

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie « Houari Boumediene »
Faculté d'Electronique et d'Informatique



MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTERE

En : INFORMATIQUE

Spécialité : Informatique Mobile

Par : BENMANSOUR Tariq

Sujet

**LA GÉSTION DES GROUPES DANS LES RÉSEAUX DE
CAPTEURS MOBILES**

Soutenu publiquement le 25/05/2011, devant le jury composé de :

Mr- M. AHMED-NACER, Professeur à l'USTHB.

Président

Mme- S. MOUSSAOUI, Maître de Conférences/A à l'USTHB.

Directrice de mémoire

Mr- A. BELKHIR, Professeur à l'USTHB.

Examineur

Mme- F. KHELLAF, Maître de Conférences/A à l'USTHB.

Examinatrice

RESUME

Notre travail consiste à la conception d'une nouvelle technique de clustering pour les réseaux de capteurs mobiles qui répond aux défis imposés par la ressource énergétique limitée des nœuds. Notre objectif est double : (a) minimiser la consommation d'énergie en réduisant les coûts liés à la communication des paquets et (b) répartir la charge de travail uniformément sur l'ensemble des nœuds formant le réseau. Ceci maximisera la durée de vie du réseau.

Pour se faire, nous avons étudié dans un premier temps les approches de clustering proposées pour les réseaux de capteurs avec des nœuds fixes. Ensuite, nous avons étudié les techniques existantes traitant le cas des nœuds capteurs mobiles. Ces solutions ont été analysées et critiquées. Elles nous ont permis de concevoir, dans un second temps, deux nouveaux protocoles de clustering plus efficaces en consommation d'énergie et qui prennent en considération la mobilité des nœuds.

Notre contribution a été d'apporter en premier lieu, une amélioration au protocole de clustering ACE-L. Cette amélioration consiste à ajouter des mécanismes pour que le protocole résultant (DACE-L) s'adapte à la variation de la densité des nœuds dans le réseau. En second lieu, nous avons conçu une nouvelle technique de clustering (GMAC) qui supporte la mobilité des nœuds. Elle se base sur un facteur de mobilité des nœuds en groupe pour sélectionner leur responsable de cluster (cluster-head).

Les simulations effectuées ont permis de monter l'efficacité de nos solutions concernant la réduction de l'énergie consommée par les nœuds capteurs, participant ainsi que dans le prolongement de la durée du réseau.

Mots clés : Clustering, Réseaux de capteurs, Algorithmes distribués, modèles de mobilité.

Remerciements

Louange à notre Seigneur 'ALLAH' qui m'a doté de la merveilleuse faculté de raisonnement. Louange à notre créateur qui m'a incité à acquérir le savoir. C'est à lui que s'adresse toute ma gratitude en premier lieu.

En second lieu, je tiens à remercier Mme S.MOUSSAOUI ma directrice de mémoire pour sa sympathie, sa disponibilité, ses idées, ses conseils et ses encouragements qui m'ont permis de mener à bien mon travail.

Je remercie particulièrement les honorables membres de jury qui ont accepté de juger et d'évaluer ce mémoire.

Je tiens aussi à remercier l'ensemble des enseignants de l'USTHB, sans exception, ainsi que les employés qui ont rendu plus confortable notre formation au sein de l'université.

Sans oublier de remercier tous les membres du laboratoire d'intelligence artificielle de l'école militaire polytechnique qui m'ont offert un excellent cadre de travail.

Un grand **MERCI** à tous les membres de ma famille et spécialement mes parents et mon épouse qui m'ont soutenu tout au long de mes études et qui ont fait, en sorte, par leur amour et leur affection, que je puisse avoir les meilleures conditions possibles pour continuer mes études et aller de l'avant. Qu'ils trouvent ici ma gratitude et mon amour pour eux.

Sommaire

Introduction générale	1
Chapitre 1	4
Généralités sur les réseaux de capteurs sans fil	4
Introduction	4
1.1. Définition d'un nœud capteur sans fil	4
1.2. Les réseaux de capteurs sans fil	8
1.2.1. Description	8
1.2.2. Types de réseaux de capteurs	9
1.2.3. Conception des réseaux de capteurs sans fil	10
1.2.4. Architectures adoptées pour les réseaux de capteurs	12
1.3. Domaines d'application des réseaux de capteurs	13
1.3.1. Applications militaires	13
1.3.2. Applications environnementales	14
1.3.3. Applications médicales	15
1.3.4. Applications domestiques	15
1.3.5. Autres applications commerciales et industrielles	16
Conclusion	16
Chapitre 2	18
Techniques de clustering dans les réseaux de capteurs sans fil	18
Introduction	18
2.1. Définition	19
2.2. Principes de construction de clusters	20
2.2.1. Election des cluster-heads	20
2.2.2. Communication intra-cluster et inter-cluster	21
2.2.3. Maintenance des clusters	22
2.3. Objectifs de clustering	23
2.3.1. Équilibrage de la charge (<i>Load balancing</i>)	23
2.3.2. Connectivité augmentée et latence réduite	23
2.3.3. Durée de vie étendue du réseau	24
2.3.4. Passage à l'échelle (<i>scalability</i>)	24
2.4. Classification des techniques de clustering	25
2.4.1. Propriété des clusters	25
2.4.2. Caractéristiques des cluster-heads	26
2.4.3. Processus de clustering	27
2.5. Techniques de clustering pour les réseaux de capteurs statiques	28

2.5.1. LEACH (<i>Low Energy Adaptive Clustering Hierarchy</i>)	28
2.5.2. HEED (<i>Hybrid Energy-Efficient Distributed Clustering</i>)	32
2.5.3. PEGASIS (<i>Power-Efficient Gathering in Sensor Information Systems</i>)	33
2.5.4. TEEN (<i>Threshold-sensitive Energy Efficient sensor Network protocol</i>)	35
2.5.5. EECM (<i>An Energy Efficient Clustering Method for Wireless Sensor Networks</i>)	37
2.5.6. PC (<i>Passive Clustering</i>)	38
Conclusion	40
Chapitre 3	42
Techniques de clustering dans les réseaux des capteurs mobiles	42
Introduction	42
3.1. Les réseaux de capteurs mobiles	43
3.1.1. Types de capteurs mobiles	43
3.1.2. Architectures des MWSNs (<i>Mobile Wireless Sensor Networks</i>)	43
3.1.3. WSNs et MWSNs	44
3.2. Techniques de clustering dans les réseaux de capteurs mobiles	45
3.2.1. ACE-L (<i>Algorithm of Cluster-head Election by Location</i>)	46
3.2.2. CM (<i>Clustering with Mobility</i>)	48
3.2.3. Protocole de clustering basé zone	48
3.2.4. CES (<i>Cluster-based Energy-efficient Scheme for Mobile Wireless Sensor Networks</i>)	50
3.2.5. LEACH- Mobile	53
3.2.6. LEACH-ME (<i>Mobile-Enhanced</i>)	55
3.2.7. CBR-Mobile	56
Conclusion	59
Chapitre 4	61
Protocole de clustering distribué pour les réseaux de capteurs mobiles	61
Introduction	61
4.1. Modèle d'environnement et hypothèses	63
4.2. Density based ACE-L	63
4.2.1 Génération des points de référence	65
4.2.2. Sélection des cluster-heads	69
4.2.3. Construction des clusters	69
4.2.4. Maintenance des clusters	72
4.3. GMAC: (<i>Distributed Group Mobility Adaptive Clustering algorithm MWSNs</i>)	73
4.3.1 Métrique de mobilité en groupe (<i>MobilityGroup</i>)	74
4.3.2. Sélection des cluster-heads	79
4.3.3. Définition des zones :	80
4.3.4. Construction des clusters	81

Conclusion	81
Chapitre 5	82
Simulations et résultats	82
Introduction	82
5.1. Environnement de simulation	82
5.1.1 Description de l'environnement de développement.....	83
5.1.2. Définition du système	84
5.2. Exemple d'application	86
5.3. Paramètres de simulation.....	89
5.4. Les métriques d'évaluation de performances.....	90
5.4.1 Energie Moyenne Restante (EMR)	91
5.4.2 Nombre moyen de paquets délivrés à la station de base (NMPL)	91
5.4.3 Durée de vie du réseau	91
5.5. Discussion des résultats.....	92
Conclusion	100
Conclusion générale	101
Références bibliographiques	103

Liste des figures

Figure 1. Architecture générale d'un nœud capteur sans fil.....	6
Figure 2. Exemple d'un capteur sans fil MICAZ de Crossbow.....	7
Figure 3. Schéma général d'un réseau de capteurs.	9
Figure 4. Architecture de communication de données dans un réseau de capteurs	12
Figure 5. Exemple de clustering dans les WSNs.	19
Figure 6. La représentation d'un WSN par un graphe.....	20
Figure 7. Communication inter cluster via des nœuds passerelles.	22
Figure 8. Communication inter cluster via des cluster-heads.....	22
Figure 9. Formation de clusters par LEACH.	29
Figure 10. Opérations de l'étape d'initialisation de LEACH.....	29
Figure 11. Collecte de données dans PEGASIS.....	34
Figure 12. Les architectures (a) plate, (b) 2-tiers, (c) 3-tiers	44
Figure 13. La sélection des cluster-heads dans ACE-L	46
Figure 14. La sélection des cluster-heads	50
Figure 15. Les phases de clustering dans CES	52
Figure 16. Le mécanisme de maintenance des clusters dans LEACH-Mobile	55
Figure 17. Le mécanisme de maintenance des clusters dans CBR Mobile.....	57
Figure 18. Mécanisme d'adaptation du TDMA dans CBR-Mobile	58
Figure 19. Points de références principaux et secondaires.....	64
Figure 20. Points de référence principaux et secondaires	65
Figure 21. Exemple d'exécution de la méthode K-MEANS pour k=5,6 et 7 zones	67
Figure 22. La répartition des points de référence par des ellipses	68
Figure 23. Maintenance des clusters.....	73
Figure 24. La mobilité en groupe dans les MWSNs (ex : application militaire)	74
Figure 25. Exemple d'évaluation du paramètre <i>GroupMobility</i>	77
Figure 26. L'estimation de la position dans GMAC	78
Figure 27. Les valeurs possibles de T_{pred}	79
Figure 28. La division de la scène en zones.....	81
Figure 29. Aperçu de la topologie lors du déploiement du réseau	86

Figure 30. Aperçu de la topologie au début de la deuxième période à l'aide de l'outil NAM.....	88
Figure 31. Durée de vie du réseau en fonction du nombre de clusters	92
Figure 32. La consommation d'énergie en fonction du temps.....	94
Figure 33. La durée de vie du réseau pour DACE-L	95
Figure 34. Nombre de paquets de données reçues par la station de base en fonction du temps pour DACE-L.....	96
Figure 35. Energie moyenne restante en fonction du temps pour GMAC	97
Figure 36. La durée de vie du réseau pour GMAC	98
Figure 37. Nombre de données reçues par la station de base pour GMAC	98
Figure 38. Durée de vie du réseau en fonction de la taille des groupes et leurs diamètres pour GMAC	99
Figure 39. Energie moyenne restante en fonction du temps pour GMAC avec une mobilité suivant le modèle RWM.....	100

Liste des tableaux

Tableau 1. Tableau comparatif des techniques de clustering pour les réseaux de capteurs mobiles.....	59
Tableau 2. Paramètres utilisés dans l'exemple.....	87
Tableau 3. Données recueillies au début de la deuxième période	87
Tableau 4. Exécution d'ACE-L.....	88
Tableau 5. Exécution de GMAC	89
Tableau 6. Exécution de DACE-L.....	89
Tableau 7. Paramètres de simulation	90
Tableau 8. Nombre de clusters générés par les protocoles ACE-L et DACE-L	93

Introduction générale

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

De nombreux domaines d'application sont alors envisagés pour ces réseaux, tels que le contrôle de l'environnement, la détection et la surveillance des désastres, le domaine militaire et médical, la domotique et bien d'autres. En effet, la petite taille des nœuds capteurs impose des limites sur leurs capacités de traitement, de stockage et surtout d'énergie; car ils sont généralement alimentés par des batteries à capacités limitées. Recharger les batteries dans un réseau de capteurs est parfois impossible en raison de l'emplacement des nœuds, de la taille de ces réseaux et de leur localisation. Il est donc couramment admis que la limitation énergétique est une question incontournable dans la conception des réseaux de capteurs en raison des contraintes strictes qu'elle impose sur son exploitation.

A l'heure actuelle, l'effort essentiel de la recherche porte sur les moyens de réduire au minimum la consommation d'énergie afin de maximiser la durée de vie du réseau. Ceci est à travers l'utilisation de techniques de conservation innovantes portants sur tous les niveaux de la hiérarchie du réseau : physique, MAC et routage. En effet, si nous voulons qu'un réseau de capteurs fonctionne de manière satisfaisante aussi longtemps que possible, nous devons prendre en considération la contrainte énergétique imposée par ses nœuds capteurs. De façon générale, économiser l'énergie revient à optimiser la communication (envoi/réception) des paquets qui représente la principale source de consommation d'énergie.

Une solution couramment utilisée dans l'optimisation de l'énergie consommée par les communications, adopte le clustering dans l'organisation du réseau. Ce concept permet de concevoir des protocoles de communication efficaces en énergie en limitant les messages et en évitant la leur redondance.

D'autre part, la mobilité des nœuds est une question clé pour certains types de réseaux de capteurs, telles que les applications de surveillance où des nœuds capteurs sont embarqués sur, par exemple, des dispositifs mobiles ou sur des animaux. Lorsque cette mobilité devient importante, elle doit être considérée comme un problème nouveau et important sur ce type d'environnement où les solutions classiques déjà élaborées n'apportent pas de réponses satisfaisantes. En effet, des problèmes tels que la localisation des nœuds et la reconfiguration du réseau exigent habituellement un nombre important de messages de contrôle échangés entre les nœuds, ce qui cause un surcoût en énergie sur le réseau. En outre, un autre type de mobilité pourrait concerner la station de base. Cette dernière est généralement exploitée pour assurer des tâches adaptées de collecte et d'agrégation de données.

Dans ce travail, nous nous intéressons à l'étude des techniques de clustering qui consistent globalement à construire une topologie virtuelle au dessus de la topologie physique. Cette topologie virtuelle est constituée de groupes ou clusters. Un cluster est vu comme un nœud abstrait qui est en fait un sous-ensemble des nœuds du réseau. Cette abstraction apporte de nombreux avantages dont la possibilité d'optimiser le nombre des communications nécessaires à la réalisation d'une tâche donnée. Par exemple, lors d'une diffusion, il suffit de diffuser l'information vers les clusters ou les représentants des groupes (*cluster-heads*); ces derniers se chargent alors de la propagation de l'information au niveau de leur groupe respective. Nous nous intéressons aux environnements des capteurs mobiles où la mobilité soulève de nouvelles questions auxquelles il faut apporter des solutions adaptées. Dans ce cadre, nos contributions proposent des techniques adaptées aux clustering pour les réseaux de capteurs mobiles. Ces propositions apportent des réponses satisfaisantes aux défis imposés par la ressource énergétique limitée des nœuds. Leur apport est double : (a) minimiser la consommation d'énergie en réduisant les frais liés à la communication des paquets et (b) répartir la charge de travail uniformément sur l'ensemble des nœuds formant le réseau. L'intérêt est alors de maximiser la durée de vie du réseau. Pour se faire, nous avons étudié dans un premier temps les approches de clustering proposées pour les réseaux de capteurs avec des nœuds fixes. Ensuite, nous avons étudié les techniques existantes traitant le cas des nœuds capteurs mobiles. Ces solutions ont été analysées et critiquées. Elles nous ont permis de concevoir, dans un second temps, deux nouveaux protocoles de clustering plus efficace en consommation d'énergie et qui prend en considération la mobilité des nœuds.

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

Le premier chapitre présente des généralités sur les réseaux de capteurs sans fil avec une description de leurs architectures et leurs caractéristiques principales ainsi que leurs domaines d'application. Dans le deuxième chapitre, nous présentons dans un premier temps le clustering, ses principes et ses objectifs, ensuite nous présentons les classifications possibles des techniques proposées pour les réseaux de capteurs avec nœuds fixes.

Dans le troisième chapitre, nous présentons la mobilité dans les réseaux de capteurs et les techniques de clustering proposées dans ce contexte. Une étude détaillée et des critiques sont apportées à ces méthodes. Le quatrième chapitre est consacré à la description détaillée de nos contributions. Ces dernières sont proposées comme solutions aux problématiques posées par la mobilité des nœuds en termes de construction efficace de clusters de capteurs mobiles. Une évaluation des performances de nos propositions a été réalisée sous le simulateur NS2, les résultats obtenus sont discutés et analysés dans le cinquième chapitre. Enfin, nous terminons par une conclusion générale qui résume nos conclusions tirées de l'étude de l'état de l'art et de la discussion des résultats obtenus par nos solutions et présente les perspectives envisageables comme amélioration de notre travail.

Chapitre 1

Généralités sur les réseaux de capteurs sans fil

Introduction

Les progrès réalisés durant ces dernières décennies dans les domaines de la microélectronique, de la micromécanique et des technologies de communication sans fil ont permis de produire à un coût raisonnable des composants de quelques millimètres cubes de volume. De ce fait, un nouveau domaine de recherche s'est créé pour offrir des solutions économiquement intéressantes et pratiques pour la surveillance à distance et le traitement des données dans des environnements complexes et distribués ; à savoir les réseaux de capteurs sans fil et leurs applications.

Les réseaux de capteurs ont de nombreuses perspectives d'application dans des domaines très variés : applications militaires, surveillance industrielle ou surveillance des phénomènes naturels etc. De ce fait, les micro-capteurs sont de véritables systèmes embarqués. Le déploiement de plusieurs d'entre eux, en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome, forme un réseau de capteurs.

Dans le présent chapitre, nous détaillons les principaux concepts liés aux réseaux de capteurs. En premier lieu, nous définissons l'élément de base dans les WSNs (*Wireless Sensor Networks*), le capteur sans fil et nous détaillons son architecture interne. Puis, nous donnons une description des WSNs en présentant les caractéristiques, les paramètres entrant dans leur conception et leurs différentes applications possibles.

1.1. Définition d'un nœud capteur sans fil

D'une façon générale, un capteur est un dispositif qui répond au stimulus physiques (tels que la chaleur, la lumière, le bruit, la pression, le magnétisme, etc.) et il converti la quantité ou le paramètre de stimulus physique à des signaux enregistrables (tels que signaux électriques, signaux mécaniques, etc.). Ces signaux sont numérisés pour produire des données de détection. Les capteurs représentent une interface entre le monde physique et le monde des dispositifs

électriques, tels que les ordinateurs. Les capteurs facilitent aux personnes de comprendre, de surveiller, et de commander des machines et des environnements.

Les capteurs diffèrent par leurs types et leurs formes, ils peuvent mesurer presque toutes sortes de stimulus physiques. Par convention, les capteurs peuvent être classifiés dans différents types selon le phénomène à mesurer [1]; on recense des capteurs qui mesurent des quantités mécaniques, fluides, électriques, thermiques, chimiques, biologiques, etc.

En plus de la fonctionnalité essentielle d'un capteur qui est la détection et la mesure d'un phénomène précis, le nœud capteur sans fil (appelé aussi couramment "capteur"), quand à lui, inclut également d'autres unités pour traiter et fournir des données de détection.

Un capteur sans fil est un dispositif physique, de taille miniature, qui a pour but essentiel d'accomplir trois tâches complémentaires : le relevé d'une grandeur physique, le traitement éventuel de cette information, et la communication avec d'autres capteurs afin de transmettre cette information. Pour pouvoir accomplir ces tâches, le capteur est composé principalement de quatre unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie [2]. Selon le domaine d'application, il peut aussi contenir des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellules photovoltaïques). Quelques micro-capteurs, plus volumineux, sont dotés d'un système mobilisateur chargé de les déplacer en cas de nécessité.

Nous donnons ci-dessous les rôles des unités de base d'un nœud capteur :

- **L'unité de captage** : généralement composée de deux sous-unités : le récepteur et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique. Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement.
- **L'unité de traitement** : elle comprend un processeur généralement associé à une petite unité de stockage. Elle fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs. Elle exécute les protocoles de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud collecteur.

- **L'unité de transmission** : elle effectue toutes les émissions et réceptions des données dans un milieu sans-fil. Elle peut être de type optique (comme dans les nœuds *Smart Dust* [3]), ou de type radiofréquence.
 - Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, ne pouvant pas établir des liaisons à travers des obstacles, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes.
 - Les unités de transmission de type radiofréquence comprennent des circuits de modulation, démodulation, filtrage et multiplexage ; ceci implique une augmentation de la complexité et du coût de production du micro-capteur.
- Concevoir des unités de transmission de type radiofréquence avec une faible consommation d'énergie est un défi car pour qu'un nœud ait une portée de communication suffisamment grande, il est nécessaire d'utiliser un signal assez puissant et donc une énergie consommée importante. L'alternative consistant à utiliser de longues antennes n'est pas possible à cause de la taille réduite des micro-capteurs.
- **Unité de contrôle d'énergie** : l'unité de contrôle d'énergie est un système essentiel. Elle doit répartir l'énergie disponible aux autres modules et unités, de manière optimale (par exemple en réduisant les dépenses inutiles et en mettant en veille les composants inactifs). Cette unité peut aussi gérer des systèmes de recharge d'énergie à partir de l'environnement via des cellules photovoltaïques par exemple.

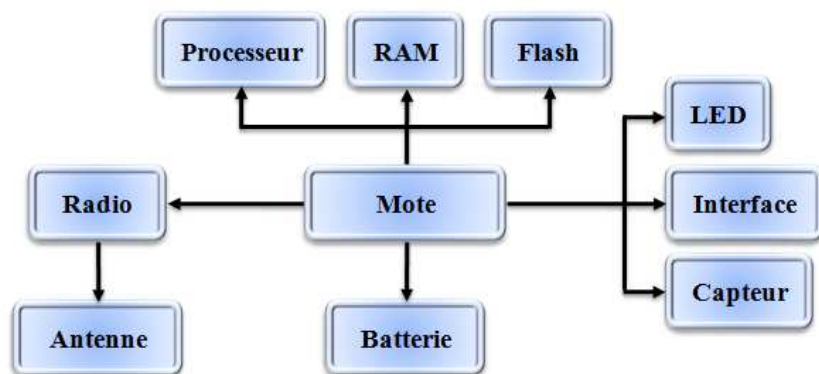


Figure 1. Architecture générale d'un nœud capteur sans fil.

Généralement les capteurs ont la même structure matérielle, elle est représentée par le Figure 1.

- **Mote, Processeur, RAM, Flash** : c'est la carte physique, composée de processeur, RAM et Flash, l'ensemble assure les traitements binaires et le stockage. La taille d'un programme qui s'exécute sur le capteur est limitée par la taille de la RAM.
- **Radio et antenne** : pour pouvoir communiquer avec l'ensemble du réseau, un capteur est équipé d'un radio émetteur/récepteur et d'une antenne.
- **LED, Interface, Capteur** : le but principal d'un capteur est de capter. Il est équipé de plusieurs types de détecteurs des différents phénomènes à observer, de plusieurs LEDs qui indiquent l'état de certains composants selon leurs valeurs (on/off) et d'interfaces de communications (Serial, RG45, ...) ou d'autres interfaces d'extensions.
- **Batterie** : le capteur est équipé d'une batterie qui assure son alimentation autonome. Certains capteurs sont dotés d'un panneau solaire permettant le rechargement continu de cette batterie.

Le nœud capteur sans fil est mis en application par un panneau de capteur qui intègre tous les composants mentionnés ci-dessus et d'autres circuits nécessaires. Actuellement, beaucoup de nœuds capteurs ont été fabriqués dans le cadre de la recherche et du développement des réseaux de capteurs sans fil. La Figure 2 présente un nœud capteur sans fil MICAZ fabriqué par Crossbow Corporation [4]. La couche supérieure du nœud est le panneau de capteur MPR2400TM [5], de 58mm×32mm×7mm de taille et 18g de poids. Le microcontrôleur est basé sur l'Atmel ATmega128L et trois types de mémoire, à savoir, mémoire flash de programme 128KB, mémoire de configuration EEPROM 4KB et une mémoire de mesure 512KB.



Figure 2. Exemple d'un capteur sans fil MICAZ de Crossbow.

Le panneau de capteur peut supporter plusieurs types de capteurs, y compris acoustiques, lumineux, la magnéto compteur et le capteur d'accéléromètre. Il comporte aussi un convertisseur analogiques-numériques (ADC) de 10 bits. Un nœud MICAZ fonctionne avec une fréquence de 2.4 GHz et un débit qui va jusqu'à 250 Kbps, conformément à la norme IEEE 802.15.4. Le fond de capteur est une caisse de batterie qui peut inclure deux batteries de type AA. La consommation électrique du microcontrôleur en mode actif est de 8mA et en mode veille est moins de 15µA. La consommation de l'émetteur-récepteur est de 19.7mA en mode de réception, 17.4mA en mode transmission et 20µA en mode inactivité (avec le régulateur de tension allumée) et de 1µA en mode veille (avec le régulateur de tension éteinte).

1.2. Les réseaux de capteurs sans fil

1.2.1. Description

Les réseaux de capteurs forment une sous-classe des réseaux mobiles ad-hoc. Les nœuds capteurs, déployés en grand nombre, sont capables de récolter et de transmettre des données environnementales d'une manière autonome. Les positions des nœuds ne sont pas obligatoirement prédéterminées. Les capteurs sont dispersés aléatoirement dans une zone géographique, appelée champ de capture, qui définit le terrain d'intérêt pour le phénomène capturé. Cela implique que les protocoles réseau utilisés doivent posséder des capacités d'auto-organisation. Une autre caractéristique des réseaux de capteurs est la coopération des capteurs. Les données capturées sont acheminées grâce à un routage multi-sauts vers un nœud considéré comme "point de collecte", appelé nœud collecteur (Station de Base ou *Sink*). Ce dernier peut être connecté à l'utilisateur du réseau via un satellite ou Internet. L'utilisateur peut ainsi adresser des requêtes aux nœuds du réseau en précisant le type de données requises, les données environnementales capturées seront récoltées par le collecteur. Les capteurs sont équipés d'un processeur embarqué. Au lieu d'envoyer des données brutes aux nœuds responsables de la fusion, ils utilisent leurs capacités de traitement pour effectuer localement des traitements et transmettre les données nécessaires partiellement traitées [2].

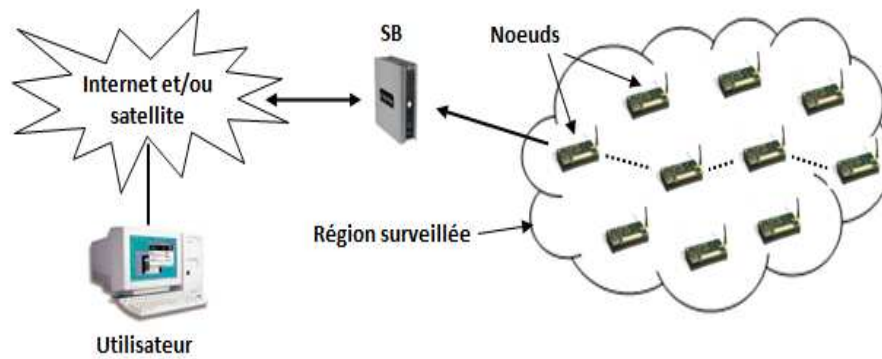


Figure 3. Schéma général d'un réseau de capteurs.

Chaque application a ses propres contraintes. Dans tous les domaines, le rôle d'un réseau de capteurs est cependant le même. Les nœuds doivent surveiller certains phénomènes grâce à leurs capteurs puis envoient les informations capturées à un collecteur. Le collecteur est un nœud particulier doté généralement d'une puissance de calcul supérieure et d'une quantité d'énergie potentiellement infinie. Ce collecteur peut être connecté à Internet ou possède un lien radio de type GSM ou GPRS qui lui permet d'envoyer les informations (données ou alertes) à un centre de contrôle pour l'utilisateur final (Figure 3). Il peut y avoir plusieurs collecteurs mobiles ou fixes dans un réseau mais pour des raisons de coût, il y a beaucoup moins de collecteurs que de nœuds ordinaires.

1.2.2. Types de réseaux de capteurs

Les réseaux de capteurs peuvent être classés selon différents critères, suivant la mobilité des nœuds, les réseaux de capteurs sans fil peuvent être de deux types selon [6] :

- **Réseaux de capteurs mobiles** : le réseau est constitué d'un ensemble de capteurs mobiles évoluant dans un environnement statique. Le but de tels réseaux est souvent l'exploration de zones inaccessibles ou dangereuses. Nous trouvons souvent des travaux de recherche orientés vers la robotique où les nœuds jouent à la fois le rôle de capteur et d'actionneur. Un actionneur étant un capteur particulier doté de caractéristiques importante en puissance d'énergie et de calcul. Ce type de nœud a la capacité de prise de décision en fonction des données récoltées.
- **Réseaux de capteurs fixes (stationnaires)** : le réseau est constitué de capteurs fixes servant à la surveillance d'occurrence d'évènements dans une zone géographique. Dans ce cas, le réseau n'effectue que la surveillance. Les données mesurées sont

transmises en mode multi-sauts au nœud collecteur. Ce dernier est chargé après réception de l'information, de mettre en œuvre les actions nécessaires. Ce collecteur peut être connecté, de manière filaire à un autre réseau. Ce type de réseaux de capteurs est le plus répandu et le plus traité dans les travaux de recherche.

