dst $MAC_BROADCAST
    set link_dst $chID
    set msg [list $maxd]
    set spreading_factor $opt(spreading)
    set datasize [expr $spreading_factor * $BYTES_ID * [length $msg]]
    $self send $mac_dst $link_dst $Jeton $msg $datasize $dist_ $code_
    pp " $chID : sending Jeton at time [$ns_ now] / Max_dist : $maxd "
}
}

```

```

#####
Application/HEBM instproc addUpStreamNeighbor id {
    $self instvar upstream_
    set nodeID [$self nodeID]
    set upstream_ [lappend upstream_ $id]
    puts " [$self nodeID] ==> upstream_ : $upstream_ "
}

#####
Application/HEBM instproc removeUpStreamNeighbor id {
    $self instvar upstream_
    set index [lsearch $upstream_ $id]
    set upstream_ [lreplace $upstream_ $index $index]
}

#####

```