Selon la nature de la station de base, les réseaux de capteurs peuvent être aussi répartis en deux classes :

- **Réseaux de capteurs avec station de base fixe** : les positions des collecteurs dans le réseau de capteurs sont connues à l'avance par tous les nœuds du réseau.
- **Réseaux de capteurs avec station de base mobile** : les collecteurs peuvent être statiques ou mobiles, ils peuvent se déplacer vers des positions déterminées pour interroger le réseau et réaliser une collecte optimale des données capturées.

1.2.3. Conception des réseaux de capteurs sans fil

La conception des réseaux de capteurs est influencée par de nombreux paramètres. Ces paramètres ont été traités par de nombreux chercheurs. Toutefois, aucune de ces études n'a pu intégrer tous les paramètres qui sont à l'origine de la conception des réseaux de capteurs. Ces paramètres sont importants car ils constituent un guide pour la conception des réseaux de capteurs. En outre, ces paramètres déterminants peuvent être utilisés pour comparer et analyser les performances d'un WSN.

Nous définissons ci-après ces paramètres [2] :

- **Tolérance aux fautes** : l'échec des capteurs dû aux différentes raisons telles que l'épuisement d'énergie, dommages physiques ou la présence d'obstacles ne devrait pas affecter la tâche globale du réseau. La tolérance aux fautes est la capacité de maintenir les fonctionnalités du réseau sans aucune interruption due à la défaillance de quelques capteurs. Le niveau de tolérance aux fautes devrait être adapté en fonction de l'hostilité du milieu dans le quel est déployé le réseau.
- **Passage à l'échelle** : le nombre de capteurs dans un réseau dépend de la nature du phénomène observé. Dans certains cas, il peut atteindre l'échelle des milliers voir même des millions de capteurs. Les réseaux de capteurs doivent être en mesure de supporter ce

très grand nombre de capteurs. Ainsi, les protocoles et les algorithmes développés doivent fonctionner correctement malgré ses dimensions.

- **Les contraintes matérielles :** comme nous l'avons détaillé plus haut, le capteur est composé de plusieurs unités de bases en plus d'autres modules supplémentaires. Tous ces composants consomment de l'énergie. La capacité de la batterie d'alimentation est donc la contrainte la plus forte, d'autres contraintes s'imposent sur la taille et le volume de capteur, tous ces composants doivent être regroupés dans un volume qui ne dépasse pas quelques centimètres de cube, et un poids assez léger. Par ailleurs, les transmissions radios sont plus complexes et consomment plus d'énergie que les liaisons optiques.
- **La topologie du réseau :** la topologie d'un réseau de capteurs est en perpétuel changement à cause de la défaillance des capteurs, de l'épuisement d'énergie des capteurs, des obstacles qui se présentent dans le réseau ou du besoin d'ajouter d'autres capteurs. Ces raisons engendrent une nécessité de gestion des changements et une maintenance de la topologie. On distingue trois phases [6]:
 - **Le pré-déploiement :** les nœuds capteurs peuvent être déployés d'une manière aléatoire (largués en masse d'un avion par exemple). Ils peuvent aussi être placés un par un, soit par un humain ou un robot dans des positions prédéterminées, on parle alors d'un déploiement déterministe.
 - **Le post-déploiement :** après le déploiement, les changements de topologie sont dus aux changements des états des capteurs : la position, l'accessibilité (à cause de brouillage, obstacles mobiles), disponibilité de l'énergie, dysfonctionnement, etc.
 - **Redéploiement des nœuds supplémentaires :** d'autres nœuds supplémentaires peuvent être redéployés à tout moment pour remédier aux dysfonctionnements des nœuds.
- **Environnement :** un capteur peut être déployé tout près de l'objet à surveiller ou carrément à l'intérieur de l'objet. Ceci peut être l'intérieur d'une grosse machine, le fond d'un océan, un lieu contaminé biologiquement ou chimiquement, dans un champ de bataille au-delà des lignes ennemies, dans une maison ou un immeuble, etc. Ces situations très variées engendrent de très fortes contraintes sur l'environnement de déploiement des capteurs.

- **Support de transmission** : il est clair que dans ces réseaux, la communication se fait via des liens sans fil. Ces liens peuvent être formés par radio, infrarouge, ou un support optique. La majorité des capteurs utilisent la radiofréquence comme support de transmission. La transmission via infrarouge ou support optique est résistante aux interférences provoquées par les appareils électriques, mais elle est écartée car elle exige une vue directe entre l'émetteur et le récepteur.
- **La consommation en énergie** : la seule source d'énergie dans un capteur est la batterie qui fournit une quantité d'énergie très limitée. Généralement, l'énergie emmagasinée n'est pas renouvelable. Les transmissions de données entre nœuds demandent beaucoup de traitement, surtout, si des nœuds tombent en panne ce qui exige la réorganisation du réseau. C'est pour quoi, des algorithmes et des protocoles ont été spécialement développés avec comme principale contrainte l'économie d'énergie, parfois, même au détriment de la qualité de service fournie par le réseau.

1.2.4. Architectures adoptées pour les réseaux de capteurs

Les architectures dans les réseaux de capteurs dépendent des applications et des techniques utilisées pour faire acheminer l'information des capteurs à la station de base.

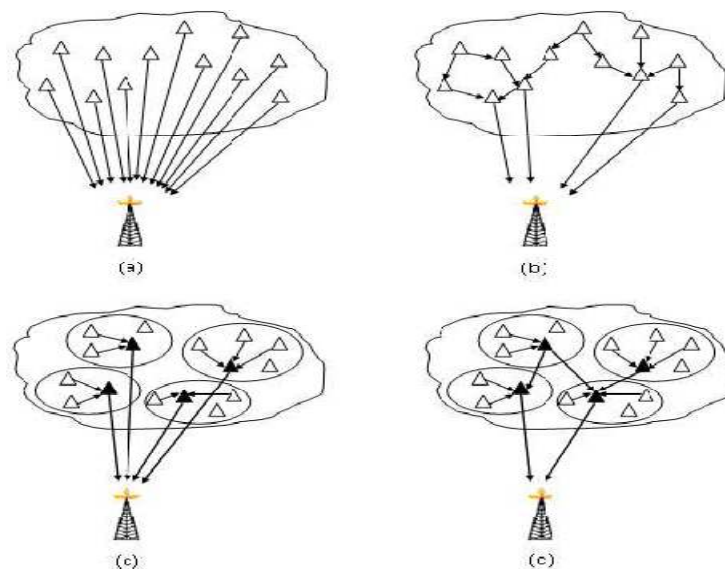


Figure 4. Architecture de communication de données dans un réseau de capteurs

Le processus d'acheminement de l'information des capteurs à la station de base peut prendre quatre formes. Dans les architectures plate, les capteurs peuvent communiquer directement avec la station de base en utilisant une forte puissance (Figure 4 (a)), ou via un mode multi-sauts avec des puissances très faibles (Figure 4 (b)), alors que dans les architectures hiérarchisées, le nœud représentant le groupe (cluster), appelé cluster-head, transmet directement les données à la station de base (Figure 4 (c)), ou via un mode multi-saut entre les cluster-heads (Figure 4 (d)).

1.3. Domaines d'application des réseaux de capteurs

Un réseau de capteurs peut être composé de différents types de capteurs : sismiques, magnétiques, thermiques, visuel, infrarouge, acoustiques, et beaucoup d'autres types qui sont capables de surveiller une large variété de conditions ambiantes telle que : la température, l'humidité, le mouvement, la pression, la présence de certains objets, etc. Les capteurs sont caractérisés essentiellement par leur coût très réduit et leurs souplesses de support de communication utilisé : le sans fil. Ces avantages ont permis aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre les applications existantes et de faciliter la conception d'autres systèmes tels que le contrôle et l'automatisation des chaînes de montage. Les réseaux de capteurs ont le potentiel de révolutionner la manière même de comprendre et de construire les systèmes physiques complexes. Les réseaux de capteurs peuvent se révéler très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement.

Parmi les domaines d'applications de ces réseaux de capteurs, nous citons : militaire, environnemental, sanitaire, domestique, commercial, etc. Voici des exemples détaillés d'applications potentielles dans ces différents domaines.

1.3.1. Applications militaires

Les réseaux de capteurs sans fil peuvent être intégrés dans le commandement militaire, le contrôle, les communications, le renseignement, la surveillance et les systèmes de reconnaissance et détection des cibles [2]. Le déploiement rapide, l'auto-organisation et la tolérance aux fautes caractéristiques des réseaux de capteurs en font une technique de détection très prometteuse pour les militaires. Bien que les réseaux de capteurs soient basés sur le déploiement dense des capteurs, caractérisé par leur faible coût, la destruction de certains nœuds

n'affecte pas une opération militaire autant que la destruction d'un capteur traditionnel, ce qui dote les réseaux de capteurs d'une grande utilité dans le champ de bataille. Une partie des applications militaires de réseaux de capteurs sont : surveillance des forces amies, des équipements et des munitions, la surveillance du champ de bataille ; reconnaissance des forces ennemies et le terrain, la détection de cible, l'évaluation des dommages de combat, et la détection des armes nucléaires, biologiques et chimiques.

Surveillance des forces amies, l'équipement et la munition : les dirigeants et les commandants peuvent constamment surveiller l'état des troupes amies, l'état de la disponibilité de l'équipement et des munitions dans un champ de bataille par l'utilisation de réseaux de capteurs [2]. Tous les soldats, véhicules, matériels et munitions peuvent être attachés avec de petits capteurs qui établissent des rapports de leurs statuts. Ces rapports sont rassemblés et envoyés aux chefs de troupes. Les données peuvent également être transmises aux niveaux supérieurs de la hiérarchie de commandement, tout en étant regroupées avec les données provenant d'autres unités à chaque niveau.

Surveillance du champ de bataille : terrains critiques, parcours, chemins et les ponts peuvent être rapidement couverts par des réseaux de capteurs ce qui permet de surveiller de près les activités des forces ennemies [2]. A tout moment, des nouveaux réseaux de capteurs peuvent être redéployés pour assurer la surveillance de l'évolution des opérations militaires.

Reconnaissance des forces ennemies et de terrain : les réseaux de capteurs peuvent être déployés dans des zones critiques, ce qui permet d'intercepter certains renseignements précieux, détaillés, en temps opportun sur les forces ennemies.

1.3.2. Applications environnementales

Les réseaux de capteurs sont utilisés dans certaines applications environnementales notamment pour localiser les mouvements d'oiseaux, de petits animaux et d'insectes; surveillance des conditions environnementales qui affectent les cultures et le bétail, détection chimique et biologique, agriculture de précision, biologique, la surveillance de la terre et de l'environnement marin, les sols et l'atmosphère, détection des incendies de forêt, la recherche météorologique ou géophysique, détection d'inondation et la pollution.

Détection d'incendie dans les forêts : les nœuds capteurs sont déployés de manière stratégique, ou aléatoire, et d'une façon dense dans une forêt où ils peuvent détecter l'origine exacte de l'incendie et alerté l'utilisateur final avant que le feu se propage et devient incontrôlable [2].

Détection des inondations : un exemple d'une détection d'inondation est le système ALERT [7] déployés aux États-Unis. Plusieurs types de capteurs déployés dans le système ALERT, tels que les capteurs de niveau d'eau et des capteurs météorologiques. Ces informations capturées alimentent une base centrale de données.

Agriculture de précision : les réseaux de capteurs offrent la possibilité de contrôler le niveau de pesticides dans l'eau potable, le niveau d'érosion des sols, et le niveau de pollution de l'air en temps réel [2].

1.3.3. Applications médicales

Certaines applications fournissent des interfaces pour les handicapés ; elles intègrent la surveillance des patients, les diagnostics, l'administration de médicaments dans les hôpitaux, la télésurveillance des données physiologiques de l'homme ainsi que le suivi et la surveillance des médecins et des patients dans un hôpital.

Le suivi et la surveillance des médecins et des patients dans un hôpital : un ensemble de capteurs minuscules pourraient être attachées au patient [2]. Chaque nœud capteur a une tâche spécifique (la détection du rythme cardiaque ou la pression artérielle). Les médecins peuvent aussi être équipés de capteurs, afin de pouvoir les localisés au sein de l'hôpital.

L'administration de médicaments dans les hôpitaux : des nœuds peuvent être attachés à des médicaments [2], ce qui pourrait réduire le taux d'une mauvaise prescription des médicaments aux patients. Ceci est possible grâce aux nœuds capteurs attachés aux patients qui permettent d'identifier leurs allergies et les médicaments nécessaires.

1.3.4. Applications domestiques

Domotique : le progrès technologique en matière de construction des capteurs et des actionneurs intelligents a permet leurs intégration dans les appareils, tels que les aspirateurs, fours à micro-ondes, réfrigérateurs et les magnétoscopes [2]. Ces nœuds capteurs à l'intérieur des appareils domestiques peuvent interagir les uns avec les autres et avec le réseau externe via Internet ou par

satellite. Ils permettent aux utilisateurs finaux de gérer des dispositifs d'accueil localement et à distance plus aisément.

L'environnement intelligent : les nœuds capteurs peuvent être intégrés dans les meubles et les appareils, et ils peuvent communiquer les uns avec les autres et avec le serveur de la pièce. Le serveur de la pièce peut également communiquer avec d'autres serveurs de pièces pour connaître les services qu'ils offrent, par exemple, l'impression, numérisation et la télécopie. Ces serveurs de pièces et les nœuds capteurs peuvent être intégrés avec les dispositifs embarqués existants pour devenir auto-organisés, autorégulés. Un autre exemple d'environnement intelligent est «*The Residential Laboratory*» à l'institut de technologie de Georgia [8].

1.3.5. Autres applications commerciales et industrielles

Les exemples d'application n'en manque pas dans ce domaine, on peu cité quelques uns : le contrôle des matériaux, gestion des stocks, contrôle de qualité des produits, contrôle de l'environnement dans les immeubles et les bureaux, contrôle de robots dans les environnements de fabrication automatique, les musées interactifs, diagnostic des machines.

Détection et surveillance des vols de voitures : les capteurs doivent être déployés pour détecter et identifier les menaces dans une région géographique, le rapport de ces menaces sera envoyé par Internet aux utilisateurs finaux pour analyse.

Gérer le contrôle d'inventaire : les capteurs peuvent être attachés à chaque élément dans un entrepôt. Les utilisateurs finaux peuvent trouver l'emplacement exact de l'article et de les dénombrer facilement. Ainsi, les gérants peuvent suivre et localiser les stocks a tout moment.

Conclusion

Dans ce chapitre, nous avons présenté le nœud capteur sans fil avec son architecture interne. Nous avons ensuite présenté les réseaux de capteurs et leurs propriétés tout en mettant en évidence leurs applications, ainsi que les défis auxquels ils sont confrontés.

A travers ce chapitre, nous avons montré l'apport des capteurs et en l'occurrence des réseaux de capteurs mais nous avons aussi montré du doigt les contraintes de ces environnements dont la limitation énergétique, une déficience en environnements hostiles présentant des obstacles divers ainsi que les difficultés de déploiement.

Dans le chapitre suivant, nous introduisons le concept du clustering dans les réseaux de capteurs sans fil, ses principes et ses objectifs. Ensuite, nous donnons les principales techniques de clustering proposés dans la littérature.

Chapitre 2

Techniques de clustering dans les réseaux de capteurs sans fil

Introduction

Les réseaux de capteurs sans fil sont fortement influencés par la limitation de leurs ressources physiques, et plus particulièrement l'énergie disponibles au niveau de chaque nœud. Actuellement, les principaux travaux de recherche sur ce type de réseau portent sur le prolongement de la durée de leur vie, via la conception et l'implémentation des protocoles de communications plus efficaces en termes de consommation énergétique. Divers mécanismes peuvent contribuer à l'optimisation de la consommation d'énergie lors des communications au sein d'un réseau de capteurs. Une structuration hiérarchique du réseau en clusters ou groupes de nœuds (clustering) s'est avérée parmi les approches les plus efficaces et les plus prometteuses pour atteindre cet objectif. Car elle peut permettre la réduction des messages de contrôle, l'augmentation de la réutilisabilité de la bande passante et l'équilibrage de charge des nœuds formant le réseau de capteurs. D'autre part, on trouve un certain nombre d'applications où les champs de captage sont naturellement constitués de nœuds plus ou moins proches formant des sous ensembles de nœuds ou groupes. Des solutions adaptées qui tirent avantages de cette organisation peuvent être très efficaces.

Le processus de clustering consiste à regrouper les nœuds proches géographiquement en groupes (*clusters*) et à élire localement un chef (*cluster-head*) selon un processus distribué d'élection. Cette hiérarchisation a pour objectifs une utilisation optimale des ressources, un routage plus efficace et un passage à l'échelle.

Dans ce chapitre nous définissons les concepts relatifs au mécanisme de clustering, ainsi que ses objectifs qui nous ont amené à choisir cette structure pour les réseaux de capteurs. Ensuite nous décrivons les principales techniques de clustering proposées dans la littérature pour les réseaux de capteurs avec nœuds stationnaires.

2.1. Définition

Un protocole de clustering est un ensemble d'algorithmes conçus pour la construction et la maintenance d'une structure hiérarchique logique. Cette structure consiste en un ensemble de groupes, nommés clusters, formés généralement des nœuds proches géographiquement comme illustré dans la Figure 5. Ces mêmes groupes peuvent à leur tour être partitionnés en sous groupes et ainsi de suite. Chaque cluster est identifié par un nœud particulier appelé cluster-head. Le choix du cluster-head se fait sur la base d'une métrique spécifique ou d'une combinaison de métriques telles que : l'identifiant, le degré, l'énergie et la mobilité etc.

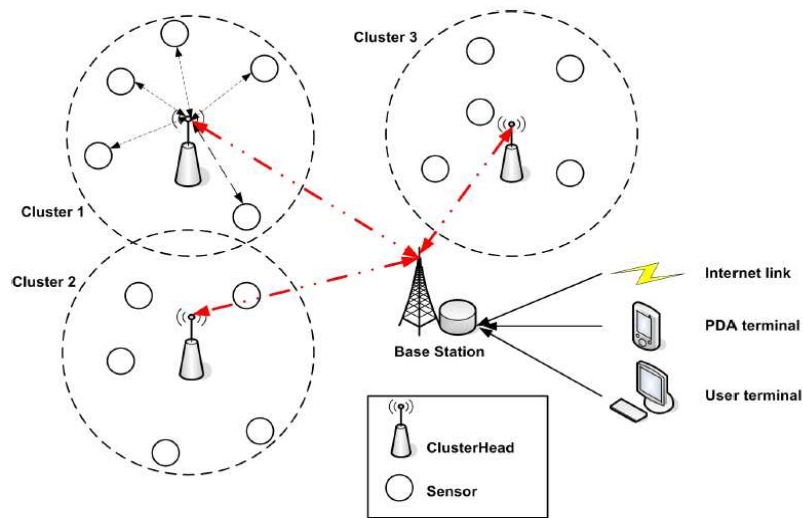


Figure 5. Exemple de clustering dans les WSNs.

Ce principe peut être formalisé comme suit [9]:

Soit un réseau de capteurs sans fil modélisé par un graphe connexe non orienté $G = (V, E)$ où les sommets V sont les nœuds et les arêtes E représentent les liens de communication. Le processus de clustering consiste au découpage virtuel de V en un ensemble de groupes proches géographiquement $\{V_1, V_2 \dots V_k\}$ tel que:

$$V = \bigcup_{i=1}^k V_i$$

Où chaque sous-ensemble V_i induit un sous-graphe connexe de G ou une composante connexe du graphe G . L'ensemble de ces composantes peuvent former un graphe réduit ou (graphe de composantes) $G' = (V', E')$ de G figure 6, où :

- les sommets $v'_i \in V'$ correspondent aux composantes connexes V_i de G qui représentent dans le réseau les clusters ou leurs cluster-heads.
- E' contient les arêtes (v'_i, v'_j) , tel qu'il existe dans le graphe G une arête d'un sommet $u_i \in V_i$ vers un sommet $u_j \in V_j$, ces arêtes représentent les liens entre les clusters.

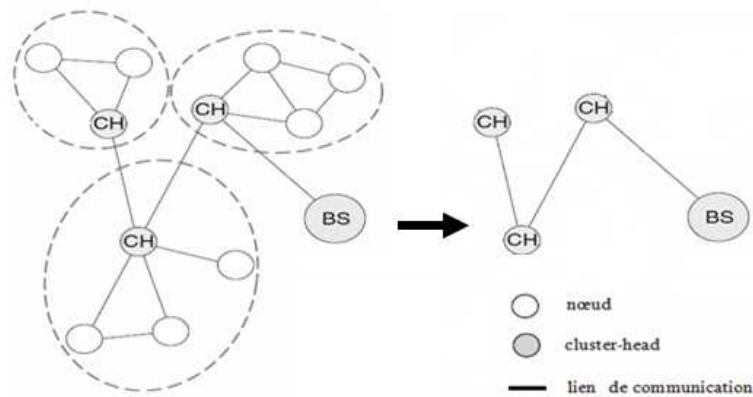


Figure 6. La représentation d'un WSN par un graphe

2.2. Principes de construction de clusters

Il existe plusieurs méthodes de construction de clusters [10, 11, 12, 13...]. Les plus répandues s'exécutent comme suit :

- Chaque nœud prend la décision pour être cluster-head ou non selon sa connaissance locale de la topologie ou un critère interne comme l'énergie résiduelle.
- Chaque cluster-head diffuse son statut dans son voisinage et invite ses voisins qui ne sont pas encore affiliés à d'autres clusters de le rejoindre.
- Une éventuelle maintenance en cas de défaillance des cluster-heads ou le changement de la structure des clusters.

2.2.1. Election des cluster-heads

Un cluster-head peut être sélectionné aléatoirement parmi l'ensemble des nœuds capteurs déployés, ou bien choisi suivant une métrique telle que le plus grand/petit ID dans son voisinage, le degré de connectivité, la puissance de transmission, l'énergie résiduelle ou la mobilité des nœuds ou bien un poids qui représente une combinaison de quelques métriques [14, 15...].

Il est essentiel d'attribuer le rôle cluster-head à un nouveau nœud régulièrement afin de ne pas charger certains nœuds de plus de tâches que les autres et dégrader leur niveau d'énergie. Il y a plusieurs possibilités pour la réélection d'un nouveau cluster-head. Il y a des techniques qui utilisent l'expiration d'un temporisateur pour déclencher le processus de clustering, ce qui permet la sélection régulière de nouveaux cluster-heads [10, 15, 16...].

D'autres techniques utilisent un paramètre dynamique (l'énergie résiduelle, la mobilité) pour déclencher l'algorithme de désignation d'un nouveau cluster-head. Par exemple un cluster-head pourrait déclencher le processus d'élection dans sa région locale si l'énergie de sa batterie va en dessous d'un seuil prédéfini.

Il est évident que la réélection plus fréquente des cluster-heads ait comme conséquences un "overhead" plus important et parfois une surcharge et interruption du réseau, tandis que la réélection moins fréquente peut causer la défaillance de quelques nœuds plus rapidement que d'autres. L'étude de ce compromis est essentielle pour atteindre une durée de vie de réseau optimale.

2.2.2. Communication intra-cluster et inter-cluster

Un objectif important de n'importe quelle technique de clustering est le maintien de la connectivité dans le réseau. Les nœuds membres d'un cluster communiquent avec leurs cluster-heads, et à leur tour les cluster-heads communiquent entre eux ou directement avec la station de base pour livrer les informations prélevées par les nœuds capteurs. Pour les communications intra-cluster, un membre du cluster communique avec son cluster-head directement comme dans [17, 18, 19...] ou bien en multi-sauts comme dans [20]. Pour la communication inter-cluster, deux approches ont été adoptées pour maintenir la connectivité :

- Les nœuds sur les frontières du cluster sont utilisés comme des passerelles pour retransmettre les données entre les cluster-heads (Figure 7), cette approche est adéquate dans les réseaux qui utilisent une puissance de transmission fixe [11, 12...]. La densité du réseau doit être suffisamment haute pour assurer qu'assez de passerelles soient présentes au niveau des zones d'intersection entre les clusters.

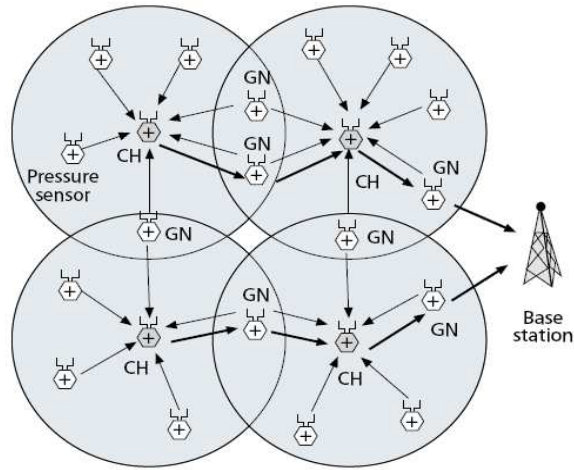


Figure 7. Communication inter cluster via des nœuds passerelles.

- Dans une autre approche le recouvrement des cluster-heads constitue l'infrastructure de routage où les communications inter-cluster sont assurées via les cluster-heads seulement (Figure 8). Un avantage de cette approche est qu'elle permet à tout nœud non cluster-head de passer en mode sommeil (*sleep*) dès qu'il n'est pas émetteur ou récepteur.

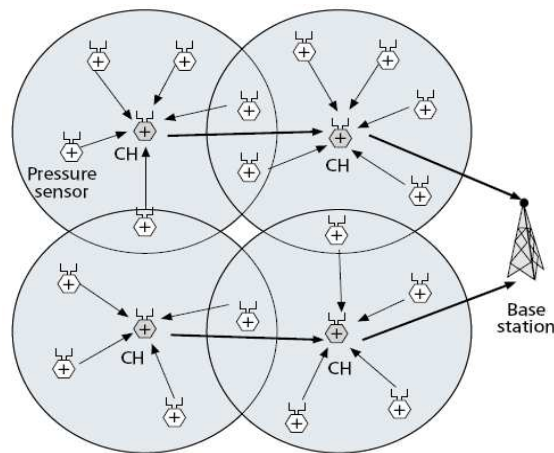


Figure 8. Communication inter cluster via des cluster-heads

2.2.3. Maintenance des clusters

Dans le but de s'adapter aux changements de la topologie du réseau, une mise à jour des clusters est dynamiquement réalisée dans le cas où un cluster-head ou un membre migre d'un cluster à un autre. D'autre part, la défaillance d'un cluster-head à cause de la consommation intégrale de son énergie nécessite la réélection d'un nouveau cluster-head.

Les mécanismes de maintenance peuvent être très onéreux parce que la défaillance d'un seul nœud ou son déplacement peut déclencher la restructuration en clusters de tout le réseau. On peut trouver deux principales approches de maintenance : une maintenance périodique en cas de mobilité importante (reconstruction des clusters globalement) ou une maintenance dynamique au besoin (à la sortie ou l'intégration d'un nœud au groupe).

2.3. Objectifs de clustering

Les algorithmes de clustering cités dans la littérature, varient dans leurs objectifs. Souvent l'objectif de clustering est d'offrir une couche de service sous-jacente aux applications pour faciliter la mise en œuvre de ces dernières. Par exemple, si l'application exige la défaillance aux pannes des nœuds, alors l'énergie résiduelle des nœuds et leur densité sont considérées comme critères pour la sélection des cluster-head et la formation des clusters.

Les principaux objectifs de clustering peuvent être cités par les points suivants:

2.3.1. Équilibrage de la charge (*Load balancing*)

La répartition des nœuds capteurs en clusters est habituellement un objectif pour les configurations des réseaux de capteurs sans fil où les cluster-heads exécutent des traitements spécifiques sur les données en plus des tâches de gestion intra et inter-cluster. Les fonctions des cluster-heads doivent intégrer, obligatoirement, l'équilibrage de charge entre les capteurs afin d'atteindre des performances de durée de vie du réseau.

L'équilibrage de charge est un besoin indispensable dans les réseaux de capteurs sans fil où les cluster-heads sont sélectionnés parmi les capteurs disponibles. Généralement, la charge de la tâche du cluster-head croît avec la taille du cluster administré. Et, donc la taille est un des critères important à considérer lors d'un équilibrage de charge. La recherche d'une répartition des nœuds en clusters de tailles proches joue un rôle important dans l'extension de la durée de vie du réseau puisqu'il évite l'épuisement rapide d'énergie d'un sous-ensemble de cluster-heads [21].

2.3.2. Connectivité augmentée et latence réduite

La connectivité dans les réseaux de capteurs sans fil peut se limiter à assurer la disponibilité d'un chemin de chaque cluster-head à la station de base d'une part et la connectivité des nœuds membres à leurs cluster-heads d'autre part. Quand la latence de données est un souci,

les communications intra-cluster devient un objectif et une contrainte de conception où la latence est habituellement assurée en mettant un nombre maximal de sauts sur un chemin de communication [21].

2.3.3. Durée de vie étendue du réseau

Puisque les nœuds capteurs sont limités en énergie, la durée de vie du réseau est un souci majeur, particulièrement pour les applications des réseaux de capteurs déployés dans des environnements hostiles.

Le clustering fournit des avantages distincts en ce qui concerne l'économie d'énergie. Le contrôle localisé des nœuds au sein d'un cluster permet la réduction du volume des transmissions entre ces nœuds et ainsi la réduction de l'énergie. D'autre part, la coordination fournie par les cluster-heads permet aux nœuds membres de passer en mode sommeil pendant une longue durée, permettant ainsi une économie significative d'énergie. Le clustering permet aussi aux réseaux de capteurs de bénéficier de beaucoup d'avantages qui ont une incidence directe ou indirecte sur l'efficacité énergétique comme la gestion de la bande passante, l'équilibrage de charge [21].

2.3.4. Passage à l'échelle (*scalability*)

Les algorithmes de clustering dans les réseaux de capteurs améliorent la scalability du réseau en traitant deux problèmes importants quant à la taille et à la mobilité du réseau. Ils offrent une abstraction du réseau où le nombre d'entités (nœuds) est réduit et où la mobilité est davantage contrôlée au sein d'un cluster car elle est souvent corrélée à la mobilité globale du cluster [22].

a) Restriction en communications

L'objectif principal du clustering est d'avoir une restriction sur les communications et l'échange de données au sein du réseau, cela est assuré au niveau des différentes opérations du réseau comme suit :

- **Medium Access Control (MAC) :** en utilisant le clustering l'accès au media de transmission peut être contrôlé et la bande passante est allouée séparément pour chaque cluster. Cela réduit les interactions inter-cluster et minimise la redondance des messages changées.
- **Routage :** la taille des tables de routage va être réduit en sauvegardant que les routes vers les cluster-heads et non pas vers tous les nœuds.

- **Inondation (*flooding*)** : le cout de flooding est réduit en diminuant le nombre de nœuds qui diffusent (*broadcast*) les messages seulement aux cluster-heads.
- **Collecte de données** : les données collectées seront agrégées au niveau des cluster-heads et envoyées vers la station de base ce qui diminue le nombre de messages envoyé à la station de base.
- **Découverte de service** : les cluster-heads maintient un répertoire de services fournit par les nœuds membres, ainsi les messages de découverte de services sont envoyées qu'aux cluster-heads.

b) Offrir une topologie plus stable

Le clustering peut être aussi un moyen pour rendre une topologie dynamique plus stable, cela est assuré dans les opérations du réseau suivantes :

- **Routage** : Les informations de routage sont mises à jour seulement pour le routage intra-cluster. Le routage inter-cluster est réalisé en cachant le détail de la topologie interne des clusters pour les nœuds externes. De ce fait on limite les réactions de grandes envergures causé par les changements de la topologie.
- **Traitements collaboratifs** : la création des clusters à base des nœuds qui se déplacent ensembles permet le traitement collaboratif des taches dans un cluster (*Body sensors*) [22].

2.4. Classification des techniques de clustering

Dans cette partie nous allons énumérer l'ensemble des attributs qui caractérisent le processus de clustering et qui peuvent être utilisés pour classifier et différencier les techniques de clustering proposées pour les réseaux de capteurs sans fil [21].

2.4.1. Propriété des clusters

Généralement les techniques de clustering ont pour objectif d'assurer certaines caractéristiques spécifiques aux clusters générés. Ces caractéristiques peuvent être liées à la structure interne d'un cluster ou à l'architecture d'interconnexion de ces clusters.

- **Nombre de clusters** : le processus de clustering améliore la scalability du réseau en fournissant une couche sous-jacente aux applications, qui leur permet de voir le réseau plus petit en taille et plus stable. Néanmoins, la taille et le nombre de clusters générés

influent sur les performances de la technique de clustering, ces performances dépendent généralement de l'application pour laquelle on conçoit cette technique.

- **Stabilité** : Lorsque le nombre de clusters varie et les nœuds membres évoluent et changent de clusters dans le temps, la technique de clustering est dite adaptative. Et elle est considérée fixe ou statique lorsque les nœuds ne commutent pas entre les clusters et le nombre de clusters reste le même tout au long de la durée de vie de réseau [10]
- **Communication intra-cluster** : Plusieurs techniques de clustering sont basées sur la transmission directe à un saut entre les capteurs et leurs cluster-heads. Cependant, la connectivité des capteurs aux cluster-heads en multi-saut est parfois exigée ; particulièrement quand la portée de transmission des capteurs est limitée et/ou le nombre des cluster-heads est limité.
- **Connectivité Inter-cluster**: Lorsque le cluster-head a des capacités de transmission réduite en longueur de portée, la connectivité directe des cluster-heads à la station de base risque de ne pas être assurée. Dans ce cas, les techniques de clustering doivent assurer les moyens d'établir une route inter- cluster-head entre les cluster-head et la station de base. Cependant, plusieurs travaux publiés supposent que les cluster-heads peuvent atteindre directement la station de base [10, 16, 23...].

2.4.2. Caractéristiques des cluster-heads

Les capacités et les caractéristiques des cluster-heads ont une grande influence sur les techniques de clustering, parmi ces caractéristiques on cite:

- **Mobilité** : Lorsqu'un cluster-head se déplace, les capteurs membres de ce cluster risquent de perdre la connexion avec leur cluster-head, par conséquent les clusters devraient être continuellement mis à jour. D'autre part, les cluster-heads stationnaires tendent généralement à former des clusters stables et facilitent la gestion du réseau au niveau de l'intra- et de l'inter-cluster.
- **Type de nœud** : Les cluster-heads sont généralement des nœuds ordinaires, mais dans quelques approches de clustering comme [24], les cluster-heads sont avantagés par des ressources de traitement et d'énergie supplémentaires (*super sensor*).
- **Rôle** : Un cluster-head peut agir en tant que relais pour le trafic produit par les capteurs dans le cluster vers la station de base ou exécuter des tâches supplémentaires comme l'agrégation et la fusion des données prélevées par les capteurs membres.

2.4.3. Processus de clustering

Les caractéristiques du processus de clustering varient et dépendent de : la méthodologie adoptée, les objectifs à satisfaire, les critères de sélection de cluster-heads et la complexité des protocoles que devra supporter le réseau. Parmi les attributs qui influent sur ce processus nous citons:

- **Méthodologie:** L'exécution d'un algorithme de clustering peut être effectuée d'une façon centralisée par la station de base, ou distribuée sur les différents nœuds du réseau. Les approches centralisées exigent des informations globales sur la topologie de réseau comme l'état des connexions et les positions des nœuds capteurs. Dans les approches distribuées, si un nœud décide de joindre un cluster ou devenir un cluster-head, il se base seulement sur les informations locales obtenues à partir de ses voisins.
- **Objectifs du clustering:** Comme on a vu précédemment, les techniques de clustering dépendent des objectifs fixés comme la scalability, l'équilibrage de charge, la connectivité du réseau et l'augmentation de la durée de vie du réseau.
- **Métriques de choix cluster head:** La désignation d'un capteur en tant que cluster-head ou son adhésion à un cluster est généralement basée sur les métriques suivantes:
 - o **Temps :** Un nœud peut devenir cluster-head dans son voisinage en se basant sur le temps, c-à-dire qu'il est le premier qui se déclare comme cluster-head [25].
 - o **Probabilité :** Un nœud peut devenir cluster-head selon une mesure probabiliste. La probabilité est peut être définie comme le nombre désiré de cluster-heads. Cette probabilité peut dépendre du nombre de nœuds dans le réseau, de l'énergie globale, de l'énergie résiduelle d'un nœud, de la taille des clusters et du nombre de fois où les nœuds ont été élus comme cluster-heads [10, 23...].
 - o **Poids :** Un poids est un nombre lié à l'application, calculé par chaque nœud dans le réseau. Le poids peut dépendre de plusieurs paramètres, tels que le degré d'un nœud (nombre de voisins), la vitesse de déplacement, l'énergie restante et les capacités d'un nœud. Un nœud peut devenir cluster-head s'il a le poids le plus élevé parmi un groupe de nœuds (ou plus petit). Contrairement à la métrique de probabilité, la métrique de poids est déterministe.

- o **Sémantique :** Les propriétés sémantiques se rapportent aux relations entre deux nœuds ou plusieurs dans un groupe. C'est-à-dire au rôle que joue un nœud par rapport à un autre tel que, par exemple, voisin ou source de données. Les propriétés sémantiques incluent les distances entre les nœuds, la disponibilité des chemins entre les nœuds, la mobilité relative, l'information de localisation et le type d'événement détecté.

Le processus de décision peut dépendre de plusieurs types de métriques citées ci-dessus. Par exemple, dans [10] le choix du cluster-head est fait de façon probabiliste, mais les nœuds membres choisissent leurs cluster-heads en se basant sur une propriété sémantique; la distance minimale aux cluster-heads voisins.

- **Complexité de l'algorithme**

La complexité d'un algorithme de clustering est indispensable pour estimer la latence et l'overhead générés dans la construction et la maintenance des clusters. Pour évaluer la complexité en temps, on considère que l'algorithme commence à partir d'un état stable. Le temps de réaction nécessaire à l'algorithme après le déclenchement d'un événement simple (par exemple une liaison détectée ou perdue) pour restructurer les clusters est défini comme le temps de convergence [22]. La complexité en message est défini comme l'effort de transmission nécessaire pour la formation des clusters et leurs maintenance ou mise à jour [26].

2.5. Techniques de clustering pour les réseaux de capteurs statiques

Afin d'avoir une topologie plus stable et plus favorable au passage à l'échelle pour les réseaux de capteurs sans fil, de nombreuses solutions basées sur le clustering ont été proposées dans la littérature. En outre, ces protocoles permettent une optimisation de l'utilisation de la bande passante ainsi que la réduction de la consommation d'énergie. Dans cette section, nous présentons les principaux protocoles de clustering proposés pour les réseaux de capteurs statiques.

2.5.1. LEACH (*Low Energy Adaptive Clustering Hierarchy*)

Dans [10], Heinzelman et al. ont proposé un algorithme de clustering distribué appelé LEACH pour le routage dans les réseaux de capteurs homogènes. LEACH choisit aléatoirement les cluster-heads et attribue ce rôle aux différents nœuds selon la politique de gestion Round-

Robin (tour à tour) pour garantir une dissipation équitable de l'énergie entre les nœuds. Dans le but de réduire la quantité d'informations transmise à la station de base, les cluster-heads agrègent les données capturées par les nœuds membres qui appartiennent à leur propre cluster, et envoient l'agrégation des informations récoltées à la station de base.

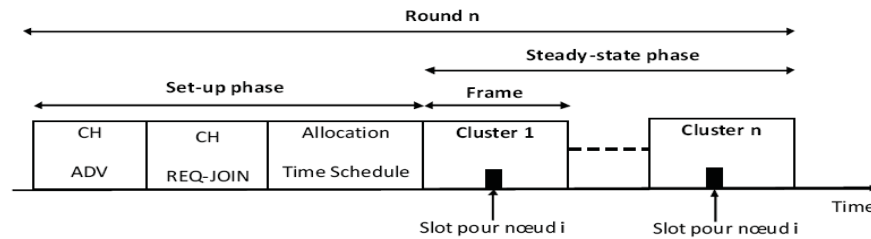


Figure 9. Formation de clusters par LEACH.

LEACH est exécuté en deux phases : la phase de construction des clusters (*set-up*) et la phase d'état stationnaire (*steady-state*) comme illustré dans la figure 9. La phase *set-up* est subdivisée en trois sous phases : la phase d'annonce, la phase d'organisation de groupes et la phase d'ordonnancement. Dans ces sous phases, les cluster-heads sont sélectionnés et les clusters sont formés, et dans la seconde phase, le transfert de données vers la station de base est effectué.

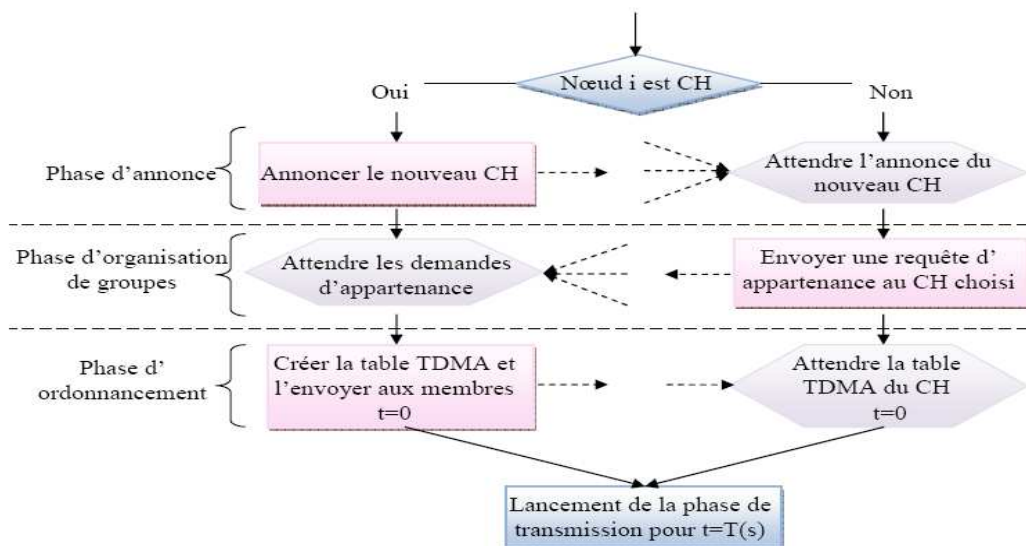


Figure 10. Opérations de l'étape d'initialisation de LEACH.

a) Phase d'annonce

Durant cette phase, le processus d'élection des cluster-heads est déclenché pour choisir les futurs cluster-heads. Ainsi, une fraction p prédéterminée des nœuds s'élisent comme cluster-heads selon le schéma d'exécution suivant :

- Durant une période r , un nœud u choisit un nombre aléatoire r_u dont la valeur est comprise entre 0 et 1 ($0 < r_u < 1$).
 - o **Si** r_u est inférieur à une valeur seuil $T(u)$ alors le nœud u deviendra cluster-head durant la période courante.
 - o **Sinon** le nœud u devrait rejoindre le cluster-head le plus proche dans son voisinage.
- La valeur seuil $T(u)$ est calculée comme suit :

$$T(u) = \begin{cases} \frac{p}{1 - p \times \left(r \times \text{mod} \left(\frac{1}{p} \right) \right)}, & \text{si } u \in G \\ 0, & \text{sinon} \end{cases}$$

Où :

- p : le pourcentage de nœuds cluster-heads ($p = 5\%$).
- r : la période courante.
- G : l'ensemble des nœuds qui n'étaient pas cluster-heads durant les périodes précédentes.

b) Phase d'organisation de groupes

Après qu'un nœud soit élu cluster-head, il doit informer les autres nœuds non cluster-head de son nouveau rang dans le round courant. Pour cela, un message d'avertissement (*ADV_CH*) contenant l'identificateur du cluster-head est diffusé à tous les nœuds non cluster-head en utilisant le protocole CSMA (*Carrier Sense Multiple Access*) pour éviter les collisions entre les cluster-heads. La diffusion permet de s'assurer que tous les nœuds non cluster-heads ont reçu le message. Par ailleurs, elle permet de garantir que les nœuds qui appartiennent au cluster-head requièrent le minimum d'énergie pour la communication. La décision d'adhésion au cluster est basée sur l'amplitude du signal reçu; le cluster-head ayant le signal le plus fort (le plus proche) sera choisi. En cas d'égalité des signaux, les nœuds non cluster-heads choisissent aléatoirement leurs cluster-heads.

Chaque membre informe son cluster-head de sa décision en lui envoyant un message de demande d'adhésion (*REQ_JOIN*).

c) Phase d'ordonnancement

Après la réception de tous les messages de demandes d'adhésion, chaque cluster-head, en se basant sur l'approche TDMA (*Time Division Multiplexed Access*) établit et diffuse un ordonnancement affectant à chaque membre du cluster un slot de temps durant lequel il peut communiquer les données captées au cluster-head. Cette phase inclue également la distribution des codes CDMA (*Code Division Multiplexed Access*) utilisés pour éviter les interférences possibles entre les nœuds se trouvant dans des clusters adjacents.

d) Phase de transmission de données

Une fois l'ordonnancement est connu par tous les nœuds du même cluster, chacun d'eux utilise le slot temps qui lui a été consacré pour transmettre au cluster-head les données captées. Le cluster-head exécute, par la suite, des algorithmes de fusion et d'agrégation des données reçues afin de minimiser la quantité d'information transmise à la station de base.

Critiques : Bien que LEACH présente beaucoup d'avantage en termes d'augmentation de la durée de vie du réseau, il présente certaines limitations.

- LEACH suppose que tous les nœuds peuvent transmettre des données avec une grande puissance pour atteindre la station de base. Par conséquent, LEACH n'est pas intéressant pour les réseaux déployés dans de vastes régions.
- LEACH choisit aléatoirement la liste des cluster-heads et il ne pose aucune contrainte sur leur distribution ainsi que sur leur niveau d'énergie. Par conséquent, les cluster-heads peuvent se concentrer dans une région restreinte; le champ de captage est alors non couvert.
- Dans LEACH, la tâche de collecte et d'agrégation des données est centralisée et elle est exécutée périodiquement. Or, dans certains cas, la transmission périodique des données pourrait ne pas être nécessaire et par suite elle pourrait consommer inutilement l'énergie limitée des capteurs.

LEACH-C : Une autre variante de LEACH appelée LEACH-C [27] a été conçue pour améliorer les performances de LEACH. Cette variante utilise une architecture centralisée pour choisir les cluster-heads tout en impliquant la station de base et l'information de localisation des

capteurs. La transmission de données à la station de base se fait comme dans LEACH. Cependant, cette variante augmente considérablement l'overhead du réseau puisque tous les capteurs doivent envoyer leurs informations de localisation à la station de base en même temps pendant chaque phase d'élection de cluster-heads. Ceci fait que cette architecture centralisée ne supporte pas le passage à l'échelle et n'améliore pas la durée de vie du réseau.

2.5.2. HEED (*Hybrid Energy-Efficient Distributed Clustering*)

Younis et Fahmy ont présenté le protocole hiérarchique HEED [28]. Ce dernier est une variante des protocoles de clustering qui adopte une stratégie périodique pour choisir les cluster-heads en se basant sur l'énergie résiduelle et sur un coût de communication telle que la proximité d'un nœud à ses voisins ou son degré (généralement utilisé comme deuxième paramètre de sélection).

Dans le processus de clustering utilisé par HEED, chaque nœud prend sa décision selon ses informations locales. Par ailleurs, les deux paramètres mentionnés précédemment sont utilisés pour la sélection d'un ensemble de cluster-heads initial. Cependant, un nœud avec une grande quantité d'énergie résiduelle a plus de chance de s'élire cluster-head. Le second paramètre sert à résoudre les conflits; lorsqu'un nœud se trouve dans la portée de plusieurs cluster-heads. Dans ce cas, si le niveau de puissance utilisé pour la communication intra-cluster est fixe pour tous les nœuds, alors le coût de communication est proportionnel au degré du nœud. Cela signifie qu'un nœud rejoint le cluster-head ayant le degré minimum pour distribuer la charge des clusters head. Comme, il peut rejoindre le cluster head ayant le degré maximal pour créer des clusters denses. Si des portées de transmission variables sont permises au sein d'un même cluster, chaque nœud a la possibilité de choisir le niveau de puissance appropriée pour atteindre son cluster-head. D'où le paramètre AMRP donné par :

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

Où

$MinPwr_i$: La puissance minimale

M : Le nombre de nœud.

$AMRP$ représente la puissance moyenne pour la communication intra-clusters. Il est utilisé comme troisième paramètre dans la fonction du coût de communication.

La conception de ce protocole a été faite de façon qu'il se termine en un nombre limité d'itérations $N_{iter} = O(1)$. Durant chaque période $T_{cp} + T_n$ un nouveau cluster-head est choisi où : T_{cp} est l'intervalle de temps du processus de clustering, et T_n est le temps qui sépare la fin d'un intervalle T_{cp} et le début de la période suivante. Mais, avant qu'un nœud commence l'exécution de HEED, il calcule sa probabilité pour devenir cluster-head comme suit:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}}$$

Où :

C_{prob} : Probabilité qui détermine le nombre de cluster-heads.

$E_{residual}$: Énergie résiduelle estimée sur le nœud.

E_{max} : Énergie maximale (correspond à une batterie entièrement chargée).

HEED présente plusieurs arguments en sa faveur. C'est un protocole non centralisé convergeant; le clustering se termine dans $O(1)$ itérations. Il prolonge la durée de vie du réseau en distribuant l'énergie de la communication et le nombre de cluster-heads d'une façon uniforme, il produit des clusters compacts tout en minimisant la charge des messages de contrôle. Son processus d'itérations est indépendant de la topologie et de la taille du réseau. En outre, il ne fait aucune supposition sur la distribution ou la densité des nœuds, ainsi que sur leurs capacités. Cependant, vu la stratégie adoptée pour la communication entre les cluster-heads est la station de base qui se fait via un seul saut, HEED ne permet pas le passage à l'échelle. D'autre part, les clusters générés avec HEED ne sont pas vraiment équilibrés en taille.

2.5.3. PEGASIS (*Power-Efficient Gathering in Sensor Information Systems*)

Dans [29], Lindsey et Raghavendra ont proposé PEGASIS. L'idée principale de PEGASIS est de former une chaîne entre les nœuds de sorte que chaque nœud reçoive et communique avec un voisin proche. Les données collectées sont transmises d'un nœud à un autre qui les rassemble jusqu'à ce qu'elles arrivent à un nœud particulier qui les transmet à la station de base. Les nœuds qui transmettent les données à la station de base, sont choisis tour à tour selon un round-robin dans le but est de réduire l'énergie moyenne dépensée par un nœud durant une période (round). Ce protocole considère que le réseau de capteurs répond aux critères et hypothèses suivantes :

- La station de base est fixée à une distance constante des nœuds, et chaque nœud a la possibilité de transmettre les données captées à cette base directement ou à travers des nœuds intermédiaires.
- Les nœuds disposent l'information de localisation de tous les autres nœuds du réseau.
- Les nœuds sont homogènes avec des énergies uniformes.
- Les nœuds sont immobiles.

Selon les auteurs de ce protocole, initialement les nœuds capteurs sont déployés aléatoirement dans le champ de captage, ensuite, les nœuds sont organisés pour construire une chaîne. Cette chaîne contient un nœud appelé chef (*leader*), son rôle est la transmission de données vers la station de base. La position du nœud leader change aléatoirement. A chaque round le processus de création de la chaîne commence par le nœud le plus éloigné de la station de base. Si un nœud dans la chaîne tombe en panne la chaîne est reconstruite dynamiquement.

Le principe de collection des données durant chaque round est résumé comme suit : chaque nœud reçoit les données de l'un de ses deux voisins dans la chaîne, puis, il les fusionne avec ses propres données captées, ensuite, il transmet les données fusionnées vers l'autre voisin dans la chaîne. Quand les données atteignent le nœud leader, ce dernier fusionne les données reçus avec ses propres données, puis il les transmet vers la station de base.

Dans un round donné, un jeton de contrôle est initié par le nœud leader pour lancer le processus de transmission de données de bords de la chaîne vers la station de base. La taille du jeton est très petite, par conséquent, son coût n'est pas important. Par exemple, dans la figure 11 : le nœud C2 est le leader, il va passer le jeton le long de la chaîne vers le nœud C0. Le nœud C0, par la suite va envoyer ses données vers le nœud C2 en passant par le nœud intermédiaire C1. Ce dernier fusionne les données reçues avec ses propres données. Quand C2 reçoit les données de C1, il va passer le jeton vers C4, ensuite C4 va envoyer ses données vers C2 en passant par le nœud C3. Quand C2 reçoit les données de ses deux voisins, il les fusionne avec ses données. Finalement, le nœud C2 transmet les données fusionnées vers la station de base.

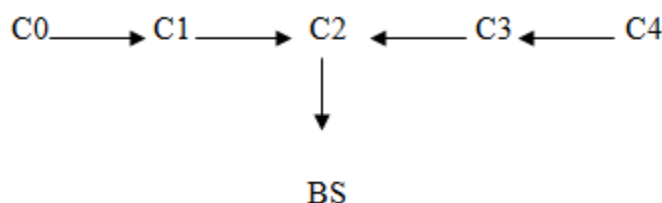


Figure 11. Collecte de données dans PEGASIS.

Les résultats de simulation ont montré que PEGASIS peut prolonger de deux à trois fois la durée de vie d'un réseau de capteurs relativement à LEACH en fonction du critère choisi pour évaluer la durée de vie d'un réseau. Ce critère est défini comme étant le délai au bout duquel un certain pourcentage des nœuds ont épuisé leur énergie. Ceci a été vérifié pour les cas où 1%, 20%, 50% et 100% des nœuds ont épuisé leurs énergies [29]. Un tel gain de performance est réalisé par l'élimination de l'overhead causé par le processus de formation de clusters dans LEACH, et par la réduction du nombre de transmissions et de réceptions par utilisation de l'agrégation de données. Bien que l'overhead du clustering soit évité, PEGASIS exige toujours une connaissance locale de la topologie, chaque nœud devrait connaître le niveau d'énergie de ses voisins avant de relayer ses données. Cependant, une telle connaissance de la topologie pourrait causer un overhead important.

PEGASIS suppose que tout nœud est capable de communiquer directement avec la station de base. Or, cette supposition est loin de la réalité car les capteurs communiquent généralement en mode multi-sauts pour atteindre la station de base. D'autre part, PEGASIS requiert que chaque nœud maintienne une table contenant les localisations de tous les autres nœuds dans le réseau, d'où une connaissance de la topologie complète.

2.5.4. TEEN (*Threshold-sensitive Energy Efficient sensor Network protocol*)

Manjeshwar et Agrawal [30] ont proposé une technique de clustering appelée TEEN pour les applications critiques où le changement de certains paramètres peut être brusque. L'architecture du réseau est basée sur un groupement hiérarchique à plusieurs niveaux où les nœuds les plus proches forment des clusters, puis ce processus passe au deuxième niveau jusqu'à ce que la station de base soit atteinte. Après la formation des clusters, chaque cluster-head transmet à ses membres deux seuils : un seuil Hard H_T (*hard threshold*), qui représente la valeur seuil du paramètre surveillé et un seuil Soft S_T (*soft threshold*) représentant un seuil de variation de la valeur du paramètre surveillé. L'occurrence de ce seuil minimale S_T de variation indique au nœud qui la détecte la nécessité de la signaler à la station de base en transmettant un message d'alerte. Par conséquent, le seuil Soft limite le nombre de transmissions, puisqu'il ne permet pas la transmission s'il y a peu ou pas de variation de la valeur du paramètre contrôlé. Au début, les nœuds écoutent le médium continûment et lorsque la valeur captée du paramètre surveillé dépasse le seuil Hard, le nœud transmet les données. La valeur captée est stockée dans une variable interne appelée SV. D'autre part, les nœuds ne transmettent des données que si la valeur

courante du paramètre contrôlé est supérieure au seuil hard H_T ou diffère du SV d'une quantité égale ou plus grande que la valeur du seuil Soft S_T . Puisque la transmission d'un message consomme plus d'énergie que la détection des données, la consommation d'énergie dans TEEN est moins important que dans les protocoles proactifs ou ceux qui transmettent des données périodiquement tels que LEACH. Cependant, l'inconvénient principal de ce protocole est que, si les seuils H_T et S_T ne sont pas reçus, les nœuds ne communiqueront jamais les données. Aucune donnée ne sera transmise à l'utilisateur, et ainsi la station de base ne peut pas connaître les nœuds qui ont épuisés leur énergie.

Pour remédier à ces limitations, les auteurs de [31] ont proposé une extension de TEEN appelée APTEEN (*Adaptive Threshold-sensitive Energy Efficient sensor Network protocol*). APTEEN est un protocole hybride qui change la périodicité et les valeurs seuils utilisées dans TEEN selon les besoins de l'utilisateur et le type d'application. Dans APTEEN, les cluster-heads transmettent à leurs membres les paramètres suivants :

- l'ensemble des paramètres physiques qui intéressent l'utilisateur.
- Les seuils : seuil Hard H_T et seuil Soft S_T .
- Un Schedule TDMA permettant d'assigner à chaque nœud slot de temps.
- Une période de temps (CT) : temps maximum entre deux transmissions successives d'un nœud.

Dans APTEEN, les nœuds surveillent en continu l'environnement. Ainsi, les nœuds qui détectent une valeur du paramètre qui dépasse le seuil H_T , transmettent leurs données. Ensuite, lorsque ce nœud détecte une autre fois une valeur qui dépasse H_T , il ne va transmettre ses données au cluster-head que si la valeur de ce paramètre change d'une quantité égale ou plus supérieure à S_T . Si un nœud ne transmet pas de données pendant une période de temps CT, il devrait faire une capture de données et les transmettre.

APTEEN offre une grande flexibilité qui permet à l'utilisateur de choisir l'intervalle de temps CT et les valeurs seuils H_T et S_T pour que la consommation d'énergie soit contrôlée par la variation de ces paramètres. Cependant, APTEEN nécessite une complexité supplémentaire pour implémenter les fonctions de seuils et de périodes de temps CT. Ainsi, l'overhead et la complexité associée à la formation des clusters à plusieurs niveaux par TEEN et APTEEN sont assez élevés.

2.5.5. EECM (*An Energy Efficient Clustering Method for Wireless Sensor Networks*)

Dans [15], les auteurs proposent une méthode de clustering des nœuds de capteurs afin de préserver leurs énergies, cette méthode combine l'énergie résiduelle des nœuds et le nombre de voisins autour de chaque nœud pour sélectionner les cluster-heads. Pour cela chaque nœud calcule son poids selon la formule suivante :

$$P = \frac{E_r}{E_i} + (\alpha \times \frac{N_n}{N_i}) \quad \text{avec } 0 < \alpha < 1$$

Où E_i est la moyenne des énergies résiduelles des nœuds voisins et E_r est l'énergie résiduelle du nœud lui-même. N_i est la moyenne du nombre de nœuds voisins des nœuds voisins et N_n est le nombre de nœuds voisins du nœud lui-même. α est un facteur de réglage d'accord qui contrôle la possibilité de sélection de cluster-head basée sur le nombre de nœuds voisins. Si N_n est plus grand que N_i le nœud sera de nouveau choisi comme cluster-head dans l'itération suivante. Pour empêcher la répétition de cette situation, la valeur de α est diminuée chaque fois que le nœud est choisi comme un cluster-head.

Sélection des cluster-heads et construction des clusters

À la fin de chaque itération, chaque nœud envoie un message *HELLO*, qui contient son énergie résiduelle et le nombre de nœuds voisins. Avec ces informations, un nœud calcule les poids de ses voisins. Si son poids est plus grand que ceux des voisins, il s'élit comme un cluster-head et envoie un message d'avertissement (*ADV*) à ses nœuds voisins. Quand un nœud voisin a la même valeur de poids que sa propre valeur, le nœud avec l'identification la plus petite est élue comme cluster-head. En outre, si un nœud n'a aucun nœud voisin autour de lui, le nœud devient cluster-head.

À la réception du message *ADV*, chaque nœud voisin sait qu'un nœud autour de lui a été élu comme cluster-head. Et il envoie un message (*REQ_JOIN*) au cluster-head pour appartenir au cluster.

Dans les simulations les auteurs ont mesuré la durée de vie de réseau (en termes de durée de temps avant la mort du premier nœud). Ils ont montré qu'EECM améliore la durée de vie du réseau par rapport au protocole LEACH d'un facteur qui peut atteindre jusqu'à 50%.

2.5.6. PC (*Passive Clustering*)

Les auteurs dans [32] proposent un protocole de clustering réactif, il est totalement distribué et il n'a besoin d'aucun mécanisme de synchronisation ni de phase d'initialisation. Contrairement aux protocoles proactifs basés sur l'échange périodique d'informations, le *passive clustering* a été conçu dans le but de n'utiliser aucun message de contrôle pour construire la topologie en clusters. Pour se faire, le protocole interagit avec la couche MAC de la façon suivant :

- A chaque émission ou réception d'une trame, le protocole est exécuté. Ainsi une exécution périodique est évitée.
- A chaque trame émise sera ajouté un champ contenant l'état du nœud émetteur : cluster-head, nœud ordinaire,...

a) Construction des clusters

Un nœud peut avoir quatre états : *initial*, *cluster-head*, *ordinary-node* et *getway*. En plus d'un état interne *cluster-head-ready*.

Au début tous les nœuds sont initialisés à l'état *initial*, les nœuds restent dans cet état jusqu'à la réception d'une trame. Si l'état du nœud émetteur est *cluster-head*, alors le nœud récepteur passe à l'état *ordinary-node*. Dans le cas contraire il passe à l'état *cluster-head-ready*. Cet état est transitoire et signifie que le nœud n'a pas reçu un message de *cluster-head* et possède au moins un voisin. Quand un nœud dans l'état *cluster-head-ready* reçoit une trame d'un *cluster-head*, il passe à l'état *ordinary-node*. Si par contre, il continue à ne pas recevoir de messages depuis des *cluster-heads* jusqu'à sa prochaine émission d'une trame, il devient lui-même *cluster-head*. Ce changement sera signalé dans la trame émise.

Chaque nœud collecte les informations sur son voisinage en lisant l'identificateur et l'état des nœuds dans les trames reçus. Ainsi chaque nœud construit une table de voisinage contenant l'identificateur et l'état de chaque voisin écouté. Ces entrées sont estampillées par la date de réception. Pour éliminer les entrées trop vieilles le protocole parcourt la table à chaque émission ou réception.

b) Maintenance des clusters

Contrairement aux protocoles de clustering proactifs, dans le *passive clustering* la structure hiérarchique est maintenue et non reconstruite. Cette maintenance est implémentée par les deux règles suivantes :

- Quand un nœud s'aperçoit après balayage de sa table de voisinage, qu'il ne possède aucun cluster-head voisin, il se réinitialise en se mettant à l'état initial puis ré exécute le protocole à nouveau.
- Lorsqu'un nœud cluster-head reçoit un message d'un autre cluster-head il transite automatiquement vers l'état *ordinary-node*.

c) Connexion des clusters

Ce protocole utilise une heuristique pour élire les *getways*, un nœud se déclare *getway* si la condition suivante est vérifiée :

$$\alpha.NC + \beta > NG$$

Avec :

- NC : le nombre de cluster-heads voisins.
- NG : le nombre de *getways* voisins.
- α et β : deux paramètres du protocole permettant de contrôler le nombre de *getway*.

Dans les protocoles qui utilisent l'inondation (*flooding*), un nœud diffuse un message à tous ses voisins, les voisins transmettent à leur tour le message à leurs voisins jusqu'à ce que le message soit propagé dans tout le réseau. Cette inondation aveugle dégrade considérablement les performances du réseau en termes de bande passante et consommation d'énergie. Il y a plusieurs heuristiques pour réduire les rediffusions. Lors de la réception d'un paquet un nœud décide s'il le transmet à son voisin ou pas. Parmi les heuristiques, on trouve ceux qui se basent sur les distances entre les nœuds, sur la localisation des nœuds ou sur l'organisation en clusters [33]. Le *passive clustering* se diffère de ces heuristiques, parce qu'il fournit une plate-forme de l'inondation efficace basée sur les informations localement collectées (informations des voisins, états de cluster...etc.). Chaque nœud participe à l'inondation en se basant sur son rôle ou son état dans le cluster [33].

Les auteurs dans [34] ont combiné le *passive clustering* et *Direct Diffusion* [35] qui utilise le flooding dans les différentes phases de détermination des chemins de routage. L'objectif principal de cette combinaison est l'économie d'énergie dans les phases de flooding en impliquant que les cluster-heads et les *gateways* dans l'inondation. Les auteurs ont introduit des modifications dans le *passive clustering* par la sélection des cluster-heads et des *gateways* en se basant sur l'énergie résiduelle des nœuds et les distances entre eux. Ils proposent également d'appliquer le mode *sleep* périodiquement pour les membres des clusters ce qui exige une synchronisation entre des nœuds.

Un autre travail est présenté par les auteurs de [36] où est présentée une autre solution hybride du *passive clustering* et *Direct Diffusion* [35]. Cette solution essaye de tirer profit des avantages de chacune des solutions afin d'assurer une efficacité énergétique en terme de durée de vie globale du réseau et non seulement en terme de consommation d'énergie dans les nœuds. Pour ce faire les auteurs ont défini le niveau d'énergie d'un nœud (L) comme suit.

$$L = \frac{N_1 E_c}{E_i} \quad \text{Avec :}$$

- N_1 : Le nombre de niveau d'énergie.
- E_c : L'énergie consommée par le nœud.
- E_i : L'énergie initiale d'un nœud.

Ils ont introduit aussi la notion de candidature pour être un cluster-head ou un *gateway* en définissant le niveau d'énergie du réseau qui dépend du niveau d'énergie de tous les nœuds du réseau. Pour qu'un nœud soit un cluster-head ou *gateway*, son niveau d'énergie doit être inférieur au niveau d'énergie du réseau, un nœud cluster-head ou *gateway* ajuste la valeur de son niveau d'énergie du réseau et l'envoie avec son niveau d'énergie dans le prochain paquet transmis. En outre, un cluster-head ou un *gateway* abandonnera son rôle dès que son niveau d'énergie atteindra le niveau d'énergie du réseau.

Conclusion

Dans ce chapitre, nous avons donné un aperçu sur l'approche de clustering dans les réseaux sans fil de capteurs statiques. Nous avons commencé par la présentation de la notion de clustering ensuite nous avons détaillé les différentes phases de construction des clusters, la communication intra-cluster et inter-cluster et la maintenance des clusters, ainsi que les

différents objectifs d'une gestion en groupes d'un réseau de capteurs. Les principales techniques utilisées par les protocoles de clustering et publiées dans la littérature ont été étudiées et présentées. A chaque étape, les différents paramètres essentiels des stratégies de clustering, ont été précisés et critiqués en montrant l'apport d'une solution par rapport à une autre.

Bien que la technique de clustering semble prometteuse pour des réseaux tels que les réseaux de capteurs, il reste encore de nombreux défis à relever afin de l'adapter aux besoins de ces environnements à des coûts les plus bas possibles. D'autres challenges restent à défier dont la mobilité des nœuds capteurs. Effectivement, le monde du futur est un monde qui sera parsemé de capteurs. Ces derniers se retrouveront facilement sur les différents moyens de transport, sur les animaux, et même sur les être humains (suivi des prisonniers, des malades dangereux, ...). Enfin, un bon nombre d'applications existent et d'autres sont envisageables dans un futurs proche. Une question importante est alors de proposer des solutions adaptées dont de nouvelles approches de clustering plus efficaces et appropriées à ce nouveau type d'environnement.

Chapitre 3

Techniques de clustering dans les réseaux des capteurs mobiles

Introduction

La majorité des techniques de clustering proposées dans la littérature pour les réseaux de capteurs supposent que les nœuds capteurs sont stationnaires. Mais pour certaines applications, il est important de prendre en charge la mobilité des nœuds [37]. Sous de telles conditions, le clustering devient une tâche très ardue. Les nœuds changent continuellement leurs positions et les clusters se voient obligés de s'adapter aux différentes configurations du réseau. Lorsque un nœud membre se déplace en dehors de son cluster, les données captées par ce nœud seront perdues, ainsi une éventuelle réaffectation de ce nœud à un autre cluster est nécessaire pour maintenir la topologie en clusters du réseau.

Dans ce chapitre nous abordons la mobilité dans les réseaux de capteurs et ses influences sur la topologie hiérarchique du réseau. Ensuite nous décrivons les principales techniques de clustering proposées pour les réseaux de capteurs mobiles.

3.1. Les réseaux de capteurs mobiles

La mobilité est une question clé pour certains types de réseaux de capteurs. Particulièrement dans des applications militaires et de détection où les capteurs sont embarqués sur des dispositifs mobiles, des véhicules, des animaux ou des soldats. Dans ce type d'applications la mobilité est considérée comme un problème de premier ordre, Surtout lorsqu'elle est trop fréquente. Ceci est dû au changement fréquent des voisins et la reconfiguration continue du réseau qui exigent un nombre important de messages de contrôle, ce qui induit à une dépense importante de l'énergie. En outre, un autre type de mobilité pourrait être pris en compte, qui est la mobilité de la station de base. Cette mobilité induit à des difficultés dans la livraison des données à la station de base.

3.1.1. Types de capteurs mobiles

Les nœuds capteurs mobiles peuvent avoir des types différents selon leur mobilité dans le réseau:

- **Capteur mobile embarqué** (*Mobile Embedded Sensor*) : ce sont des nœuds qui ne contrôlent pas leur propre mouvement. Ils sont généralement dirigés par des forces externes ou attachés à des entités mobiles comme les animaux, les soldats ou les véhicules.
- **Capteur mobile autonome** (*Mobile Actuated Sensor*) : ce sont les nœuds qui peuvent avoir la capacité de contrôler leurs propres mouvements. Ils disposent d'un système de mobilité qui leur permet de parcourir toute la région de capture.

3.1.2. Architectures des MWSNs (*Mobile Wireless Sensor Networks*)

Les réseaux de capteurs mobiles peuvent être organisés selon leurs architectures: plates, 2-tiers, ou 3-tiers hiérarchiques [38].

Dans les architectures plates (Figure 12.a), le réseau comprend un ensemble de nœuds hétérogène et ils peuvent jouer le même rôle.

L'architecture en deux-tiers (Figure 12.b) consiste en un ensemble de nœuds stationnaires et mobiles. Les nœuds mobiles forment un réseau où ils se comportent comme des collecteurs de données (*data mules*) à partir d'autres nœuds dans le réseau [39]. Ces entités mobiles peuvent avoir des ressources supplémentaires de traitement, de communication et d'énergie. Le système *NavMote* est un exemple cette approche [40].

Dans l'architecture en trois-tiers (Figure 12.c) les nœuds stationnaires transmettent leurs données aux nœuds mobiles, qui à leur tour, les transmettent à l'ensemble des points d'accès. Ce réseau hétérogène est destiné à être déployé dans de larges réseaux et à être compatible avec plusieurs applications simultanément [38].

Par exemple, le déploiement d'un réseau de capteurs pour le contrôle de la disponibilité des places dans un parking. Le nœud capteur (premier tiers) diffuse des mises à jours de la disponibilité des places aux dispositifs mobiles (deuxième tiers), tels que un PDA ou un téléphone portable. A son tour, le dispositif mobile transmet cette donnée aux points d'accès (troisième tiers). Ensuite les données sont sauvegardées dans un serveur centralisé. Enfin, l'information de la disponibilité des places peut être consultée par les utilisateurs finaux [38].

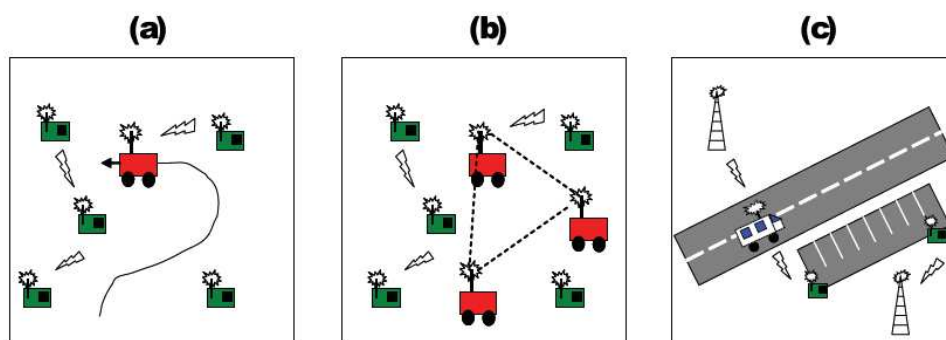


Figure 12. Les architectures (a) plate, (b) 2-tiers, (c) 3-tiers

3.1.3. WSNs et MWSNs

Afin de traiter l'effet de la mobilité des nœuds dans les réseaux de capteurs sans fil, il est important de montrer l'inadéquation des hypothèses sur les réseaux avec capteurs statiques dès lors que la mobilité est introduite. Ainsi, il n'est pas évident d'appliquer les solutions déjà connues pour les réseaux de capteurs.

- **La localisation:** dans les réseaux stationnaires, la position d'un nœud peut être déterminée une seule fois durant l'initialisation du réseau. Cependant, lorsque les nœuds sont mobiles, ils doivent obtenir leurs positions continuellement. Ce qui nécessite plus de temps et plus d'énergie, en plus de la nécessité de déploiement d'un service de localisation.
- **Le routage:** les protocoles de routage dans les réseaux de capteurs stationnaires décrivent la façon dans laquelle les données sont routées de la source vers la destination. Cela est assuré par la construction des tables de routage. Généralement, ces protocoles

utilisent les chemins les plus récents. Par contre, Dans le cas d'une topologie dynamique, les données de la table de routage deviennent non cohérentes plus rapidement, et les chemins doivent être reconstitué continuellement. Cela nécessite un coût considérable en termes d'énergie, de temps et de communication.

- **La consommation d'énergie:** les modèles de consommation d'énergie [41] diffèrent beaucoup entre les WSNs et les MWSNs. Pour les deux types de réseaux, les communications sans fil présentent un coût d'énergie significatif. Cependant, dans le cas des MWSNs, les entités mobiles nécessitent plus d'énergie pour gérer leur mobilité. Souvent, ces entités mobiles sont équipées d'une énergie importante, ou bien ils ont la capacité de recharger leurs batteries (*self-charging capability*).

3.2. Techniques de clustering dans les réseaux de capteurs mobiles

Il existe peu de techniques qui ont été proposées pour le clustering dans le cas des nœuds mobiles. Ces techniques peuvent être classifiées selon leur processus d'exécution. La première catégorie [14, 16, 42, 43...] consiste aux techniques traitant la mobilité des nœuds en *Offlines*. Ces techniques proposent des méthodes de formation des clusters qui résistent à la mobilité des nœuds le plus longtemps possible. Elles utilisent les informations de topologie comme la mobilité, le degré et la densité des nœuds pour la sélection des cluster-heads et la formation des clusters. L'autre catégorie englobe les techniques qui traitent la mobilité en *Onlines* [23, 46, 47...]. Dans de telles techniques, la mobilité est prise en charge par des mécanismes de maintenance de la structure des clusters. Ces mécanismes s'exécutent après la phase de construction des clusters, ils déclenchent des mises à jour des clusters dès qu'il y a un changement dans la structure d'un cluster.

Une autre classification qui prend en considération le type de mobilité peut être envisagée. Le premier type concerne les techniques qui supposent que la mobilité des nœuds est passive (c-à-dire que les nœuds sont embarqués sur des objets mobiles) [14, 16, 23, 42, 43 ...]. Le seul souci de ces techniques est de minimiser la consommation d'énergie dans les communications et d'assurer la stabilité de la topologie en clusters. Le deuxième type, ce sont les techniques qui supposent que les nœuds sont dotés d'un système de mobilisation [44]. Dans ce cas, le clustering doit prendre en considération l'énergie nécessaire pour assurer la mobilité des nœuds. Dans un troisième type de techniques, les nœuds sont supposés stationnaires mais la station de base mobile [45].

3.2.1. ACE-L (*Algorithm of Cluster-head Election by Location*)

Les auteurs dans [16] proposent une architecture basée sur le clustering pour la collecte de données dans les MWSNs afin de minimiser la consommation d'énergie dans ces réseaux.

Le processus de clustering est généralement subdivisé en deux phases; la phase *set-up* pour la sélection des cluster-heads et la formation des clusters et la phase *steady-state* pour la création des *schedules* et la transmission des données vers la station de base. Les auteurs proposent ACE-L un algorithme distribué pour la sélection des cluster-heads en se basant sur l'information de localisation des nœuds et ils font les hypothèses suivantes:

- La station de base est fixe et se situe loin du champ de captage.
- Les nœuds sont mobiles et homogènes en ressources physiques.
- Les nœuds sont équipés d'un système de localisation (GPS....).

Pour que les algorithmes de clustering donnent les meilleurs résultats en termes de consommation d'énergie et de prolongation de la durée de vie du réseau de capteurs, les auteurs mettent trois conditions pour la sélection des cluster-heads [16]:

- Il y a au moins un cluster-head élu à chaque round.
- Le nombre de cluster-heads généré doit être uniforme lors de tous les rounds.
- Les clusters générés doivent être de mêmes tailles.

Dans l'objectif de satisfaire ces trois conditions les auteurs de ACE-L ont défini quelques points de référence fixes sur le réseau dont leur nombre égal au nombre de clusters à générer. L'idée fondamentale de cette technique est d'utiliser la mobilité des nœuds pour alterner le rôle cluster-head sur les différents nœuds du réseau. A chaque période, les nœuds les plus proches de ces points de référence vont être sélectionnés comme cluster-heads.

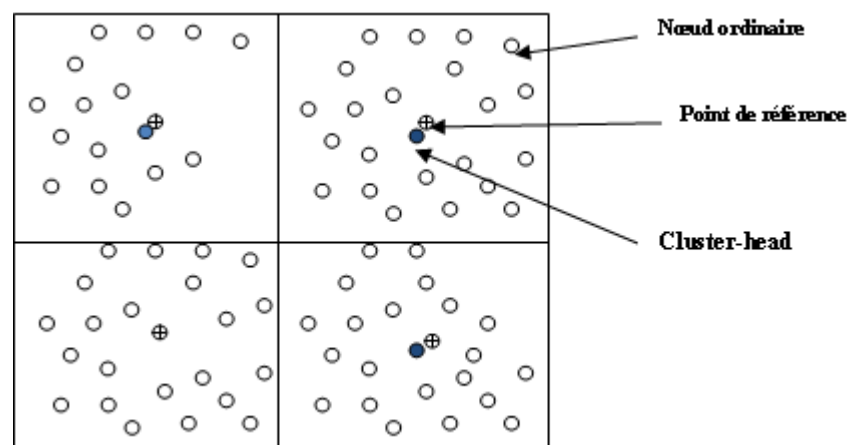


Figure 13. La sélection des cluster-heads dans ACE-L

Initialement tous les nœuds calculent les distances à tous les points de référence et définissent les points de référence les plus proches comme NRP (*Near Reference Point*). Ensuite, chaque nœud examine s'il y a reçu un message d'avertissement d'un autre nœud (cluster-head) et qui a le même NRP. S'il y'en a un, l'algorithme s'arrête. Sinon, le nœud établit un délai d'attente suivant la distance qui le sépare de son NRP. Pendant ce délai, il guette toute déclaration d'un cluster-head pour ce NRP. Au bout de ce délai, s'il ne reçoit aucune déclaration, ce nœud déclare par un message d'avertissement son élection comme cluster-head.

Algorithme

Entrées:

Le nœud détermine son NRP en se basant sur sa position et celles des points de référence.

delay time = fonction (distance)

Tant que (*delay time* > 0) **faire**

Si (réception d'un message d'avertissement d'un autre nœud *u* qui a le même NRP **alors**

cluster-head = true

Exit

Fin si

Fin Tant que

Envoyer un message d'avertissement pour déclarer son élection comme cluster-head

Fin si

Fin

Critiques : si on considère le cas où les nœuds s'arrêtent ou réduisent leur mobilité pour une période de temps, les mêmes nœuds peuvent s'élire comme cluster-heads pendant plusieurs périodes consécutives. Cela va causer la décharge de leurs batteries. Cette situation peut se produire aussi dans le cas où les nœuds se déplacent selon une trajectoire circulaire autour d'un point de référence.

Cet algorithme est sensible aux modèles de mobilité et de déploiement. Suivant le modèle, on peut avoir des clusters de tailles différentes à cause de non-uniformité de la distribution des nœuds dans le réseau. On peut avoir aussi moins de clusters par rapport au nombre de zones dans le cas où tous les nœuds capteurs se concentrent dans la même zone. En outre, la sélection des cluster-heads se fait sur la base de la distance qui sépare le nœud de son NPR. Ce critère n'est

pas toujours valable à cause de la variation du niveau d'énergie des nœuds capteurs. On peut choisir un nœud avec un niveau d'énergie faible, ce qui le rend plus vulnérable à la panne.

3.2.2. CM (*Clustering with Mobility*)

Dans la majorité des techniques de clustering, une fois les cluster-heads sont sélectionnés, l'étape qui suit est la formation des clusters où les nœuds choisissent le cluster-head le plus proche, en se basant sur la puissance du signal reçu. Si on néglige la mobilité des nœuds dans cette phase, il se peut qu'un nœud choisisse le cluster-head le plus proche, alors qu'en réalité ce cluster-head devient plus loin après un instant à cause de sa mobilité.

Pour faire face à ce problème, les auteurs dans [42] proposent une méthode pour l'affectation des nœuds membres aux clusters en prenant en compte la mobilité des nœuds. Cette méthode considère que les nœuds disposent d'un système de localisation et qu'ils sont capables de calculer leur vitesse et leur direction afin de prédire la position relative des nœuds par rapport aux cluster-heads au début de la phase de transmission de données.

A la fin de la phase de sélection des cluster-heads, chaque cluster-head diffuse un message d'avertissement (*ADV_CH*) qui contient sa position, sa vitesse et sa direction. Les nœuds membres calculent les distances d_1 et d_2 correspondant aux instants t_1 et t_2 . t_1 représente l'instant de la réception du message *ADV_CH* et t_2 est l'instant du début de la prochaine période. Les nœuds estiment les distances relatives aux cluster-heads à l'instant t qui correspond au début de la phase de transmission de données suivant la formule ci-dessus. Ensuite, Les nœuds choisissent le cluster-head le plus proche à cet l'instant.

$$D_i(t) = \frac{d2_i - d1_i}{t2 - t1} \times (t - t1) + d1_i, \quad i = 1, 2, \dots, n.$$

A cause de la difficulté de déterminer au préalable la valeur de l'instant de début de la phase de transmission de données, les auteurs ont fait des expériences qui montrent que cette méthode donne de meilleurs performances en consommation d'énergie si l'instant t vérifie l'équation suivante:

$$0.75 < ((t - t1) / (t2 - t1)) < 0.9$$

3.2.3. Protocole de clustering basé zone

Dans [43], les auteurs proposent un protocole de routage basé sur le clustering pour les réseaux de capteurs mobiles. Ce protocole se base sur la notion de zones pour créer les clusters où chaque nœud appartient à une seule zone. Les auteurs ont défini un facteur de mobilité

comme critère de sélection des cluster-head. Pour concevoir ce protocole les auteurs se basent sur les hypothèses suivantes:

- Les nœuds capteurs dans le réseau sont homogènes en termes de caractéristiques physiques comme la portée de communication et l'énergie.
- Les nœuds capteurs disposent d'un système de localisation (GPS....).
- Les nœuds capteurs sont mobiles.

a) Définition des zones

Afin de faciliter la division du réseau en zones les auteurs utilisent une forme géométrique carrée. En supposant que quelque soit la forme du réseau réel, il est toujours possible de l'entourer par un grand carré qui peut être ensuite partagé en petites zones carrées de même tailles et non recouvertes.

Pour définir la taille des zones carrées, les auteurs prennent en considération les facteurs suivants:

- la longueur du chemin désiré entre les cluster-heads et la station de base. Si on veut avoir des chemins longs en termes de nombre de saut, on définit des zones de petites tailles et l'inverse si l'on veut avoir des chemins courts.
- la portée maximale de transmission des nœuds capteurs MTR (*Maximum Transmission Range*) [43].
- la vitesse maximale de chaque nœud capteur MMV (*Maximum Moving Velocity*) [43].

b) Sélection des cluster-heads et construction des clusters

Comme toute méthode de clustering, chaque cluster ou zone a un cluster-head qui joue le rôle de passerelle entre les nœuds membres d'un cluster et la station de base via d'autres cluster-heads. Pour le choix du cluster-head, on utilise un facteur de mobilité défini par le nombre de fois qu'un nœud se déplace d'une zone à une autre pendant un période de temps. Le nœud qui a le plus petit facteur de mobilité dans sa zone est choisi comme cluster-head.

Le cluster-head joue un rôle très important dans ce protocole; il est responsable de l'envoi des demandes de découverte des routes aux zones voisines, de l'agrégation et l'envoi des données des nœuds membres à la station de base, et enfin de la sauvegarde des données des nœuds membres et le maintien des tables de routage.

Au début de chaque round, tous les nœuds sont initialisés comme cluster-heads, ensuite si un nœud détecte la présence d'un autre cluster-head dans la même zone et ayant un facteur de

mobilité plus petit, il abandonne ce rôle et devient un nœud membre de ce cluster-head. A la fin de cette phase, un seul cluster-head sera définis dans chaque zone comme le montre la figure suivante.

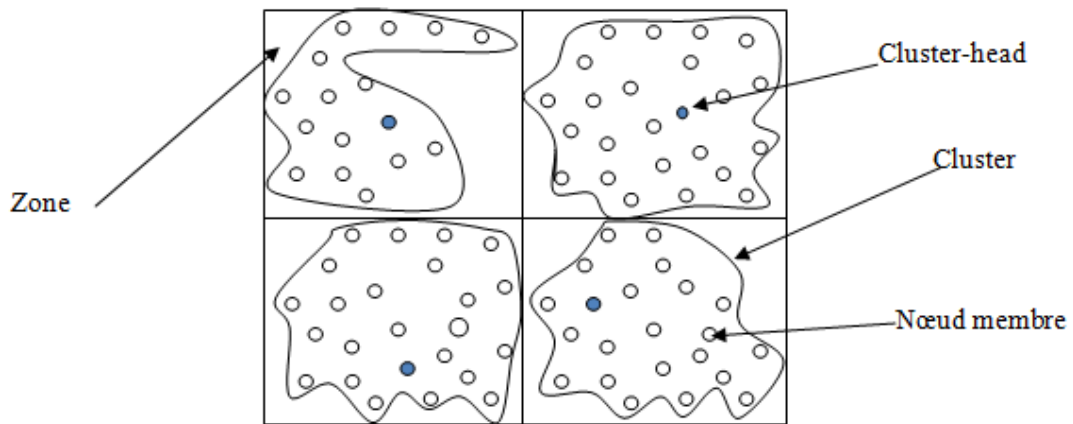


Figure 14. La sélection des cluster-heads

3.2.4. CES (*Cluster-based Energy-efficient Scheme for Mobile Wireless Sensor Networks*)

La technique proposée dans [14] consiste à regrouper les nœuds proches géographiquement en clusters, elle implique des paramètres déterminants pour produire un nombre réduit de clusters homogènes en taille et en rayon et qui sont stables. Le poids de chaque capteur est calculé en fonction des paramètres suivants : 2-densité [14], énergie résiduelle et mobilité. Le capteur ayant le plus grand poids dans son voisinage à deux sauts devient le cluster-head. En outre, la taille des clusters générés est comprise entre deux seuils **ThreshLower** et **ThreshUpper**, qui représentent respectivement le nombre minimal et maximal de capteurs dans un cluster. Ces deux seuils sont choisis arbitrairement ou dépendent de la topologie du réseau. Dans un cluster, chaque capteur membre est au plus à deux sauts de son cluster-head correspondant. Cette technique fait les hypothèses suivantes:

- chaque capteur possède une antenne omnidirectionnelle lui permettant par une transmission simple de couvrir tous les capteurs se trouvant dans son voisinage.
- les capteurs sont déployés dans un espace bidimensionnel.
- les capteurs sont stables durant une période raisonnable pendant l'exécution du processus de clustering.

a) Notations et structures

Dans CES chaque cluster a un cluster-head qui est choisi dans son voisinage à deux sauts selon la valeur de son poids qui est une combinaison de : k-densité [14], énergie résiduelle et

mobilité. Les auteurs ont défini des coefficients (α, β, γ) pour chaque paramètre. Ces coefficients peuvent être choisis selon l'application. Par exemple, dans les réseaux de capteurs déployés dans des zones hostiles et inaccessibles, on favorise le paramètre énergie sur les autres pour assurer une longue durée de vie au réseau. Par contre, dans les réseaux de capteurs à forte mobilité, on favorise les paramètres k-densité et mobilité pour garantir la stabilité de l'architecture hiérarchique.

Dans cette technique chaque capteur est identifié par un vecteur d'état comme suit :

$$\mathbf{Nœud}(\mathbf{Node}_{Id}, \mathbf{Node}_{CH}, \mathbf{Weight}, \mathbf{Hop}, \mathbf{Size}, \mathbf{Thresh}_{Lower}, \mathbf{Thresh}_{Upper})$$

- $\mathbf{Node}_{Id}$: l'identifiant du capteur.
- $\mathbf{Node}_{CH}$: l'identifiant de son cluster-head.
- $\mathbf{Hop}$: le nombre de sauts à son cluster-head.
- $\mathbf{Size}$: la taille du cluster auquel il appartient.
- $\mathbf{Weight}$: le poids du nœud calculé comme suit :

$$\mathbf{Weight}(u) = \alpha * 2 - \mathbf{density}(u) + \beta * \mathbf{Res} - \mathbf{Energie}(u) + \gamma * \mathbf{Mobility}(u)$$

$$\alpha + \beta + \gamma = 1$$

Tel que :

- $2 - \mathbf{density}(u)$: représente le rapport entre le nombre de liens d'un nœud u dans son voisinage à deux sauts (liens entre u et ses voisins et liens entre deux voisins à deux sauts de u) et son nombre de voisins à deux sauts.
- $\mathbf{Res} - \mathbf{Energie}(u)$: L'énergie résiduelle d'un nœud capteur.
- $\mathbf{Mobility}(u)$: La mobilité d'un nœud u représente la vitesse de déplacement de u relativement à tous ses voisins directs durant un intervalle de temps Δt .
- α, β et γ : des coefficients fixés par l'administrateur. Ils dépendent de l'application.

Chaque nœud capteur est responsable aussi du maintien d'une table appelée $\mathbf{Table}_{Cluster}$ pour sauvegarder les informations des membres locaux du cluster. Les nœuds capteurs collaborent entre eux pour construire et mettre à jour cette table en utilisant des messages "Hello". En plus, chaque cluster-head a une table appelée $\mathbf{Table}_{CH}$, dans laquelle l'information des cluster-heads est stockée. Le format de ces deux tables est défini comme suit :

$$\mathbf{Table}_{Cluster}(\mathbf{Node}_{Id}, \mathbf{Node}_{Ch}, \mathbf{Weight})$$

$Table_{CH}(Node_{CH}, Weight)$

Afin de générer des clusters homogènes en tailles les auteurs ont définis une phase de ré-affiliation des nœuds. A la fin du processus de formation des clusters, si la taille de ces derniers ne respecte pas les seuils ***ThreshLower*** et ***ThreshUpper*** la phase de ré-affiliation est exécutée.

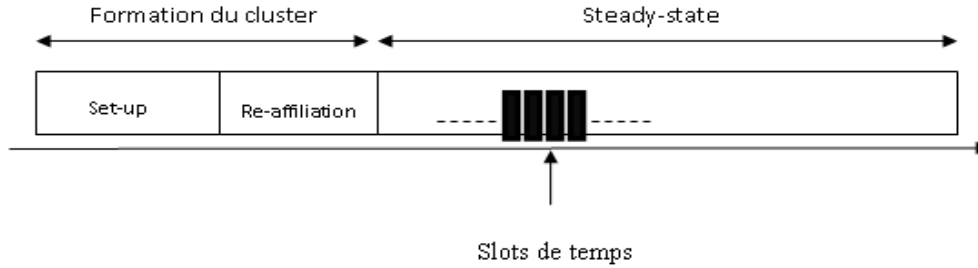


Figure 15. Les phases de clustering dans CES

b) construction des clusters

Au début de chaque période, chaque nœud capteur calcule son poids et génère un message *Hello* qui contient : *Weight* et *Node_{CH}*, où *Node_{CH}* est initialisé à zéro. Puis, il le diffuse à ses voisins à deux sauts. Le nœud capteur ayant le plus grand poids parmi ses voisins à deux sauts non encore affiliés est élu cluster-head pour la période courante. Ce dernier met à jour son vecteur d'état en assignant la valeur de son identifiant *Node_{Id}* à *Node_{CH}* et attribue respectivement à *Hop* et *Size* les valeurs 0 et 1. Puis, il diffuse le message d'avertissement *ADV_CH* contenant son vecteur d'état dans son voisinage à deux saut pour les inviter à le rejoindre. Tout nœud capteur de ses voisins directs qui reçoit le message de diffusion et qui n'est pas encore affilié à d'autres clusters et dont le poids est inférieur à celui du cluster-head, transmet le message *REQ_JOIN* à ce cluster-head pour le rejoindre. Le cluster-head correspondant vérifie si la taille de son cluster n'a pas encore atteint le *ThreshUpper*. Il transmettra alors un message *ACCEPT_CH* à ce nœud demandant l'affiliation, sinon il ignorera le message de demande d'adhésion *REQ_JOIN*. Si l'adhésion d'un nœud est acceptée alors le cluster-head incrémente de 1 la valeur du paramètre *Size* et le nouveau nœud affilié assigne respectivement aux paramètres *Hop* et *Node_{CH}* les valeurs 1 et *Node_{Id}* de son cluster-head. Puis, les nouveaux nœuds affiliés transmettront de nouveau le message reçu avec la même puissance de transmission à leurs voisins. De la même manière, chaque nœud appartenant aux voisinages à deux sauts du cluster-head, qui n'est pas encore affilié et dont le poids est inférieur à celui du cluster-head, transmet un message *REQ_JOIN* au cluster-head correspondant. Ainsi, le cluster-

head vérifie si la valeur du paramètre *Size* est toujours inférieure à *ThreshUpper*, il transmet alors un message *ACCEPT_CH* à ce nœud et il met à jour son vecteur d'état, sinon il ignore le message de demande d'adhésion. Après la fin de l'exécution de cette phase, tout nœud connaît à quel cluster il appartient et quel est son cluster-head correspondant.

c) La phase de ré-affiliation

Durant la phase de construction des clusters, il est possible qu'il y ait des clusters créés dont la taille inférieure à *ThreshLower*, puisqu'il n'y a aucune contrainte imposée sur la génération de ce type de clusters dans la première phase. De ce fait, dans la phase de ré-affiliation, les nœuds appartenant aux clusters qui n'ont pas atteint la taille *ThreshLower* sont ré-affiliés à ceux qui n'ont pas atteint *ThreshUpper* afin de réorganiser les clusters formés, réduire leur nombre et obtenir des clusters homogènes en taille.

L'exécution de cette phase se déroule de la façon suivante: les cluster-heads qui appartiennent aux clusters dont la taille est strictement inférieure à *ThreshUpper* et plus grande que *ThreshLower* émettent un nouveau message appelé *RE-AFF_CH* pour demander aux nœuds appartenant aux clusters dont la taille est inférieure à *ThreshLower* de les rejoindre. Tout nœud qui reçoit le message et qui appartient à un cluster dont la taille est inférieure à *ThreshLower* devrait être ré-affilié au cluster-head le plus proche en se basant sur la puissance du signal reçu.

3.2.5. LEACH- Mobile

Comme ça était mentionné auparavant, le protocole LEACH est exécuté en deux phases ; *set-up* et *steady-state*, la première pour l'élection des cluster-heads, la construction des clusters et la création d'un schedule TDMA et la deuxième est dédiée à la transmission des données vers la station de base. Cette dernière, est d'une durée plus grande que la première et elle est divisée en frames. Chaque frame est divisé en slots de temps réservés aux nœuds membres pour transmettre leurs données vers le cluster-head.

Chaque nœud membre envoie ses données collectées au cluster-head pendant le slot qui lui a été réservé. L'envoi est effectué à une portée égale à la distance qui sépare le nœud de son cluster-head. Cette distance est calculée en basant sur la puissance du signal des messages *ADV_CH* reçus pendant la phase *set-up*. Dans un environnement où les nœuds sont mobiles, les distances qui séparent les nœuds de leurs cluster-heads changent. Donc, dans le cas où les nœuds

membres s'éloignent de leurs cluster-heads, les messages de données envoyés vers ces derniers vont être perdus.

Les auteurs de LEACH-Mobile [23] ont proposé une solution pour minimiser le taux des messages perdus. L'idée principale de cette solution est comme suit:

- On suppose que chaque nœud a des données à transmettre au cluster-head pendant son slot de temps
- A chaque slot réservé au nœud i , le cluster-head lui envoie un message *REQ-DATA* pour lui demander la transmission de ses données.
- Si le cluster-head ne reçoit pas de données depuis un nœud membre pendant deux frames successifs, il le considère absent et il recrée le schedule TDMA (ordonnancement TDMA) des éventuels nouveaux nœuds.
- Si un nœud membre ne reçoit pas le message *REQ-DATA* pendant deux frames successifs il considère que le cluster-head est inatteignable et il diffuse un message *REQ_JOIN* pour demander l'adhésion à un nouveau cluster. Si un cluster-head reçoit ce message, il lui envoie un message d'avertissement et ainsi commence la phase *set-up*. Si un nœud reçoit plusieurs messages d'avertissements, il fait son choix suivant la force du signal reçue.

Dans les deux cas précédents, les cluster-heads mettent à jour leurs listes de membres et créent un nouveau TDMA schedule et le diffusent à leurs membres. Les simulations prouvent que LEACH-Mobile donne de meilleurs résultats que LEACH en termes de taux de réussite de transferts des données avec l'augmentation du taux des nœuds mobiles. Cependant, LEACH-Mobile consomme plus d'énergie comparativement à LEACH à cause du nombre important des messages de contrôles.

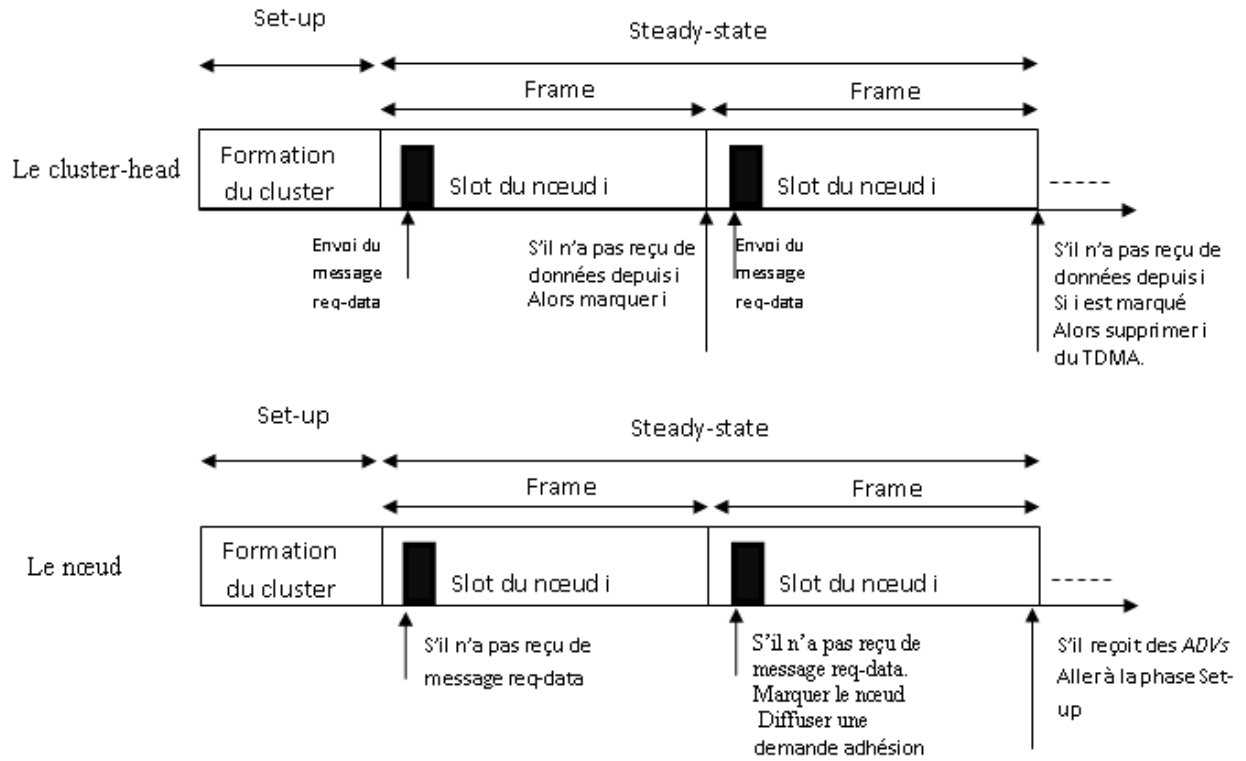


Figure 16. Le mécanisme de maintenance des clusters dans LEACH-Mobile

3.2.6. LEACH-ME (*Mobile-Enhanced*)

L'idée principale de protocole LEACH-ME présenté dans [46] est que les nœuds ayant une faible mobilité relative à leurs voisins doivent être sélectionnés comme des cluster-heads. Au début de chaque période, les cluster-heads sont sélectionnés d'une façon aléatoire comme dans le protocole LEACH. Cependant, durant la phase de transmission de données « Steady-stat » les nœuds membres enregistrent les transitions entre les clusters. Durant chaque frame, les nœuds membres envoient le nombre total de transitions à leurs cluster-heads pendant leurs slots réservés. A la fin de chaque frame, les cluster-heads calculent la moyenne des transitions des nœuds membres pendant la dernière période. Si la valeur dépasse un seuil prédéterminé, il introduit une période *ACTIVE* (*active slot*) durant laquelle les clusters seront réorganisés. Durant la période active, les nœuds diffusent leurs identifiants. Chaque nœud estime la distance à tous ses voisins et calcul le facteur de la mobilité comme suit :

$$M_i = \frac{1}{N-1} \sum_{j=0}^{N-1} d_{ij}(t)$$

Où $M_i(t)$ est le facteur de mobilité basé sur l'éloignement d'un nœud i de ses voisins, N est le nombre de voisins d'un nœud i , $d_{ij}(t)$ est la distance entre un nœud i et son voisin j . Les nœuds ayant un facteur de mobilité faible seront sélectionnés comme des cluster-heads, à condition que son niveau d'énergie dépasse un certain seuil. Le reste de la phase Steady-state est le même que dans LEACH-Mobile.

3.2.7. CBR-Mobile

Les auteurs dans [47] ont proposé une amélioration du protocole LEACH-Mobile, en proposant une nouvelle technique qui minimise les pertes de paquets. Le protocole proposé s'appelle le CBR Mobile-WSN (*Cluster Based Routing Protocol for Mobile Nodes in Wireless Sensor Network*). Dans ce protocole, le cluster-head envoie un message de demande de données (*REQ-DATA*) à ses membres, si le cluster-head ne reçoit pas de données depuis un nœud membre, le paquet est considéré perdu et par conséquent le cluster-head supprime ce nœud de sa liste des membres à la fin du frame. D'autre part, si le nœud capteur ne reçoit pas le message de demande de données du cluster-head, il essaye d'établir une nouvelle demande d'adhésion au nouveau cluster pour éviter la perte des paquets. Lorsque le nœud capteur reçoit le message de demande de données du cluster-head mais il n'a aucune donnée à envoyer, le nœud n'occupe pas le slot de temps qui lui a été réservé. Et ainsi, le slot de temps sera assigné à un autre nœud membre qui a des données à envoyer. Chaque cluster-head garde des slots de temps libres pour permettre aux nœuds sortants d'autres clusters de le rejoindre.

Ce protocole adaptatif permet d'éviter le gaspillage des slots de temps. Par conséquent, il assure une bonne utilisation de la bande passante.

Les hypothèses suivantes sont considérées :

- Tous les nœuds sont homogènes en ressources physiques.
- La station de base est fixe et loin du réseau.
- Tous les nœuds sont capables d'envoyer leurs données directement à la station de base.
- Tous les nœuds capteurs non cluster-heads sont mobiles.
- Pendant chaque période les nœuds élus comme cluster-head sont stationnaires.

Dans ce protocole, le cluster-head reçoit les données de ses membres et d'autres nœuds capteurs non assignés à des cluster-heads pendant des slots de temps libres (Figure 17) lorsqu'il a

le rôle du cluster-head libre. Le rôle du cluster-head libre dans le réseau est affecté alternativement aux cluster-heads de façon round-robin. Si un nœud capteur ne reçoit pas le message de demande de données de son cluster-head pendant le slot de temps assigné par le TDMA, il envoie ses données au cluster-head libre pour éviter leurs pertes. Ensuite, il envoie le message de demande d'adhésion aux cluster-heads voisins. Lorsque le cluster-head finit de recevoir les données de tous ses nœuds membres, il retire les nœuds qui n'ont pas envoyé leurs données et vérifie ensuite s'il reçoit des messages de demande d'adhésion des nœuds perdus et il leur envoie des messages d'avertissements. Les nœuds perdus choisissent les nouveaux cluster-heads en leur envoyant des messages de confirmation. Les cluster-heads ajouteront ces nœuds au schedule TDMA.

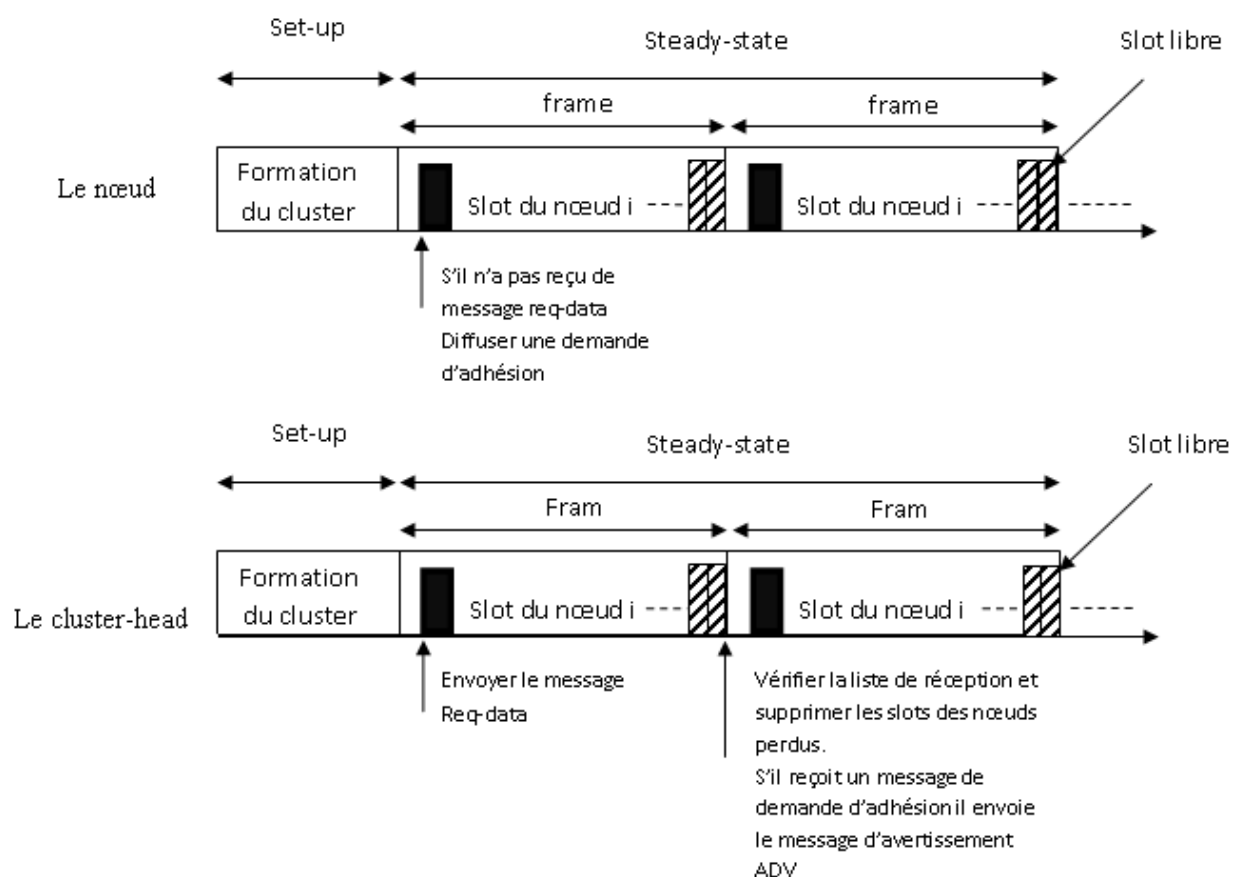


Figure 17. Le mécanisme de maintenance des clusters dans CBR Mobile

Chaque nœud se réveille avant son slot réservé d'un un slot de temps, Si le cluster-head lui a assigné le slot de temps précédent, il envoie ses données pendant ce slot, sinon il se remet en mode sommeil et il se réveille pendant son slot. Les slots de temps des nœuds qui sont sortis

du cluster ou qui n'ont pas de données à envoyer peuvent être attribués aux nœuds réveillés ou à ceux qui sont entrés dans le cluster.

Cette adaptation du protocole CBR Mobile-WSN permet :

- aux nœuds qui entrent dans le cluster de le rejoindre le plus tôt possible.
- une utilisation efficace de la bande passante en évitant le gaspillage inutile des slots de temps.
- D'éviter la perte de données qui peut se produire par la déconnexion des nœuds sortant du cluster.

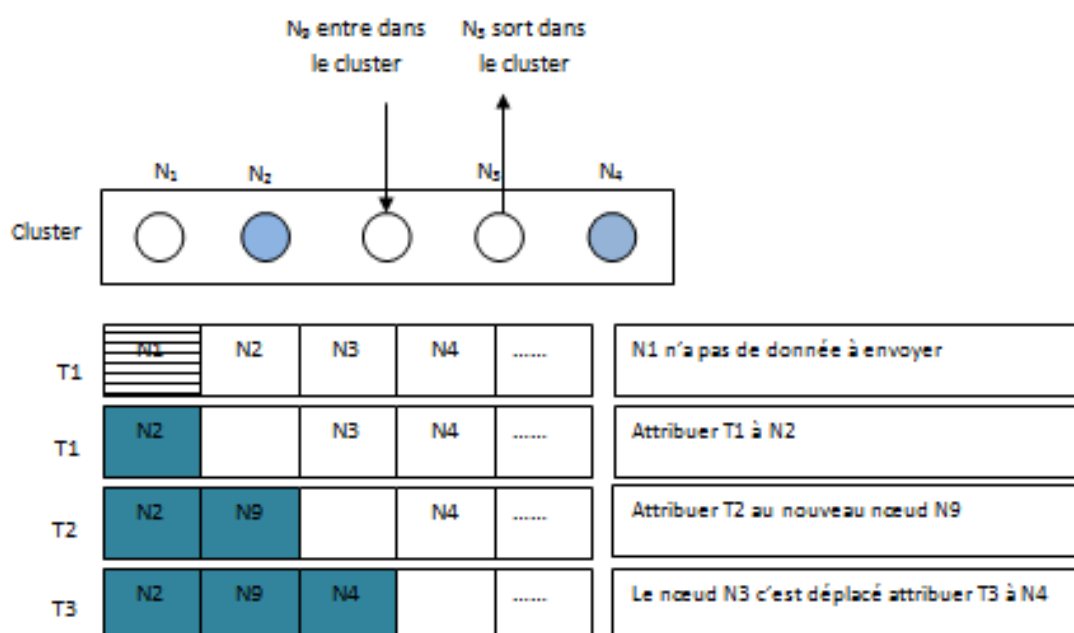


Figure 18. Mécanisme d'adaptation du TDMA dans CBR-Mobile

3.2.8. Synthèse:

A partir de cette étude que nous avons faite, nous pouvons tirer les remarques suivantes :

- Toutes les techniques de clustering proposées pour les réseaux de capteurs avec nœuds mobiles s'exécutent de façon périodique. En effet, chaque période comporte deux grandes phases : (a) la phase *Set-up* pour la sélection des cluster-heads et la construction des clusters (b) la phase *Steady-stat* pour la transmission de données à la station de base.
- La majorité des techniques proposées utilisent des critères basés sur la mobilité pour le choix des cluster-heads. Dans CES, les auteurs utilisent la k-densité afin d'assurer plus de

stabilité aux clusters générées. Dans les autres techniques (ACE-I, LEACH-ME...) ils utilisent des critères liés à la mobilité pour le choix des cluster-heads.

- La majorité des techniques étudiées supposent que la mobilité des nœuds est passive (ne nécessite pas un système de mobilisation).

Le tableau suivant résume les techniques que nous avons étudiées :

Processus d'exécution	Technique	Critères de sélection des cluster-heads	Critères d'adhésion au cluster-head	Communication intra-cluster /inter-cluster	Objectifs
Offline	ACE-L	k-densité/énergie résiduelle/mobilité	Distance en sauts	Directe/directe	Stabilité /énergie
	CES	Mobilité	Distance	2-sauts/multi-hop	Energie
	CM	Aléatoire	Distance/mobilité	Directe/directe	Stabilité
	CBZ	Facteur de mobilité	Appartenance à la zone	Directe/multi-hop	Stabilité
Online	LEACH-M	Aléatoire	Distance	Directe/directe	Stabilité/énergie
	CBR-M	Aléatoire	Distance	Directe/directe	Stabilité/énergie
	LEACH-ME	Aléatoire/ mobilité	Distance	Directe/directe	Stabilité/énergie

Tableau 1. Tableau comparatif des techniques de clustering pour les réseaux de capteurs mobiles

Conclusion

Dans ce chapitre nous avons abordé le problème de la mobilité dans les réseaux de capteurs sans fil, ses types et son influence sur le processus de clustering. Ensuite nous avons énuméré les techniques de clustering proposées dans la littérature pour les réseaux de capteurs avec nœuds mobiles. Ces techniques peuvent être classées en deux catégories. Dans la première catégorie, les techniques sont conçues spécialement pour les réseaux avec des nœuds mobiles. Elles prennent en générale la mobilité comme facteur de sélection des clusters-heads et de formation des clusters. Dans la deuxième catégorie, les techniques proposés essayent de maintenir la structure de clusters en proposant des mécanismes de mis à jour.

Toutes les techniques que nous avons étudiées traitent le cas de la mobilité aléatoire des nœuds et ne prennent pas en charge la mobilité en groupe des nœuds qui se rapprochent plus de la réalité relativement aux autres types de mobilité. Dans le chapitre suivant nous allons proposer une technique de clustering qui s'adapte à ce type de mobilité en définissant un critère adapté à cette mobilité.

Chapitre 4

Protocole de clustering distribué pour les réseaux de capteurs mobiles

Introduction

Les réseaux de capteurs sans fil ont attirés ces dernières années la communauté des chercheurs pour faire face aux défis imposés par les contraintes des nœuds capteurs : petite capacité de stockage, basse capacité de traitement, durée de vie limitée de la batterie et portée radio réduite. Les réseaux de capteurs sans fil ont plusieurs applications dans différents domaines telles que les applications médicales, militaires, environnementales et de la sécurité. Parmi ces applications certaines posent de nouvelles contraintes comme la mobilité des nœuds. Ces derniers peuvent être placés sur des patients, des soldats, des animaux ou ils peuvent disposer d'un système de mobilisation.

La mobilité des nœuds capteurs influe beaucoup sur l'acheminement de l'information dans les réseaux de capteurs sans fil à cause du changement fréquent de la topologie. Ceci pose des problèmes comme la perte de connectivité entre un nœud capteur et ses voisins, et la surcharge générée par la redondance des informations acheminées des nœuds capteurs vers la station de base ou dans le sens inverse. Dans le but de pallier au problème du changement fréquent de la topologie et pour apporter plus de stabilité à la structure du réseau, l'architecture basée sur le clustering est considéré comme une approche intéressante.

Dans la littérature, la majorité des protocoles de clustering sont conçus pour des réseaux de capteurs avec nœuds stationnaires dont les plus connus LEACH [10], HEED [28] et TEEN [30]. Ces protocoles s'exécutent de façon périodique par « round », de sorte à affecter alternativement le rôle cluster-head aux nœuds par soucis d'une distribution équitable de la charge sur le réseau. Chaque round ou période comporte une phase pour l'élection des cluster-heads et une autre plus stable et plus longue pour la transmission de données vers la station de base. D'autre part, il existe peu de travaux qui traitent du clustering dans les réseaux de capteurs sans fil avec mobilité des nœuds. Dans [23], les auteurs proposent une amélioration du protocole LEACH pour qu'il supporte la mobilité des nœuds. L'idée principale du protocole proposé (LEACH-Mobile) est de prévoir un mécanisme pour traiter la migration des nœuds d'un cluster à un autre en définissant

de nouveaux messages de contrôle (*REQ-DATA*). Ces derniers vont être envoyés par le cluster-head à ses membres pendant leurs slots de temps réservés pour transmettre les données au cluster-head. Si le cluster-head ne reçoit pas de données après l'envoi de deux messages *REQ-DATA* d'un nœud membre, il le considère absent et il procède à la mise à jour du schedule TDMA avec éventuels nouveaux nœuds qui rejoignent le cluster. D'autre part, si un nœud membre ne reçoit pas de message *REQ-DATA* après deux frames, il se considère en dehors du cluster et il diffuse un message de demande d'appartenance (*REQ_JOIN*) à un autre cluster-head. Dans [14], les auteurs proposent une technique pour regrouper les nœuds proches géographiquement en clusters en impliquant des paramètres déterminants pour produire un nombre réduit de clusters stables et homogènes en taille et en rayon. Le poids de chaque nœud capteur est calculé en fonction des paramètres suivants: 2-densité qui représente le nombre de liens entre un nœud et ses voisins divisé par leurs nombre, énergie résiduelle et mobilité. Le nœud capteur ayant le plus grand poids dans son voisinage à deux sauts devient cluster-head. En outre, la taille des clusters générés est comprise entre deux seuils : *ThreshLower* et *ThreshUpper*, qui représentent respectivement le nombre minimal et maximal de nœuds capteurs dans un cluster. Ces deux seuils sont choisis arbitrairement ou en fonction de la topologie du réseau. Dans un cluster généré, chaque nœud capteur membre est au plus à deux sauts de son cluster-head correspondant. Dans [16], les auteurs proposent deux algorithmes pour la sélection des cluster-heads garantissant la formation d'un nombre fixe de clusters. Le premier algorithme se base sur les identifiants des nœuds pour choisir les cluster-heads et attribuer régulièrement le rôle cluster-head aux différents nœuds. Le deuxième algorithme se base sur l'information de localisation des nœuds en définissant un nombre fixe de points de référence, les nœuds les plus proches de ces points deviennent des cluster-heads.

Une bonne distribution de la charge entre les nœuds capteurs est une des issues clés pour rallonger la durée de vie d'un réseau de capteurs. Lors de la répartition d'un réseau de capteurs en clusters, deux critères doivent être considérés : le nombre de clusters et leur taille. En effet, ces paramètres peuvent influencer les performances du réseau ainsi défini. Ces paramètres peuvent être étudiés pour contribuer à cet équilibre de la charge réseau entre les nœuds capteurs. Tout en sachant que dans les méthodes de clustering, si les clusters générés ont différentes tailles, les cluster-heads gérant un grand nombre de membres vont consommer plus d'énergie et vice versa. D'où une durée de vie moindre et un risque important de partitionnement du réseau. Le partitionnement du réseau empêchera l'acheminement des données même si des capteurs assurent toujours leurs tâches.

Une bonne méthode de clustering est celle qui génèrerait, à chaque phase, un nombre fixe de clusters de taille homogène. Dans cet objectif, nous proposons en premier lieu “*Density based ACE-L*” (*DACE-L*) une amélioration de l’algorithme ACE-L proposé dans [16] afin de générer des clusters plus homogènes en taille et en nombre. Cette proposition considère aussi le cas de réseau non uniforme, c'est-à-dire, où la densité des nœuds n’est pas homogène. Nous admettons, en plus, de la définition de points de référence principaux, des points de références secondaires dans chaque zone.

D’autre part, nous proposons un deuxième algorithme de clustering GMAC pour les réseaux de capteurs mobiles. Cette technique prend en considération le déplacement en groupes des nœuds capteurs. En effet, cette propriété est souvent constatée dans beaucoup d’applications (par exemple, le déplacement de troupes militaires). L’objectif de la proposition est d’améliorer les performances du réseau de capteurs en exploitant cette propriété de la mobilité des capteurs de ces applications. En effet, ce phénomène peut garantir plus de stabilité à la topologie en clusters. D’où une maintenance réduite et plus efficace avec une limitation en communications et en dépense énergétique.

4.1. Modèle d’environnement et hypothèses

Nous considérons un environnement constitué de nœuds capteurs mobiles interconnectés en réseau. Ces nœuds capteurs sont embarqués sur des entités mobiles telle que les patients, les soldats ou les animaux ou bien ils disposent d’un système de mobilité que nous supposons autonome en terme de consommation d’énergie. Ainsi, nous adoptons les hypothèses suivantes :

- La station de base est fixe et située à une distance importante du champ de captage ;
- Tous les nœuds capteurs sont mobiles et homogènes en ressources physiques ;
- Les nœuds capteurs sont alimentés par des batteries à durée de vie limitée ;
- Tous les nœuds sont équipés d’un système de localisation (GPS ou autre) ;
- Les nœuds sont synchronisés en temps ;
- Les nœuds sont déployés dans un espace bidimensionnel.

4.2. Density based ACE-L

Assurer une distribution équitable des tâches de cluster-heads est un objectif qui n’est pas évident sur une densité non uniforme des nœuds capteurs dans le temps et dans l’espace, et sans une connaissance préalable et coûteuse de la topologie du réseau. Dans [16], les auteurs ont

définit les points de référence pour fixer le nombre de clusters à former. Les nœuds les plus proches de ces points de référence sont sélectionnés comme cluster-heads. Ils supposent que ces points sont uniformément répartis dans le réseau, mais ils ne définissent aucune technique pour déterminer leur nombre et leurs positions dans le réseau. L'hypothèse d'une répartition uniforme des nœuds capteurs dans la zone d'intérêt n'est pas réaliste pour la majorité des applications envisageables comme les applications militaires, médicales et de secours, où il y a une grande variation de la densité des nœuds dans le réseau.

Dans notre proposition, nous définissons deux types de points de référence: des points de référence principaux et d'autres secondaires comme le montre la figure suivante. Nous formons des clusters autour des points de référence principaux et dans le cas où la taille d'un cluster dépasse un seuil prédéfini, nous procédons immédiatement à sa disjonction en plusieurs clusters autour des points de référence secondaires, que nous définissons selon le nombre de nœuds membres et leurs positions. Le nombre de clusters et les points de référence secondaires autour desquelles nous formons ces nouveaux clusters dépendent de nombre de nœuds dans le cluster d'origine et leurs positions.

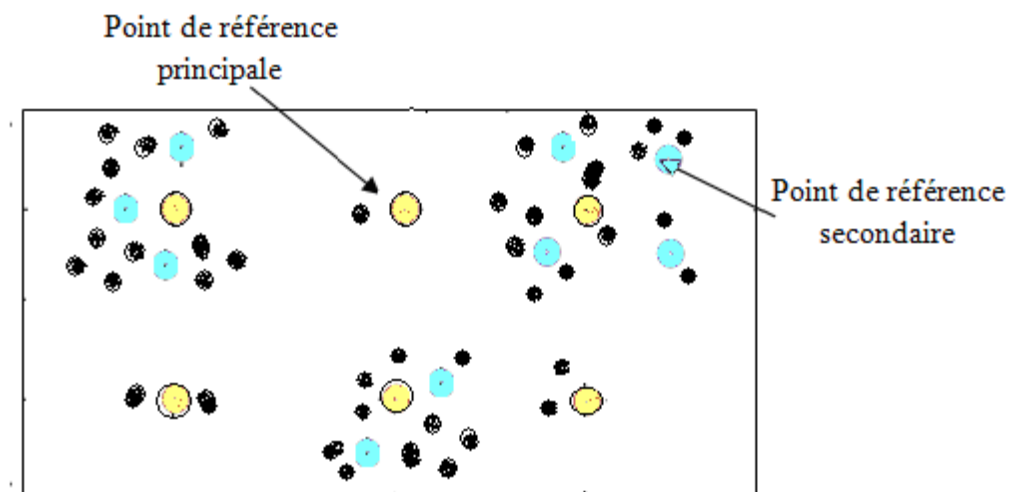


Figure 19. Points de références principaux et secondaires

Ce processus permet la construction de clusters homogènes en taille. Si la taille d'un cluster n'atteint pas un seuil minimal, nous ne créons pas de cluster autour de ce point de référence principale. L'estimation du nombre de points de référence principaux se fait par une évaluation préalable du nombre approprié de clusters pour l'application visée. Ce nombre dépend de plusieurs paramètres discutés dans [10] [48] tels que la surface de la zone de déploiement, le

nombre de nœuds, la mobilité et la distribution des nœuds capteurs ainsi que la position de la station de base.

Le choix des points de référence principaux se fait par l'administrateur dans le cas où nous avons une concentration des nœuds capteurs dans des zones fixes et connues au. Par exemple, dans les applications de secours, nous aurons dans la zone d'intérêt, des zones de soins, d'incidents et de regroupements des équipes de secours. Dans le cas contraire, où la distribution des nœuds n'est pas connue au préalable, nous générons ces points de référence de façon générale et automatique, par exemple, en définissant le centre des zones géographique en tant que points de référence.

4.2.1 Génération des points de référence

Pour la génération des points de référence principaux, nous proposons deux méthodes à exécuter au niveau de la station de base. A partir de ces points de référence principaux des points de référence secondaires sont définis.

a) Première méthode: Génération des points de référence basée rectangles

Nous considérons que l'espace occupé par les nœuds capteurs est représenté par une surface rectangulaire. Soit k le nombre estimé de clusters. Nous divisons la scène rectangulaire en k parties de surfaces égales, comme c'est illustré dans la figure 20 (pour $k=6$). Nous considérons les centres des rectangles résultants comme des points de référence principaux autour desquels nous formons les clusters.

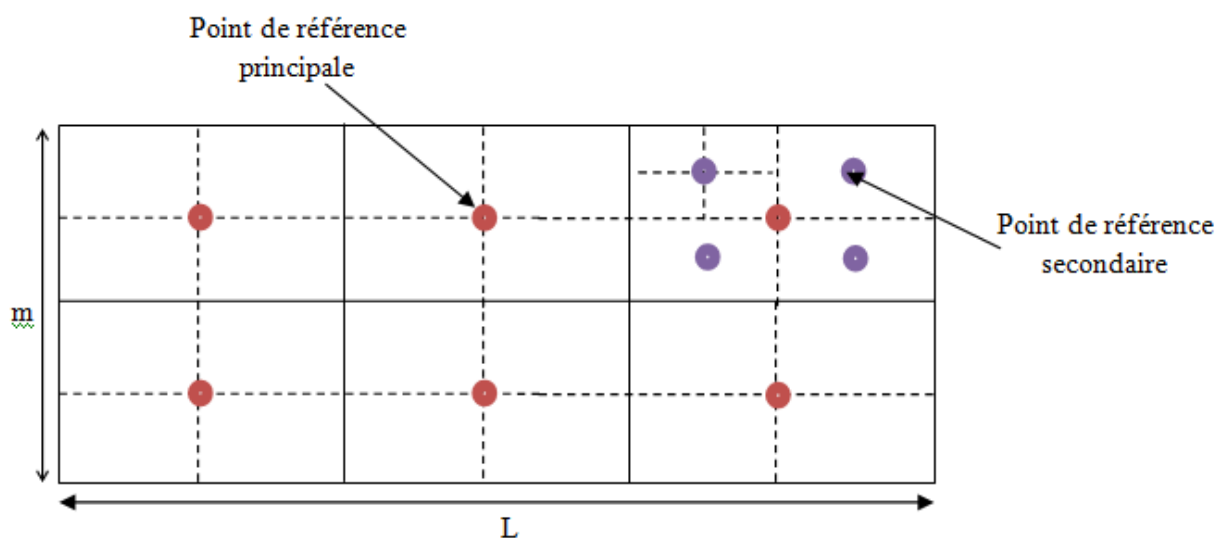


Figure 20. Points de référence principaux et secondaires

Les points de référence secondaires sont définis par les centres des parties résultantes d'une division de chaque rectangle en quatre (ou plus) parties égales.

Il nous reste à définir une méthode pour diviser la surface rectangulaire en zones équivalentes. Dans l'algorithme suivant, nous proposons la division du rectangle en parties égales en fonction du nombre de clusters voulu, par conséquent, trois cas de figures se présentent. Si le nombre de clusters k est un nombre premier alors nous ne pouvons pas diviser le rectangle en k parties. Nous le divisons alors en $k-1$ parties. Si k n'est pas un nombre premier, nous cherchons les diviseurs de k dans un ordre croissant, nous déterminons ensuite les deux diviseurs qui se suivent dont le produit est égale à k . Nous obtenons ensuite les k zones par la division de la longueur par le grand diviseur et la largeur par le petit (ou l'inverse). S'il n'existe pas de diviseurs successifs dont le produit est égal à k , nous cherchons alors le diviseur dont sa puissance de deux est égale à k .

Exemple: nous voulons diviser le rectangle en 24 parties, la listes des diviseurs est {1, 2, 3, 4, 6, 8, 12, 24}. Les deux diviseurs qui se suivent dont le produit est égal à 24, sont 4 et 6. Nous divisons alors la longueur par 6 et la largeur par 4.

Entrées :

k : nombre de clusters.

l : longueur de la scène.

m : largeur de la scène.

Programme

Si (k est premier) **alors**

Nous ré-exécutons l'algorithme pour $k=k-1$;

Sinon nous déterminons la liste des diviseurs de k dans l'ordre croissant $L = \{d_1, d_2, d_3, \dots, d_n\}$

Si (existe d tel que $d^2=k$) **alors**

Nous divisons la longueur et la largeur par d pour former les rectangles.

Sinon

On cherche d_i et d_{i+1} avec $k = d_i \times d_{i+1}$ et nous divisons la longueur par d_{i+1} et la largeur par d_i

Fin si

Fin si

Fin

Les coordonnées des centres des rectangles résultants désignés en tant que points de référence principaux sont données par les couples (x_i, y_j) avec

$$x_i = \frac{x_0}{2} + i \times x_0 \text{ avec } i = 0 \dots d_{i+1} - 1$$

$$y_j = \frac{y_0}{2} + j \times y_0 \text{ avec } j = 0 \dots d_i - 1$$

$$\text{Avec } x_0 = \frac{L}{d_{i+1}}, y_0 = \frac{M}{d_i}$$

En supposant que pour chaque point de référence principal nous définissons quatre points de référence secondaires (P_1, P_2, P_3, P_4), leurs coordonnées sont données par :

$$P_1 = (x_i + \frac{x_0}{4}, y_j + \frac{y_0}{4}), P_2 = (x_i - \frac{x_0}{4}, y_j + \frac{y_0}{4})$$

$$P_3 = (x_i + \frac{x_0}{4}, y_j - \frac{y_0}{4}), P_4 = (x_i - \frac{x_0}{4}, y_j - \frac{y_0}{4})$$

b) Deuxième méthode: Génération des points de référence basée zones

Dans cette méthode, nous essayons de créer un nombre de points de référence égale au nombre de clusters à former en utilisant la méthode k-MEANS [49]. Cette dernière divise des surfaces en zones de forme géométrique quelconques mais en taille assez homogènes. Les centres de ces zones sont définis comme points de références principaux.

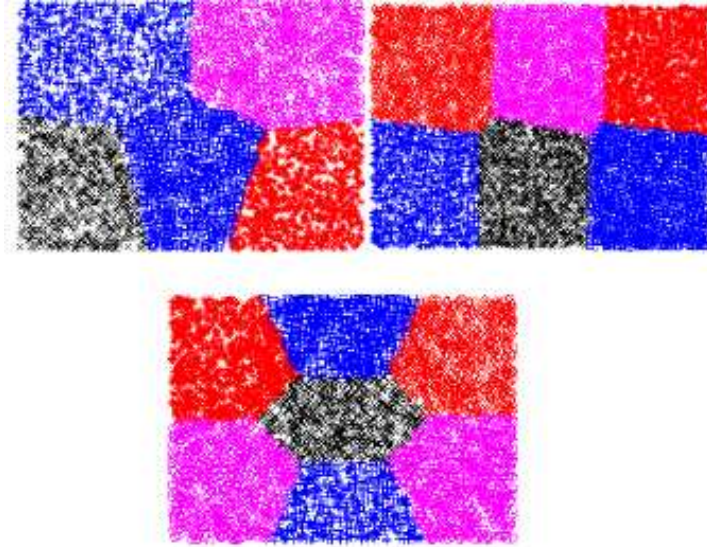


Figure 21. Exemple d'exécution de la méthode K-MEANS pour k=5,6 et 7 zones

Après la génération des points de référence principaux, nous procédons à la génération des points de référence secondaires. Pour cet effet, nous calculons la surface de notre scène (rectangle) et nous la divisons par le nombre de clusters à former. Pour une bonne couverture de

la scène, nous approximations les surfaces résultantes par des ellipses avec les mêmes centres que les zones résultantes et de rayons (a, b) calculés par les équations suivantes :

La surface du rectangle : $lm = k\pi ab \Rightarrow ab = \frac{lm}{k\pi}$

Pour que les ellipses aient la même allure que la scène initiale, nous devons avoir :

$$\frac{a}{b} = \frac{l}{m}$$

Avec :

- k : le nombre de clusters à créer.
- l : la longueur de la scène.
- m : la largeur de la scène.

Afin d'avoir une distribution homogène des points de référence secondaires, nous traçons à l'intérieur de ces ellipses d'autres ellipses de rayons a' , b' dans les quelles nos points de référence secondaires vont être définis. Ces rayons sont donnés par exemple comme suit :

$$a' = \frac{2}{3}a, b' = \frac{2}{3}b$$

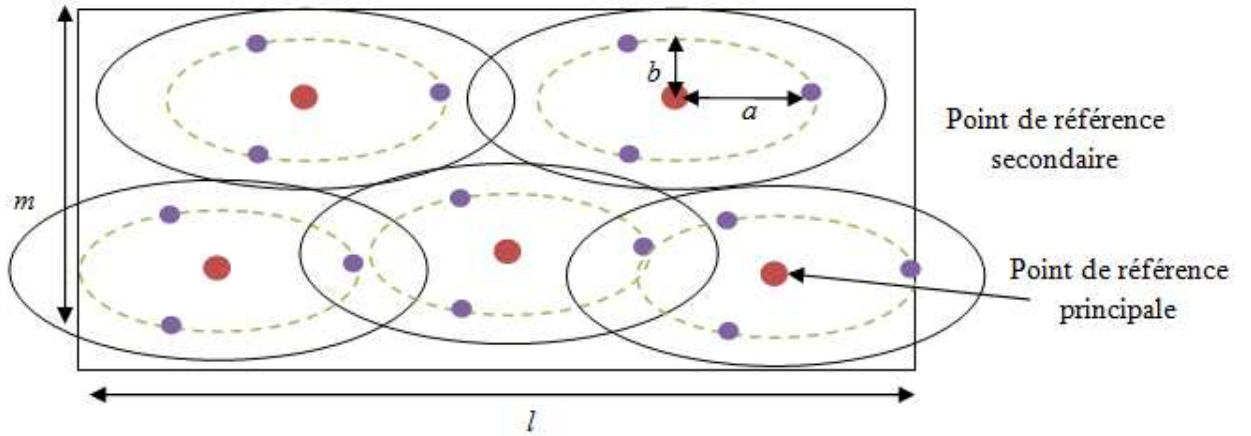


Figure 22. La répartition des points de référence par des ellipses

Les coordonnées des points de référence secondaires (x_p, y_p) sont données en fonction des coordonnées des points de référence principaux (x_i, y_i) et des rayons a' et b' par:

$$x_p = x_i + a' \cos \theta$$

$$y_p = y_i + b' \sin \theta$$

Où θ est l'angle entre l'axe horizontal et l'axe passant par le point de référence principal et le point de référence secondaire (à définir suivant le nombre et la disposition des points secondaires). Ceci en supposant que l'origine du repère est le point gauche bas de la scène.

4.2.2. Sélection des cluster-heads

Comme dans [16], nous allons élire les cluster-heads en se basant sur la distance qui sépare les nœuds de leur *NRP* (*Near Reference Point*). Afin d'attribuer une charge équitable aux nœuds capteurs par alternance du rôle cluster-head aux nœuds, nous introduisons un coefficient α qui va être multiplié par la distance qui sépare un nœud de son *NPR*. Ce coefficient est initialisé à 1 pour tous les nœuds. Une fois qu'un nœud est élu comme cluster-head, la valeur du coefficient est incrémentée dans le défavoriser ce nœud en tant que candidat au rôle cluster-head aux prochaines périodes. Après un certain nombre de périodes (à définir en fonction du nombre de clusters de nœuds), ce coefficient est réinitialisé à 1. Nous introduisons aussi l'énergie résiduelle des nœuds dans le critère de sélection des cluster-heads afin de favoriser les nœuds ayant plus d'énergie dans le cas d'égalité des distances séparant deux nœuds ou plus par rapport au *NPR*. Ainsi, chaque nœud calcule un poids par la formule suivante:

$$Poids(u) = \alpha * Distance(u/NPR) + (Emax - Energie_résiduelle(u))$$

Avec : $1 < \alpha < 2$ et *Emax* l'énergie maximale des nœuds capteurs

Les nœuds ayant les poids les plus petits dans leurs zones (avec le même *NPR*) sont élus en premier comme cluster-heads. En effet chaque nœud génère un délai d'attente inversement proportionnelle à ce poids. A l'expiration de ce délai, si le nœud n'a pas reçu un message *ADV_CH* depuis un autre nœud qui a le même *NPR* il diffusera un message *ADV_CH* pour déclarer son élection comme cluster-head.

4.2.3. Construction des clusters

Après la fin de la phase de sélection des cluster-heads, chaque nœud envoie un message *REQ_JOIN* au cluster-head. Ce message contient l'identificateur du nœud et sa position.

Après l'expiration d'un délai prédéfini pour la réception des messages *REQ_JOIN*, chaque cluster-head calcule le nombre de demandes d'adhésion. Si ce nombre est inférieur à un seuil maximal, le cluster-head envoie un message d'acceptation *ACCEPT_CH* à ces nœuds. Si ce nombre est supérieur à un seuil maximal, le cluster-head partage l'ensemble de nœuds sur

plusieurs nouveaux clusters en se basant sur leur nombre et leurs distances par rapport aux points de référence secondaires. Ensuite, le cluster-head envoie aux nouveaux cluster-heads un message *NEW_CH* pour les informer de leur désignation comme cluster-heads contenant la liste de leurs membres. Chaque nouveau cluster-head envoie un message d'acceptation d'adhésion contenant son identité à ses nœuds membres. Si le nombre des membres est inférieur à un seuil minimal, le cluster-head envoie un message aux nœuds pour qu'ils choisissent un autre cluster-head (le plus proche). Une fois les clusters définis, la phase de transmission des données commence.

Le détail du processus de construction des clusters est illustré par l'algorithme suivant:

Les structures utilisées

- Table *_ADV_CH*: une table sauvegardée au niveau de chaque nœud et contenant les informations des cluster-heads (id, distance...) ayant envoyé des messages *ADV_CH*.
- Table *Join*: une table sauvegardée au niveau des cluster-heads et contenant les informations des nœuds ayant envoyé un message *REQ_JOIN* à ce cluster-head.
- Table *Ref_Point*: une table sauvegardée au niveau de chaque nœud et contenant les coordonnées des points de référence principaux et secondaires.

Les messages utilisés

- *ADV_CH*: un message d'avertissement diffusé par les cluster-heads.
- *REQ_JOIN*: un message de demande d'adhésion envoyé par les nœuds aux cluster-heads.
- *REQ_JOIN_1*: un message de demande d'adhésion envoyé des nœuds au cluster-head après réception d'un message *NO_ACCEPT_CH*.
- *ACCEPT_CH*: un message d'acceptation d'adhésion envoyé par le cluster-head aux nœuds membres.
- *NEW_CH*: un message envoyé par le cluster-head aux nouveaux cluster-heads après sa disjonction.
- *ACCEPT_CH_1*: un message d'acceptation d'adhésion envoyé par les nouveaux cluster-heads aux nœuds membres.
- *ACCEPT_CH_2*: un message d'acceptation d'adhésion envoyé par le cluster-head aux nœuds qui envoient un message *REQ_JOIN_1*.
- *NO_ACCEPT_CH*: un message de refus d'adhésion envoyé par le cluster-head qui n'a pas assez de membres pour former un cluster avec ses membres.

Les fonctions utilisées :

- Ajout (table, u): une fonction qui extrait les informations sur les nœuds à partir des messages envoyés et les sauvegarde dans la table.
- Ré-affecte (Table_Join, Table_Ref_Point): une fonction qui réaffecte les nœuds dans les nouveaux clusters en se basant sur les positions de ces nœuds et les coordonnées des points de référence secondaires. Cette fonction calcule le centre de gravité du nuage formé par les nœuds et ordonne les points de référence secondaires suivant leurs distances relatives à ce centre de gravité. Ensuite, elle affecte les nœuds au nouveau cluster suivant les distances des nœuds à ces points de référence.

Algorithme

Entrées:

- Seuil_Max, Seuil_Min,
- Delay_ADV : délai d'attente des messages ADV_CH.
- Delay_AFFIL : délai d'attente d'acceptation d'adhésion
- Delay_JOIN : délai d'attente des messages de demande d'adhésion après l'envoi des messages d'avertissement ADV_CH.
- Delay_JOIN_2 : délai d'attente des messages de redemande d'adhésion.

Table_ADV_CH, Table_Join, Table_Ref_Point

Procédure nœud non cluster-head # pseudo code exécuté par les nœuds non cluster-heads

1. Tant que (Delay_ADV)

Si ((u reçoit un message ADV_CH d'un nœud v) alors

Ajoute (Table_ADV_CH, v)

Fin si

Fin Tant que

Envoyer REQ_JOIN au plus proche cluster-head.

2. Tant que (Delay_AFFIL)

Si (le nœud u reçoit un message ACCEPT_CH ou ACCEPT_CH_1 ou ACCEPT_CH_2 d'un cluster-head v) alors

$u \rightarrow \text{NodeCH} = v$

Fin si

Si (le nœud u reçoit un message NO_ACCEPT_CH) alors

Envoyer REQ_JOIN_1 au cluster-head le plus proche.

Fin si

Si (le nœud u reçoit un message NEW_CH) alors

Envoyer ACCEPT_CH_1 aux nœuds membres.

Fin si

Fin Tant que

Fin

Procédure nœud cluster-head # pseudo code exécuté par les cluster-heads

1. Envois *ADV_CH* à tous les nœuds

2. **Tant que** (*Delay_JOIN*)

Si (reçoit un *REQ_JOIN* d'un nœud *u*) **alors**

$CH \rightarrow Size = CH \rightarrow Size + 1$

 Ajoute (*Table_Join*, *u*)

Fin si

Fin tant que

3. **Si** ($Seuil_Min < CH \rightarrow Size < Seuil_Max$) **alors**

 Envoyer *ACCEPT_CH* aux nœuds sauvegardés dans *Table_Join*

Fin si

4. **Si** ($CH \rightarrow Size < Seuil_Min$) **Alors**

 Envoyer *NO_ACCEPT_CH* aux nœuds sauvegardés dans *Table_Join*

Fin si

5. **Si** ($CH \rightarrow Size > Seuil_Max$) **Alors**

 Ré-affecter (*Table_Join*, *Table_Ref_Point*)

 Le cluster-head envoie le message *NEW_CH* aux nouveaux cluster-heads

Fin si

6. **Tant que** (*Delay_JOIN_2*)

Si (reçoit un *REQ_JOIN_1* d'un nœud *u*) **alors**

 Ajoute (*Table_Join*, *u*)

 Envoyer le message *ACCEPT_CH_2* aux nœuds sauvegardé dans la table.

Fin si

Fin tant que

Fin

3.4.4. Maintenance des clusters

Pour faire face aux changements fréquents de la position des nœuds au sein d'un cluster une maintenance efficace est indisponible. Nous avons vu que les auteurs de LEACH-Mobile [23] proposent une méthode pour la maintenance des clusters qui engendre beaucoup de messages de contrôle, ce qui influe négativement les performances du réseau. Pour remédier à ce problème, nous améliorons cette méthode de maintenance des clusters et nous l'adoptons dans notre protocole. Pour ce faire, nous définissons une zone de confiance et une zone de doute

autour de chaque point de référence. Lorsqu'un nœud se trouve dans la zone de confiance, il considère que le cluster-head est joignable, c'est-à-dire que ce nœud appartient toujours à son cluster. S'il quitte cette zone et entre dans une nouvelle zone. Il doit alors s'assurer de son appartenance au cluster local par la réception de messages *REQ-DATA* depuis le cluster-head (voir figure 23). La maintenance se fait comme suit:

- Lorsqu'un nœud sort de la zone de confiance, il commence à envoyer des alertes au cluster-head dans les messages de données envoyés à ce dernier, en lui indiquant qu'il est en dehors de la zone de confiance. Lorsque le cluster-head reçoit cette alerte, il commence à envoyer dans les prochains slots réservés à ce nœud des messages *REQ-DATA* lui en indiquant qu'il est toujours dans sa portée.
- Si un nœud envoie deux alertes au cluster-head sans recevoir un message *REQ-DATA*, il se considère injoignable par ce cluster-head et il lance un processus de recherche d'un nouveau cluster-head.
- Si un cluster-head ne reçoit pas de données depuis un nœud durant deux frames successifs, il le considère injoignable et il le supprime de son schedule TDMA (inspiré de LEACH-Mobile [23]).

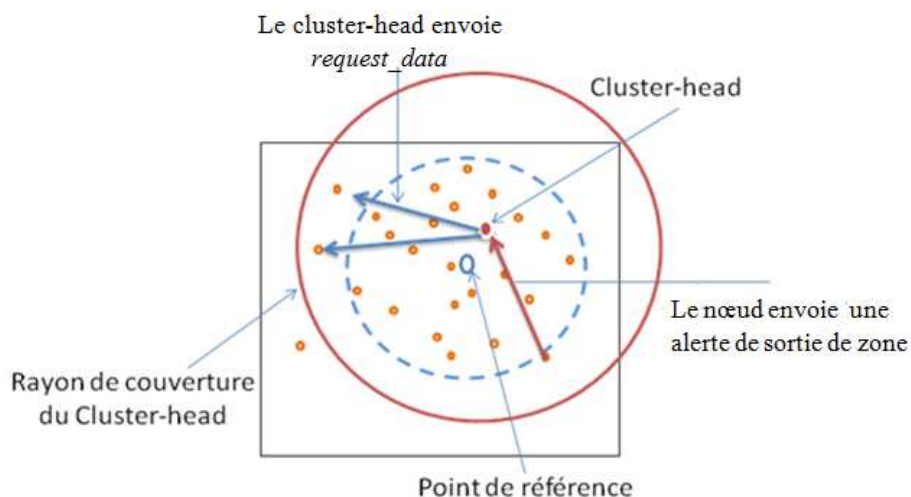


Figure 23. Maintenance des clusters

4.3. GMAC: (*Distributed Group Mobility Adaptive Clustering algorithm MWSNs*)

Ces dernières années, de la recherche scientifique porte un intérêt croissant aux réseaux de capteurs dont la mobilité des nœuds joue un rôle clé dans les performances des applications. Les modèles de mobilité des nœuds ont donc fait l'objet d'importantes interrogations par la

communauté des chercheurs. Tandis que la plupart des travaux réalisés dans ce domaine suppose que la mobilité des nœuds capteurs suit des modèles non réalistes et aléatoires comme dans SM [50], RDM [51], RWM [52], Ces modèles supposent que chaque nœud se déplace de façon aléatoire indépendamment des autres nœuds. Alors que, dans beaucoup d'applications, telles que les applications militaires et de secours, les déplacements des nœuds sont généralement faits en groupe, comme dans le cas d'un ensemble de soldats et engins militaires se déplaçant en groupes dans un champ de bataille (Figure 24).

Pour tenir compte de cette propriété de telles applications, nous définissons la métrique *MobilityGroup* qui reflète l'importance du déplacement en groupe d'un ensemble de nœuds. Plus la valeur de cette métrique est importante, plus la probabilité de maintien d'un nœud dans le groupe est grande. En se basant sur cette métrique nous définissons notre deuxième proposition GMAC; une technique de clustering distribué adapté à la mobilité des nœuds en groupe. De la même manière que notre première proposition, nous subdivisons la scène en zones, afin de fixer le nombre de clusters à créer et par conséquent maximiser la durée de vie du réseau de capteurs.

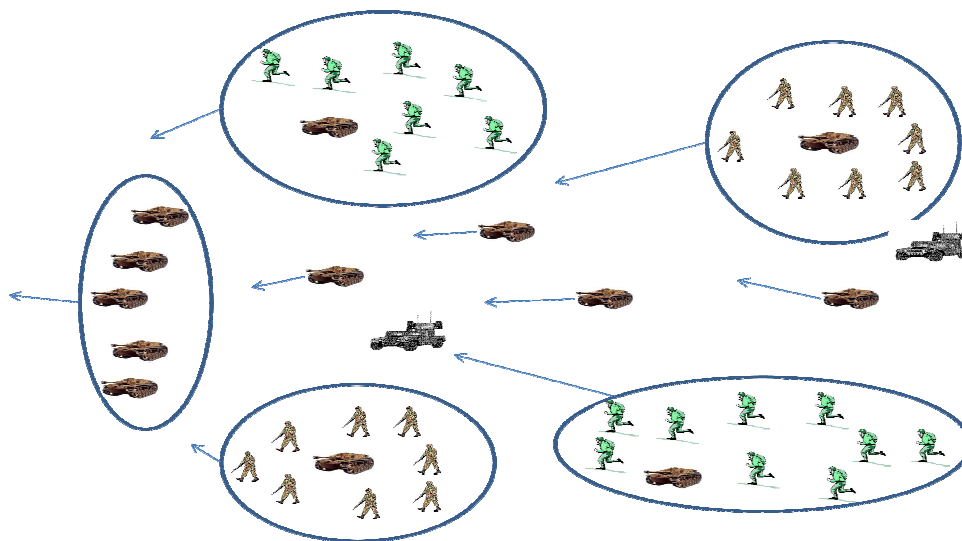


Figure 24. La mobilité en groupe dans les MWSNs (ex : application militaire)

4.3.1 Métrique de mobilité en groupe (*MobilityGroup*)

Dans cette partie, nous définissons la métrique de mobilité en groupe, qui reflète le fait qu'un nœud se déplace en groupe avec d'autres nœuds. Cette métrique représente, en fait, trois indicateurs :

- la taille du groupe, donnée par le nombre de nœuds se déplaçant ensemble,

- la compacité ou densité du groupe, exprimée par les distances qui séparent les nœuds constituant le groupe
- la connectivité dans le groupe, donnée par le nombre de lien existants dans le groupe.

Pour prendre en considération les caractéristiques de ces groupes, plus particulièrement l'alternance des nœuds entre les différents groupes de mobilité, nous définissons deux paramètres *HistoricalGroup(u)* et *PredictionGroup(u)*. Le premier paramètre détermine les groupes de mobilité antérieurs dans le voisinage d'un nœud à travers la connaissance des positions physiques enregistrées dans le passé. Le deuxième paramètre définit les groupes par prédiction de la position des nœuds voisins dans le futur.

HistoricalGroup(u) est une projection de la métrique *MobilityGroup* dans le passé. Il est calculé par chaque nœud en fonction des informations de topologie (voisinage direct) sauvegardées dans le passé. Pour se faire, chaque nœud sauvegarde la liste de ses voisins à un saut (direct) périodiquement $\{T_0 \dots T_{i-3}, T_{i-2}, T_{i-1}\}$. A l'instant de l'invocation du calcul de cette métrique, chaque nœud utilise la dernière liste des voisins sauvegardé.

PredictionGroup(u) est une projection de la métrique *MobilityGroup* dans le futur. Nous utilisons les informations de topologie dans le présent (vitesse, direction) pour prédire la liste des voisins dans le futur. Pour se faire, nous estimons pour chaque nœud sa position et celles de ses voisins dans le futur par une prédiction de la position ultérieure. Les nœuds qui resteront dans la portée de communication de ce nœud dans le futur vont être considérés comme des voisins.

Le paramètre *HistoricalGroup(u)* et *PredictionGroup(u)* à l'instant T sont donnés par:

$$HistoriqueGroup(u) = Cr \times cardinal(HG(u)) - \sum_{v \in HG(u)} distance(u, v)$$

$$PredictionGroup(u) = Cr \times cardinal(PG(u)) - \sum_{v \in PG(u)} distance(u, v)$$

$$HG(u) = N(u)_T \cap N(u)_{T_{i-1}}$$

$$PG(u) = N(u)_T \cap N(u)_{T_{pred}}$$

$HG(u)$ est la liste résultante de l'intersection entre la liste des voisins directs $N(u)_T$ à l'instant T et la dernière liste des voisins sauvegardés $N(u)_{T_{i-1}}$ dans le passé.

$PG(u)$ est la liste résultante de l'intersection entre la liste des voisins directs $N(u)_T$ à l'instant T et la liste des voisins prédits $N(u)_{T_{pred}}$ à T_{pred} .

Cr est la portée de communication (la distance maximale entre deux voisins).

La métrique de mobilité en groupe peut être évaluée par l'intersection de $HG(u)$ et $PG(u)$, ce qui permet de formaliser cette métrique par:

$$MobilityGroup(u) = Cr \times cardinal(MG(u)) - \sum_{v \in MG(u)} distance(u, v)$$

$$MG(u) = PG(u) \cup HG(u)$$

Pour avoir une vision plus étendue de cette métrique, nous la généralisons en définissant la métrique de mobilité en groupe à n saut ($MobilityGroup_n(u)$). Cette métrique est définie comme suit :

(a) La métrique de mobilité en groupe à 1 saut:

$$MobilityGroup_1(u) = MobilityGroup(u)$$

(b) La métrique de mobilité en groupe à 2 sauts:

$$MobilityGroup_2 = \sum_{v=0}^{K-1} MobilityGroup(v)$$

Avec: $v \in MG(u)$ et $k=cardinal(MG(u))$

(c) La métrique de mobilité en groupe à n sauts:

$$MobilityGroup_n = \sum_{v=0}^{K-1} MobilityGroup(v)$$

Avec:

$$v \in \{MG(u) \cup MG(u_1)/u_1 \in MG(u) \cup MG(u_2)/u_2 \in MG(u_1) \dots \cup MG(u_{n-2})/u_{n-2} \in MG(u_{n-3})\},$$

$$k = \sum_{i=0}^{n-2} cardinal(MG(u_i)),$$

$$MG(u_0) = MG(u).$$

Exemple : dans la figure 25.a le nœud u a plusieurs voisins, dont quatre nœuds se déplacent avec lui dans le même groupe de mobilité. Ces quatre nœuds (u_1, u_2, u_3, u_4) sont colorés en rouge. La métrique de mobilité en groupe à un saut de se nœud est égale à 5. Dans la figure 25.b nous montrons par les nœuds colorés en rouge les groupe de mobilité des nœuds (u_1, u_2, u_3, u_4). Par exemple le nœud u_2 a 6 nœuds dans son groupe de mobilité. Alors, la métrique de mobilité en groupe à deux sauts de u est donnée par la somme des métriques de mobilité des nœuds (u_1, u_2, u_3, u_4). Elle est égale à $(3+6+3+4=16)$.

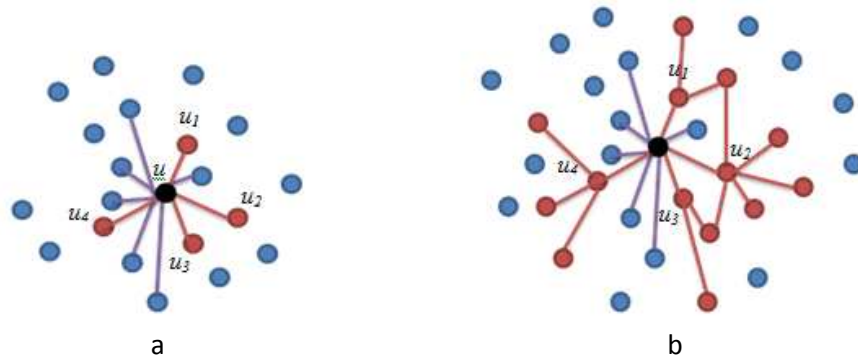


Figure 25. Exemple d'évaluation du paramètre *GroupMobility*

Remarque: la métrique *MobilityGroup* que nous avons défini dépend du nombre de nœuds dans le groupe et les distances entre ces nœuds. Nous pouvons avoir des cas où un groupe de mobilité avec une taille petite a une valeur plus grande qu'un autre de taille plus grande. Cela va nous permettre de favoriser les groupes dont les nœuds sont proches.

Pour prédire la liste des voisins d'un nœud dans le futur, nous estimons la position de ce nœud et ses voisins à l'instant T_{pred} . Les nœuds qui restent inclus dans sa zone de couverture vont appartenir à cette liste. Dans ce qui suit, nous exposons la méthode de prédiction d'une position future d'un nœud et la détermination de l'intervalle de temps auquel appartient l'instant de prédiction T_{pred} .

a) Prédiction de la position d'un nœud

Pour prédire la position des nœuds dans le futur, nous définissons un estimateur de position qui prend en considération la vitesse de déplacement des nœuds et leur direction. Dans notre proposition, nous supposons que les nœuds se déplacent suivant une trajectoire rectiligne (Figure 26) avec une vitesse fixe pendant des périodes de temps (Δt). Cela va nous permettre de prédire la position des nœuds à l'instant $T_{pred}=T+\Delta T$ par les équations suivantes:

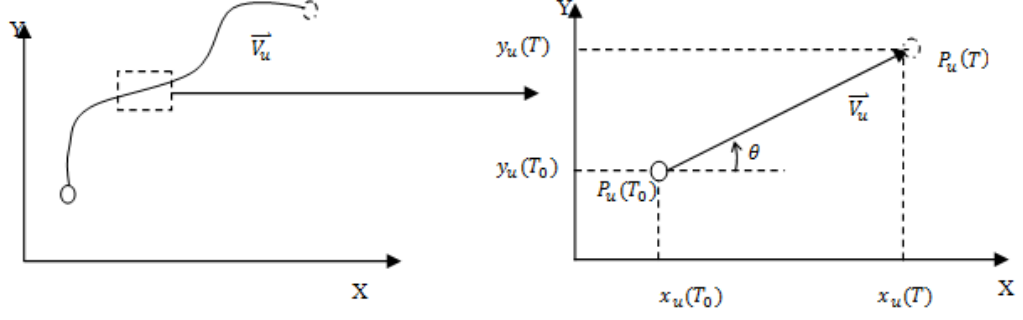


Figure 26. L'estimation de la position dans GMAC

$$P_u(T_{pred}) = \vec{V}_u \times \Delta T + P_u(T)$$

$$x_u(T_{pred}) = V_u \times \cos\theta \times \Delta T + x_u(T)$$

$$y_u(T_{pred}) = V_u \times \sin\theta \times \Delta T + y_u(T)$$

$$\text{Avec : } \Delta T = T_{pred} - T$$

θ : L'angle entre l'axe des abscisses et le vecteur de vitesse $\vec{V}_u$

Pour déterminer le vecteur de vitesse $\vec{V}_u$ pour un nœud u , nous calculons la valeur de la vitesse V_u et sa direction. La vitesse d'un nœud est calculée par la formule suivante:

$$V_u = \frac{\sqrt{((x_u(T) - x_u(T_{i-1})))^2 + ((y_u(T) - y_u(T_{i-1})))^2}}{T - T_{i-1}}$$

L'angle θ qui définit la direction du nœud est calculé par :

$$\tan \theta = \frac{y_u(T) - y_u(T_{i-1})}{(x_u(T) - x_u(T_{i-1}))}$$

b) Définition de la valeur de T_{pred}

Le clustering est un processus qui s'exécute généralement de façon périodique. Chaque période comporte deux phases : (a) une phase de construction des clusters (*set-up*) et (b) une phase pour la transmission de données (*steady-state*). Dans notre proposition, la valeur de T_{pred} dans laquelle nous cherchons à prédire la position des nœuds va être fixée par l'administrateur, en se basant sur les travaux faits dans [16] nous pouvons estimer que cette valeur peut prendre une valeur entre l'instant de la fin de la phase de formation des clusters et le milieu de la phase

de transmission de données, comme indiqué dans la Figure 27. Le choix de cet instant dépend de la vitesse des nœuds et le temps de la phase de transmission de données. En effet, si la vitesse des nœuds est grande, cet instant doit être plus proche à l'instant de la fin de la phase de sélection des cluster-heads.

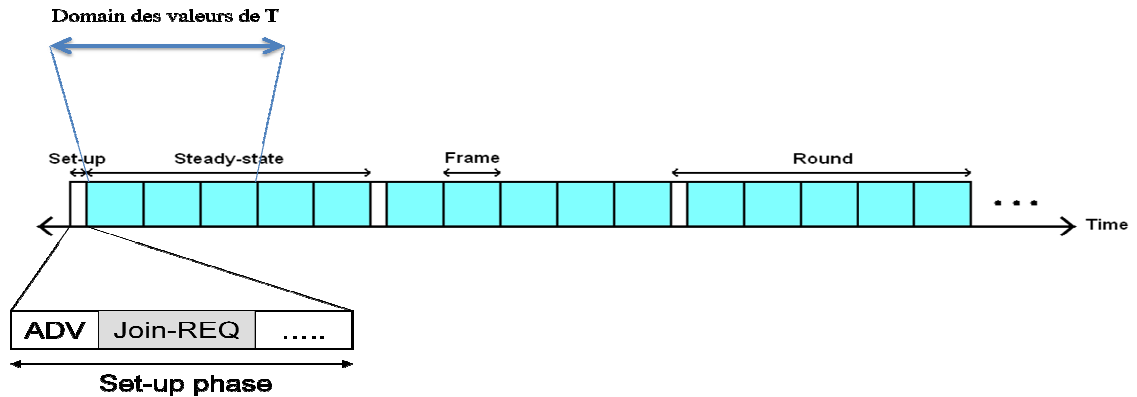


Figure 27. Les valeurs possibles de T_{pred}

4.3.2. Sélection des cluster-heads

Pour notre technique de clustering, nous proposons une méthode d'élection des cluster-heads qui permet de générer des clusters stables en se basant la métrique de mobilité en groupe et éventuellement l'énergie résiduelle, Pour qu'un nœud u devienne un cluster-head, il doit être avantagé par un déplacement dans un groupe de nœuds important et une énergie résiduelle suffisante. Les étapes de sélection des cluster-heads sont:

- Chaque nœud envoie un message contenant sa position, sa vitesse et sa direction vers ses voisins directs.
- Chaque nœud détermine la zone à laquelle il appartient en se basant sur sa position et les coordonnées des zones. Pour chaque zone, nous créerons un seul cluster.
- Chaque nœud calcule son poids en fonction de l'énergie résiduelle et la métrique de mobilité en groupe.
- Chaque nœud candidat détermine un délai d'attente, inversement proportionnel au poids calculé. Si ce délai s'écoule avant la réception d'une déclaration d'une élection d'un autre cluster-head dans sa zone alors ce nœud déclare sa sélection comme cluster-head en diffusant un message *ADV_CH* contenant l'identifiant du nœud et l'instant d'envoi de ce message.

Le détail de la sélection des cluster-heads est donné dans l'algorithme suivant, en supposant que chaque nœud possède une structure de la forme *Node (Id, Position, CH,...)*:

Entrées

α, β : Paramètres définis par l'administrateur.

Delay_CHE : Délai total pour l'élection des cluster-heads

messages_réçu : Liste des messages *ADV_CH* reçus.

Programme

Le nœud *u* détermine sa zone en fonction de sa position.

$Poids(u) = \alpha * MobilityGroup(u) + \beta * Energie-Rés(u)$

Delay(u) = fonction (*poids(u)*)

Tant que (*Delay_CHE* non expiré)

Si (*Delay* expiré && *messages_réçu* = vide) **alors**

Diffuser un message *ADV_CH*

Insérer le message dans la liste *messages_réçu*

Fin si

Si (reçoit un message *ADV_CH* d'un autre nœud de la même zone) **alors**

Insérer le message dans la liste *messages_réçu*

Fin si

Fin tant que

Choisir le premier nœud *v* qui a envoyé de message *ADV_CH*.

$u \rightarrow CH=v$

Fin

4.3.3. Définition des zones :

Dans cette proposition, nous divisons le champ d'intérêt en zones logiques. Chaque zone est représentée par un rectangle de surface $l \times m$ et identifié par un couple unique (Z_x, Z_y), chaque nœud peut savoir à quelle zone il appartient en utilisant les formules suivantes:

$$Z_x = \left\lceil \frac{x}{l} \right\rceil$$

$$Z_y = \left\lceil \frac{y}{m} \right\rceil$$

Où

- x et y sont les coordonnées cartésiennes du nœud.

- l et m sont les dimensions de la scène.

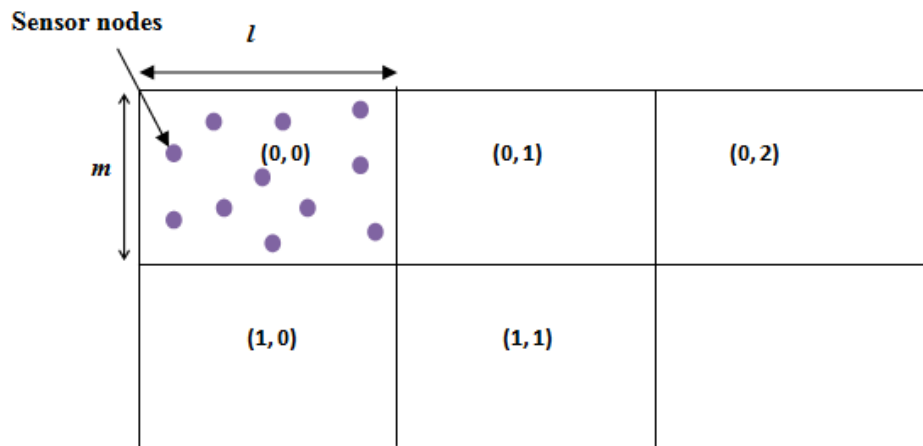


Figure 28. La division de la scène en zones

4.3.4. Construction des clusters

A la fin de la phase d'élection des cluster-heads, chaque nœud non cluster-head envoie ensuite une demande d'adhésion *JOIN_REQ* à son cluster-head. A la fin de la phase construction des clusters, chaque cluster-heads crée un schedule TDMA (un processus d'ordonnancement des slots) et envoie à ses membres leurs slots de temps correspondants. Puis la phase de transmission des données (*steady-state*) débute.

Conclusion

Dans ce chapitre, nous avons présenté une description détaillée de nos protocoles DACE-L et GMAC, en exposant leurs principes de fonctionnement et les mécanismes mis en œuvre pour structurer le réseau en clusters et communiquer les données capturées à la station de base. Dans le protocole DACE-L, nous avons amélioré le protocole ACE-L présenté dans [16], pour qu'il s'adapte à la variation de la densité des nœuds capteurs dans le réseau au cours du temps. Ce protocole a comme objectif de générer des clusters homogènes en taille afin d'équilibrer la charge dans le réseau et ainsi prolonger la durée de vie du réseau. Nous avons aussi proposé une méthode pour la maintenance des clusters générés par le protocole DACE-L et pour minimiser le nombre de paquets perdus à cause de la mobilité des nœuds. Dans GMAC nous avons définis une nouvelle métrique pour la sélection des cluster-heads en se basant sur la mobilité en groupe des nœuds capteurs c'est-à-dire la tendance des capteurs à se mouvoir en groupe dans de nombreuses applications.

Chapitre 5

Simulations et résultats

Introduction

Après avoir détaillé la conception de nos propositions, nous présentons dans ce chapitre les paramètres utilisés pour l'évaluation des performances de nos protocoles et ceux de l'environnement de simulation. Puis, nous discuterons les résultats obtenus. Ces résultats sont comparés à ceux obtenus par les protocoles LEACH et ACE-L proposés respectivement dans [10] et [16]. La raison du choix de LEACH comme protocole de comparaison s'explique par le fait qu'il est considéré comme protocole de référence dans le domaine du routage hiérarchique et comme un standard de comparaison pour les protocoles de clustering publiés dans la littérature. Son efficacité à remplir les objectifs espérés d'un protocole de clustering a motivé son utilisation dans la comparaison des performances de la majorité des protocoles de clustering proposés. ACE-L est un protocole conçu pour les réseaux de capteurs mobiles, il est basé sur l'utilisation de l'information de localisation des nœuds capteurs pour élire les cluster-heads. Ces caractéristiques ont motivé son utilisation comme protocole de base pour la proposition du protocole DACE-L. Il est évident de le considérer comme référence pour l'évaluation de DACE-L.

5.1. Environnement de simulation

Nous avons réalisé plusieurs simulations pour évaluer les performances de nos propositions DACE-L et GMAC en termes de consommation d'énergie, de durée de vie du réseau, du nombre de paquets reçus par la station de base et du nombre de clusters formés dans chaque période. Puis, nous avons comparé les résultats obtenus aux performances d'ACE-L et de LEACH. Pour implémenter les algorithmes proposés, nous avons utilisé le simulateur NS2 [56]. Comme la mobilité est prise en considération dans l'exécution des protocoles, nous avons utilisé les modèles RWM [52] et RPGM [53] pour gérer la mobilité des nœuds. Dans le premier modèle, à chaque intervalle de temps les nœuds choisissent leurs directions et leurs vitesses de déplacement de façon aléatoire. Dans le second modèle les nœuds sont organisés en groupes, chaque groupe est représenté par un point de référence qui se déplace suivant une trajectoire prédéterminé, les autres nœuds du groupe se déplacent avec une variation aléatoire de la vitesse

et de la direction par rapport à celles du point de référence. Avant de présenter les performances de nos contributions et leur comparaison aux autres techniques, nous définissons les différentes mesures et les paramètres significatives pour cette évaluation.

5.1.1 Description de l'environnement de développement

Afin d'implémenter nos protocoles et de valider leurs conceptions, le choix d'un environnement extensible qui procure des résultats proches de la réalité est nécessaire. Actuellement une multitude de simulateurs de réseaux sans fil sont déployées dans le monde informatique, on peut citer ; GloMoSim, JiST/SWANS, GTSNetS, Opnet et NS.

a) NS (Network Simulator)

NS est un simulateur en libre accès destiné aux réseaux filaires, sans fil et satellitaires, à événements discrets. Il est considéré comme l'un des outils de simulation les plus populaires et les plus utilisés dans le domaine académique. NS utilise le langage OTCL (*Object Tool Comand Language*) pour le paramétrage de l'environnement de simulation. A travers ce langage l'utilisateur définit la topologie du réseau, les caractéristiques des liens physiques, les protocoles de communication utilisés, et les communications qui ont lieu.

La simulation doit être d'abord saisie sous forme d'un script TCL que NS va exécuter. Un fichier trace qui contient les résultats de la simulation est généré ; ce fichier va contenir des informations sur le trafic ainsi que sur la mobilité des nœuds et leur niveau énergétique. Chaque fichier trace devra être traité par la suite pour en tirer les paramètres à étudier ; l'une des manières les plus courantes consiste à écrire un script Perl, awk ou des petites applications en langage compilé.

Le choix du simulateur NS-2.34 est motivé par les raisons suivantes :

- Il s'agit d'un simulateur largement utilisé dans le milieu académique et qui a réussi à s'imposer au fil des années comme étant le simulateur non commercial le plus populaire. Il possède un manuel d'utilisation et une Mailing-List très fréquentée ainsi qu'une documentation abondante.
- C'est un simulateur open source qui offre la possibilité d'incorporer des modules supplémentaires. Le modèle des réseaux dans NS-2.34 est riche de supports très utiles tels que le support de calcul mathématique (générateur de nombre aléatoires,... etc) et le modèle de consommation d'énergie.

b) Architecture de NS

NS est implémenté en C++ et en OTCL (*Objet oriented Tool Comand Language*). Le langage C++ sert à décrire le fonctionnement interne des composants de la simulation. Quant au langage OTCL, il fournit un moyen flexible et puissant de contrôle de la simulation comme la configuration du réseau et le déclenchement d'événements. NS se compose de deux éléments : un interpréteur et un moteur de simulation : L'interpréteur permet à l'utilisateur de créer et de paramétrer le modèle de simulation ; le moteur effectue les différentes tâches liées à l'exécution du modèle de simulation construit par l'utilisateur via l'interpréteur.

Le package NS inclut une hiérarchie de classe compilée (C++) et une hiérarchie de classe interprétée (OTcl). Ces deux hiérarchies sont étroitement liées ; quand l'utilisateur crée un nouvel objet interprété (correspond à la hiérarchie OTcl), cet objet sera cloné en un objet compilé (correspond à la hiérarchie C++). Les deux classes les plus importantes pour établir la correspondance entre les objets compilés et les objets interprétés sont:

- La classe TCLObject : c'est la classe racine de la hiérarchie compilée. Cette classe contient une fonction « command () » qui est utile pour ajouter des commandes à l'interpréteur : Créer une commande OTcl puis la lier à une méthode d'une classe de la hiérarchie compilée.
- La classe TCLClass est une classe qui intervient lors de l'implémentation d'un nouveau protocole. Elle contient deux fonctionnalités:
 - Compléter la hiérarchie interprétée pour correspondre à la hiérarchie compilée du nouvel objet (protocole).
 - Offrir des méthodes pour faire appel au nouvel objet (protocole) via l'interpréteur.

5.1.2. Définition du système

L'objectif de nos travaux est de proposer une méthode de clustering efficace en termes de consommation d'énergie pour les réseaux de capteurs mobiles après avoir étudié les principales techniques de clustering existantes. Le système étudié consiste en un réseau de capteurs mobiles déployés dans une zone géographique bidimensionnelle avec une station de base fixe. Tous les nœuds suivent un modèle de mobilité prédéterminé (RWM, RPGM) [52, 53]. Les paramètres affectant notre système sont :

Les paramètres liés au réseau

- La nature de la surface: la zone d'intérêt est considérée sous forme rectangulaire.

- Le nombre de nœuds déployés.

Les paramètres liés aux modèles de mobilité

Notre étude se limite à deux types de modèles; RWM et RPGM. La génération d'un modèle de mobilité nécessite la fixation des paramètres liés à chaque modèle comme la vitesse maximale des nœuds, le nombre de groupe...etc.

Les paramètres liés aux nœuds capteurs

Pour toutes les simulations, les nœuds capteurs sont considérés mobiles et homogènes. Ils ont le même rayon de couverture et de transmission et les mêmes spécificités énergétiques (énergie initiale, énergie consommée en réception et en émission). La station de base est fixe, loin de la zone d'intérêt et dispose de plus d'énergie que les autres nœuds.

Les paramètres liés au modèle d'énergie

La consommation d'énergie dans les réseaux sans fil et en particulier dans les réseaux de capteurs concerne essentiellement les communications: émission et réception de message.

a) La consommation d'énergie en émission

L'énergie consommée par un émetteur peut être subdivisée en deux parties [54] : une partie consiste à générer des signaux RF , qui dépendent de la distance d , séparant l'émetteur du récepteur, de la modulation choisie et de la puissance de transmission ϵ_{Tx} . La seconde partie est due aux composants électroniques nécessaires pour la synthèse de fréquence, la conversion de fréquence et ainsi de suite. La puissance de transmission ϵ_{Tx} générée par l'amplificateur du l'émetteur est supposée connue et constante dans notre contexte. En plus de la puissance d'amplification, d'autres circuits doivent être activés lors d'une transmission. Cette puissance est représentée par ϵ_{Elec} . Ainsi, l'énergie nécessaire pour émettre un paquet de taille k est donnée comme suit:

$$E_{Tx} = \epsilon_{Elec} \times k + \epsilon_{Amp} \times k \times d^2$$

b) La consommation d'énergie en réception

L'énergie E_{Rx} exigée pour recevoir un paquet de taille k bits correspond à l'énergie nécessaire pour préparer les composants électroniques à une réception. Elle est similaire à celle de l'activation des composants électroniques au cours d'une transmission. Cette énergie est donnée par :

$$E_{Rx} = \epsilon_{Elec} \times k$$

Dans le cas des protocoles de clustering, nous ajoutons l'énergie nécessaire à la fusion et l'agrégation de données E_{fd} au niveau des cluster-heads. Donc, pour évaluer l'énergie totale

consommée par un nœud capteur, il suffit de connaître l'énergie des émissions, des réceptions et de l'agrégation de données. Cette énergie est donnée par :

$$E(t) = E_{Tx}(t) + E_{Rx}(t) + E_{fd}$$

5.2. Exemple d'application

Pour expliciter les techniques de clustering proposées, nous réalisons une exécution des protocoles ACE-L, DACE-L et GMAC pour un réseau de petite taille et nous discutons les résultats obtenus. Pour se faire, nous générons automatiquement une topologie de 20 nœuds suivant le modèle RPGM en utilisant l'outil BonnMotion [55]. Cette topologie est composée de deux groupes de dix nœuds pour chacun, déployés dans une scène de **200m × 200 m** composée de quatre zones Z1, Z2, Z3 et Z4 de taille égale. Les nœuds générés suivent le scénario suivant: Initialement tous les nœuds sont situés dans la zone 1 sauf le nœud 16 placé dans la zone 2 comme illustré dans la figure 29. Durant les premières quarante secondes, les nœuds du premier groupe (4.5.0.6.7.1.18.3.8.2.9) se déplacent vers la zone 2 avec une vitesse maximale de 5 m/s, alors que les nœuds du second groupe (13.11.14.12.15.10.19.16.17.18) restent stationnaires.

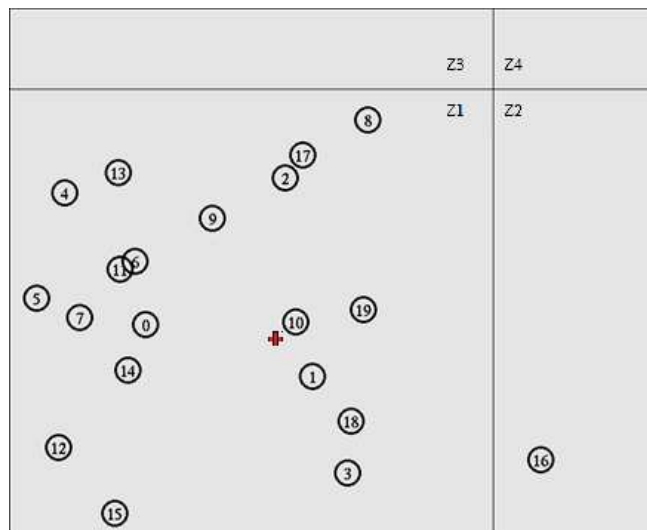


Figure 29. Aperçu de la topologie lors du déploiement du réseau

Les paramètres utilisés pour le modèle de mobilité RPGM et ceux de la simulation des protocoles sont donnés dans le tableau 2.

	ACE-L	DACE-L	GMAC
Nombre de Groupes	2		
Nombre de nœuds	10 nœuds pour chaque groupe		

Diamètres des groupes	50 m		
Vitesse maximale des nœuds	5 m/s		
Nombre de points de référence	4 points de coordonnées :(50,50),(50,150),(150,50),(150,150)		
Porté de communication			30 m
Seuil maximal	7 nœuds		
Seuil minimal	1 nœud		
Le coefficient α		1.5	

Tableau 2. Paramètres utilisés dans l'exemple

Résultats d'exécution des protocoles ACE-L, DACE-L et GMAC

Nous suivons d'exécution des protocoles ACE-L, DACE-L et GMAC pendant la deuxième période (20s-40s). La figure 30 illustre la topologie du réseau obtenu par l'outil NAM au début de cette période. Nous avons extrait les différentes données topologiques du réseau nécessaires à l'exécution de ces protocoles telles que: la distance qui sépare un nœud de son *NRP*, la métrique *MobilityGroup* à 1-saut et l'ensemble des nœuds formant les groupes de mobilité de chaque nœud. Ces données sont résumées dans le tableau 3.

Temps	ID	La métrique	$\sum$ Distances/u	Distance/NPR	Groupe
t=20	0	108,00	161,998115	8,94427191	0,5,6,7,9,10,11,14,19
t=20	1	72,46	77,5394755	26,1062564	1,3,10,18,19
t=20	2	64,95	55,05322	39,9620906	2,8,9,17
t=20	3	59,34	60,6640895	42,1073614	1,3,16,18
t=20	4	81,39	158,605055	39,4153619	4,5,6,7,9,11,13,17
t=20	5	117,76	122,237661	31,3889624	0,4,5,6,7,11,13,14
t=20	6	117,13	212,867337	19,7841291	0,4,5,6,7,9,10,11,13,17,19
t=20	7	120,58	149,424202	21,5848958	0,4,5,6,7,10,11,13,14
t=20	8	41,00	18,9982449	57,9902069	2,8
t=20	9	91,14	148,85908	25,7346685	0,2,4,6,9,10,17,19
t=20	10	98,60	141,402332	0,85358678	0,1,6,7,9,10,18,19
t=20	11	107,22	132,779648	35,6985704	0,4,5,6,7,11,13,14
t=20	12	52,03	37,9747432	52,1007291	12,14,15
t=20	13	69,82	110,175185	46,5478169	4,5,6,7,11,13
t=20	14	74,44	135,561105	33,6286036	0,5,7,11,12,14,15
t=20	15	43,41	46,586331	51,8299952	12,14,15
t=20	16	41,08	18,9206912	55,4482314	3,16
t=20	17	73,23	76,7672601	34,7034888	2,4,6,9,17
t=20	18	64,45	85,5531276	23,7432241	1,3,10,18,19
t=20	19	78,84	131,161644	16,5948994	0,1,6,9,10,18,19

Tableau 3. Données recueillies au début de la deuxième période

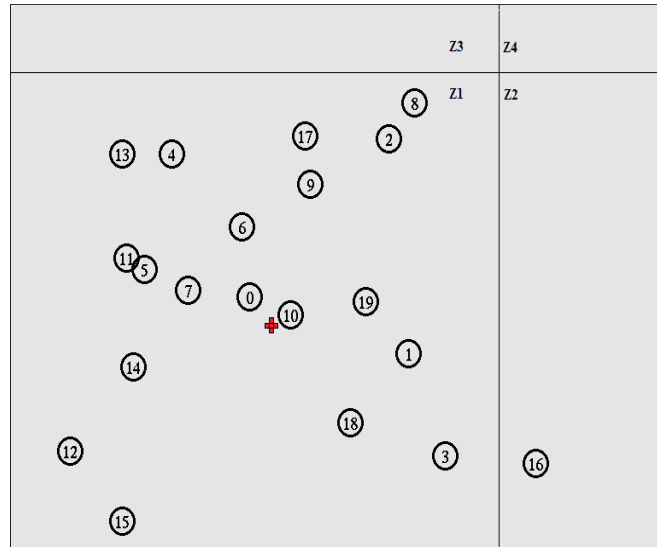


Figure 30. Aperçu de la topologie au début de la deuxième période à l'aide de l'outil NAM

A la fin de cette période nous avons obtenu les résultats d'exécution des protocoles tels que les cluster-heads sélectionnés pour chaque zone, la taille des clusters et les membres de chaque cluster.

ACE-L: Cette technique crée deux clusters dans les zones: 1 et 2. Les nœuds les plus proches des points de référence sont choisis comme cluster-heads (10: 0,85m/NRP1 et 16: 56,66m/NRP2). L'adhésion aux clusters se fait à base de la distance qui sépare les nœuds des cluster-heads, ce qui explique la variation de la taille des clusters. Le détail des clusters formés est illustré dans le tableau 4.

Temps	Cluster-heads	Zone	taille clusters	Membres
21.203944	10	1	18	{14 2 18 12 9 7 5 0 13 15 4 19 11 6 1 8 17 10}
21.203944	16	2	2	{3 16}

Tableau 4. Exécution d'ACE-L

GMAC: Cette technique crée deux clusters. Les nœuds qui ont la plus grande métrique *MobilityGroup* dans chaque zone sont choisis comme cluster-heads (11:142,18/NRP1 et le nœud 16:30/NRP2). Comme dans ACE-L les nœuds non cluster-head choisissent leurs cluster-heads selon la distance qui les séparent de ces derniers. Le tableau 3 indique le détail des clusters formés.

Temps	Cluster-heads	Zone	taille clusters	Membres
21.203944	7	1	16	{12 13 11 17 10 15 5 14 2 8 0 19 4 9 6 7}
21.203944	16	2	4	{3 18 1 16}

Tableau 5. Exécution de GMAC

DACE-L: Cette technique génère comme les deux techniques précédentes, deux clusters. Elle utilise les distances séparant les nœuds de leur NPR pour élire les cluster-heads. Cette distance est multipliée par un coefficient ($\alpha = 1.5$) une fois un nœud est choisis comme cluster-head. Dans cette période le nœud 10 est le plus proche au premier point de référence. Cependant, ce nœud est déjà choisis dans la première période ce qui fait que son poids sera augmenté en le multipliant par α . Par conséquent, c'est le nœud 0 qui est choisi comme cluster-head. Ce cluster-head reçoit un nombre de demandes d'adhésions qui dépasse la taille maximale (7), cela déclenche la subdivision de ce cluster en d'autres clusters de tailles plus petites. Les résultats obtenus de l'exécution de DACE-L sont donnés dans le tableau 6.

Temps	CH	Zone	Taille clusters	Type CHs	Membres
21.203944	0	1	18	CH	{4 12 2 5 17 9 11 15 6 0 8 1 19 18 7 14 10 13}
21.203944	12	1	6	CH fils	{12 15 0 18 7 14 10}
21.203944	17	1	7	CH fils	{2 17 9 6 8 1 19}
21.203944	4	1	4	CH fils	{13 4 5 11}
21.203944	16	2	2	CH	{3 16}

Tableau 6. Exécution de DACE-L

A partir des résultats obtenus sur cet exemple illustratif, nous pouvons conclure que nos protocoles fonctionnent suivant les objectifs que nous avons fixés à l'étape de conception. GMAC génère des clusters qui ont un très grand nombre de nœuds proches géographiquement et se déplaçant ensemble. D'autre part, DACE-L génère des clusters de taille plus homogènes en taille par rapport à ACE-L, toute en assurant l'alternance du rôle cluster-head entre les nœuds capteurs.

Dns ce qui suit, nous vérifions si nos protocoles apportent une amélioration en termes de consommation d'énergie et de stabilité des clusters générés comparativement aux protocoles ACE-L et LEACH.

5.3. Paramètres de simulation

Pour que l'évaluation des performances de nos propositions par simulation soit effective il faut que nous prenions en considération les spécificités des réseaux de capteurs pour qu'elle soit réalisable dans des conditions qui se rapprochent de la réalité. Certains paramètres ont été pris en compte lors de la simulation. Ces paramètres sont sélectionnés en se basant sur des travaux antérieurs pour des techniques similaires ainsi que sur la capacité du simulateur. Le tableau ci-dessous résume les paramètres utilisés:

Paramètre	Valeur
Surface du réseau	200 x 200
Position de la station de base	(100,300)
Nombre de nœuds	100
Taille du paquet	2000 <i>bits</i>
Période	20s
Nombre d'envois de données	4 fois par période
Bande passante	1MB/s
Nombre de cluster (points de référence)	4
ϵ_{Elec}	50 <i>nJ/bit</i>
ϵ_{Amp}	100 <i>pJ/bit/m²</i>
E_{fd}	5 <i>nJ/bit</i>
Energie initial	1J
Taille maximale d'un cluster	40 nœuds
Seuil minimal	1
Le coefficient α	1.5
Modèle de mobilité	RWM, RPGM

Tableau 7. Paramètres de simulation

Les nœuds capteurs utilisés dans les simulations sont considérés homogènes possédant la même quantité d'énergie initiale, la même capacité de calcul et de mémoire, la même portée de transmission et équipés par les mêmes interfaces de communication IEEE 802.11. De plus, l'énergie de la station de base est considérée comme illimitée. Nous allons opter à créer quatre clusters dans le réseau pour chaque période. Pour se faire, nous générerons quatre points de référence principaux de coordonnées ((50,50), (50,150), (150,50), (150,150)). Pour chaque point de référence principal nous créons quatre autres points de référence secondaire suivant la méthode discutée dans le quatrième chapitre.

5.4. Les métriques d'évaluation de performances

Pour évaluer les performances des quatre (4) protocoles : LEACH, ACE-L, DACE-L et GMAC, trois métriques ont été utilisées: nombre de paquets livrés à la station de base, l'énergie consommée (ou restante) dans le réseau et la durée de vie globale du réseau. Ces métriques sont affectées par des facteurs comme le nombre de nœuds utilisés dans les scénarios de simulation, le type de protocole...etc.

Les valeurs moyennes des paramètres sont obtenues par l'exécution de 20 simulations indépendantes pour chaque scénario.

5.4.1 Energie Moyenne Restante (EMR)

L'un des principaux critères de performances pour un nœud capteur est la durée d'utilisation efficace de son énergie embarquée avant de l'épuiser. Une fois épuisée, il n'y a pas d'autre moyen de la recharger à nouveau, surtout si les nœuds sont déployés dans des zones à accessibilité limitée.

L'Énergie Moyenne Restante (EMR) représente la quantité moyenne d'énergie restante dans le réseau. Elle est obtenue par la somme des énergies restantes des nœuds dans le réseau:

$$EMR = \sum_{i=1}^n Er$$

Avec Er l'énergie résiduelle d'un nœud capteur et n le nombre de capteurs dans le réseau.

Nos évaluations considèrent une évaluation de la moyenne de l'énergie restante après la fin de l'exécution de vingt simulations.

5.4.2 Nombre moyen de paquets délivrés à la station de base (NMPL)

Le nombre moyen de paquets livrés à la station de base représente la somme de tous les paquets reçus avec succès par la station de base. Cette métrique nous permet d'avoir une idée sur les performances du protocole et sa résistance à la mobilité des nœuds.

5.4.3 Durée de vie du réseau

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

Pour l'évaluation des performances des protocoles, nous avons suivi l'évolution de 100 nœuds et nous avons noté le temps au terme duquel le premier nœud tombe en panne FND (*First Node Die*). Ensuite, nous avons enregistré les délais au bout desquels 25%, 50% ou 96% des nœuds tombent en panne.

5.5. Discussion des résultats

Dans cette partie, nous allons évaluer et analyser les performances de nos protocoles en variant les différents paramètres liés au fonctionnement de ces protocoles.

5.5.1 Impact du nombre de clusters (points de référence) sur la durée de vie du réseau

Les techniques de clustering ont montré leur efficacité en termes d'optimisation de la consommation d'énergie, cependant, cette efficacité varie suivant le nombre de clusters formés et les caractéristiques du réseau de capteurs. Les travaux réalisés dans [10, 48] déterminent le nombre optimal de clusters pour lequel la consommation d'énergie est optimale. Ce nombre est généralement lié à certaines caractéristiques du réseau comme le nombre de nœuds, la taille de la scène, la position de la station de base etc.

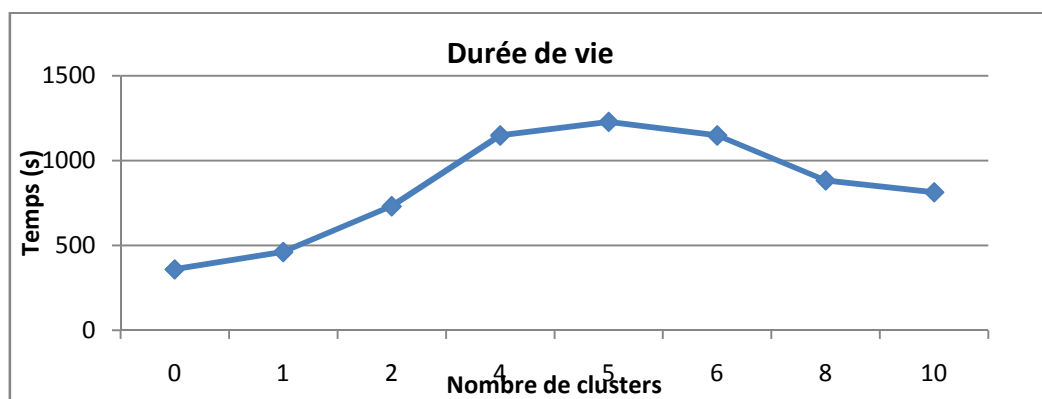


Figure 31. Durée de vie du réseau en fonction du nombre de clusters

La Figure 31 représente un graphe qui montre l'influence du nombre de clusters sur la durée de vie d'un réseau de capteurs. Nous avons réalisé des simulations sur le protocole ACE-L en variant le nombre de points de référence autour desquels les clusters seront formés (de 0 à 10). D'après ces résultats, nous pouvons constater que l'augmentation du nombre de clusters prolonge la durée de vie du réseau jusqu'à atteindre son seuil maximal pour cinq clusters; après ce seuil, l'augmentation du nombre de clusters aura un effet inverse sur la durée de vie du réseau. Nous pouvons alors conclure que le nombre de clusters (points de référence) est un facteur

déterminant pour la maximisation de la durée de vie du réseau. Ceci peu s'expliquer par le fait qu'un petit nombre de clusters avec des tailles importantes favorise la mis hors service des cluster-heads par épuisement de leur capacité énergétique. De plus les distances qui séparent les nœuds membres de leurs cluster-heads est toujours importantes ce qui implique une grande consommation d'énergie. D'autre part, un nombre trop important de clusters de petite taille avec beaucoup de singletons (cluster avec un seul nœud) produit une topologie proche de la topologie plate avec un coût inutile de gestion des clusters qui réduit la durée de vie du réseau.

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

5.5.2 Evaluation de DACE-L : Discussion des résultats

Pour l'évaluation de ce protocole nous avons considéré le scénario d'une compagnie de soldats installée dans un champ de bataille, cette compagnie est formée de 10 sections (groupes) qui se déplacent de façon aléatoire dans le champ de surveillance. Ces sections se rassemblent de façon périodique dans leur place d'arme (zone 2), ensuite ils reprennent leur déplacement de façon aléatoire.

Pour générer les nœuds mobiles suivant le modèle RPGM, nous avons utilisé BonnMotionn [55]. Nous avons évalué les protocoles ACE-L et DACE-L selon le nombre moyen de clusters générés, la moyenne d'énergie restante dans le réseau, le nombre moyen de paquets livrés à la station de base et la durée de vie moyenne du réseau. La durée de vie varie suivant les différents systèmes de durée de vie (pour 1%, 25% , 50% et 96 % de nœuds qui tombent en panne...).

a) Nombre moyen de clusters générés

L'un des objectifs des techniques de clustering est d'avoir un bon équilibrage de charge. Pour se faire, ces techniques essayent de générer un nombre homogène de clusters pour toutes les périodes.

Nombre de clusters	1	2	3	4	plus de 5
ACE-L	23.4%	13.7%	11.6%	51.3%	0%
DACE-L	0.5%	0.8%	0.7%	92,6%	4.4%

Tableau 8. Nombre de clusters générés par les protocoles ACE-L et DACE-L

Dans le tableau 8 nous avons résumé les résultats obtenus en termes du nombre de clusters générés de l'exécution des deux protocoles. Afin d'avoir un nombre homogène de clusters (4 clusters) pour chaque période, les auteurs d'ACE-L ont défini des points de référence fixes autour desquelles les clusters seront formés. Tandis que, dans notre contexte la distribution des nœuds capteurs est non uniforme dans le réseau, ce qui influe sur le nombre de clusters générée. Par exemple, dans le cas où les soldats se rassemblent dans leur place d'arme, un seul cluster est généré par cette méthode. Par contre dans DACE-L, le nombre de clusters générés et leurs tailles sont plus homogènes quelque soit la distribution des nœuds et leur densité dans le réseau. Cela est assuré par le mécanisme de partages des clusters à grand effectif en petits clusters autour des points de références secondaires.

b) Énergie Moyenne restante dans le réseau (EMR)

Le graphe ci dessous illustre la consommation d'énergie des deux protocoles de clustering: ACE-L et DACE-L. Nous remarquons sur la Figure 32 que DACE-L permet une meilleure optimisation de la consommation d'énergie par rapport à ACE-L. Cela est dû à la méthode de construction des clusters. Dans ACE-L, les clusters formés ne sont pas homogènes en taille à cause de la variation de la distribution des nœuds capteurs dans le réseau, alors que DACE-L limite la taille par un seuil maximal et un seuil minimal. Les clusters ayant une taille supérieure au seuil maximal vont être subdivisés en petits clusters de tailles homogènes. Ce qui permet aux nœuds membres de transmettre leurs données aux cluster-heads sur des distances moins importantes comparativement à ACE-L. D'où une optimisation de la consommation d'énergie.

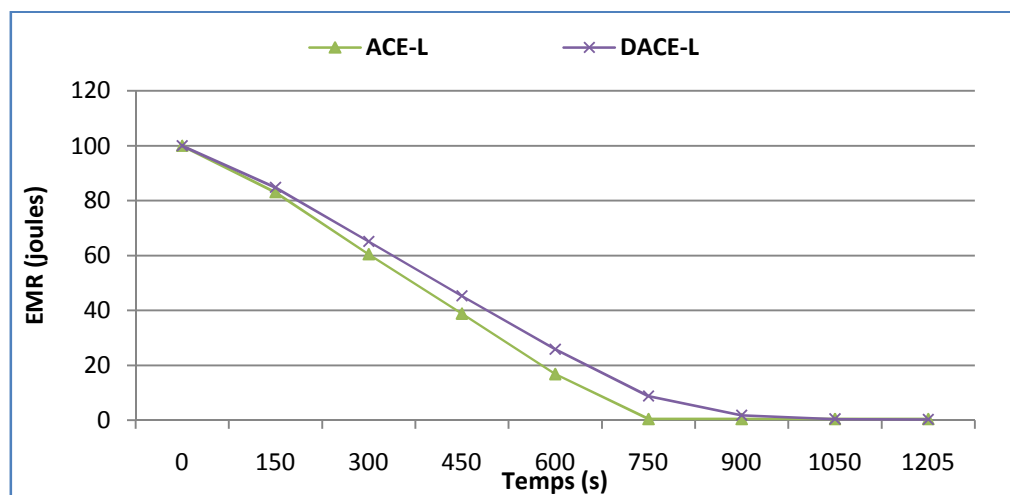


Figure 32. La consommation d'énergie en fonction du temps

c) La durée de vie moyenne

Nous remarquons sur la Figure 33 que le protocole DACE-L a une durée de vie plus longue que celle d'ACE-L suivant les différents critères de définition (pour 1%, 25%,...des nœuds en panne) de durée de vie. Cela est dû à la méthode de sélection des cluster-heads et à la stratégie de définition des clusters. Les deux utilisent les points de référence pour la sélection des cluster-heads. ACE-L se base sur les distances par rapport à ces points de référence pour choisir les cluster-heads. Alors que, DACE-L utilise à la fois la distance aux points de référence et l'énergie résiduelle des nœuds, en plus, d'un coefficient qui permet de choisir d'autres nœuds comme cluster-heads qui n'ont pas été choisis dans les périodes précédentes. Cela permet une meilleure distribution de charge entre les nœuds. En ce qui concerne la formation des clusters, ACE-L ne génère pas des clusters de taille homogène à cause de la variation de la distribution des nœuds dans le réseau. Alors que DACE-L limite la taille du cluster entre un seuil maximal et un seuil minimal, afin d'obtenir des clusters de tailles homogènes et ainsi équilibrer la charge qui assure une durée de vie plus longue.

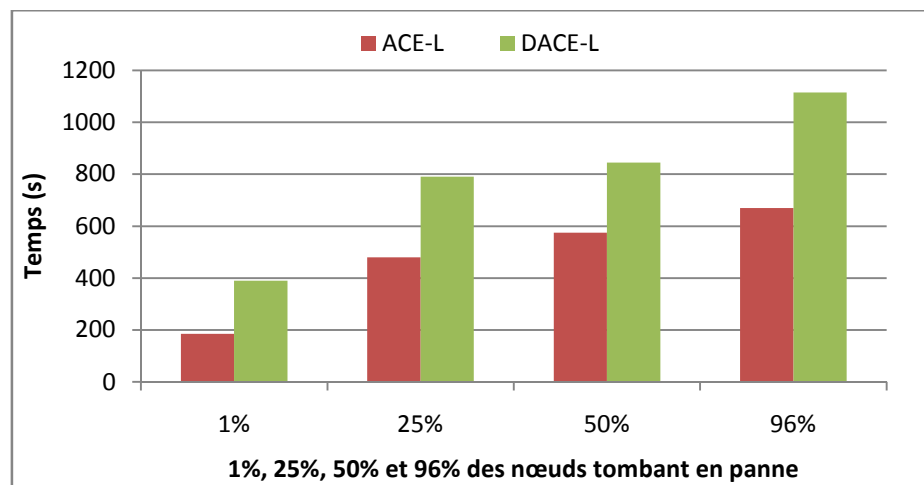


Figure 33. La durée de vie du réseau pour DACE-L

d) Nombre moyen de paquets livrés à la station de base (NMPL)

Le graphe ci dessous montre le nombre de paquets reçus par la station de base pour chacun des protocoles de clustering: ACE-L et DACE-L au cours du temps. Nous pouvons facilement faire le lien entre le nombre de paquets reçus et la durée de vie du réseau pour chaque méthode. La perte de paquets dans les deux protocoles est due à la défaillance des nœuds capteurs et à leur

mobilité (déconnexion). En effet, quand un nœud mobile s'éloigne de son cluster-head, ses données envoyées au cluster-head seront perdues.

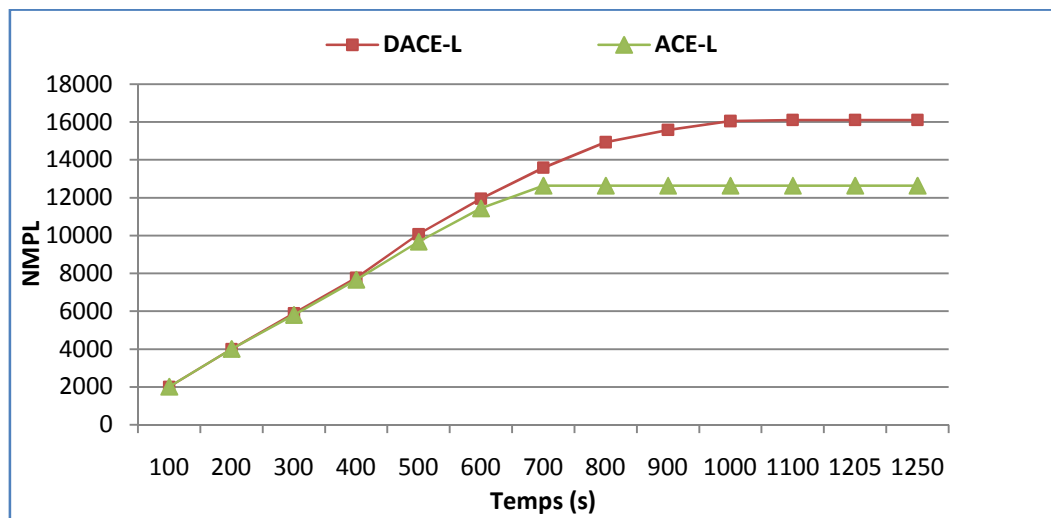


Figure 34. Nombre de paquets de données reçues par la station de base en fonction du temps pour DACE-L

5.5.3 Evaluation de GMAC : Discussion des résultats

Pour l'évaluation de ce protocole nous avons considéré en premier temps un scénario proche de celui utilisé pour l'évaluation du premier protocole où une compagnie de soldats installée dans un champ de bataille, cette compagnie est formée de 10 sections (groupes) qui se déplacent de façon aléatoire dans le champ de surveillance. Ensuite, nous varions la taille des sections et leur nombre afin d'évaluer les performances de notre protocole par rapport aux caractéristiques des groupes de mobilité. Dans une troisième étape, nous testons la fonctionnalité de ce protocole dans le cas où les nœuds se déplacent de façon aléatoire (par exemple, suivant le modèle RWM). Dans toutes les simulations de GMAC, nous utilisons la métrique de mobilité en groupe à un saut

a) Énergie Moyenne Restante dans le réseau (EMR)

Le graphe ci dessous montre la consommation d'énergie des protocoles de clustering: LEACH, ACE-L et GMAC. En observant l'énergie consommée pour chacune de ces méthodes, nous pouvons constater que les réseaux utilisant LEACH consomment leur énergie plus rapidement que les autres. Nous pouvons expliquer cette observation par la sélection aléatoire des cluster-heads, ce qui génère des clusters non homogènes ni en taille ni en nombre. ACE-L utilise une méthode basée sur les distances relatives aux points de référence pour la sélection des

cluster-heads. Cette méthode permet de former un nombre optimal de clusters durant toutes les périodes, ce qui permet de réduire la consommation d'énergie des nœuds. Notre technique GMAC se base sur la métrique *MobilityGroup* pour la sélection des cluster-heads. Cette métrique permet de former des clusters avec des nœuds se déplaçant ensemble et proches géographiquement, minimisant ainsi les distances des communications entre les membres et leurs cluster-heads, ce qui permet en conséquence une meilleur optimisation de la consommation d'énergie.

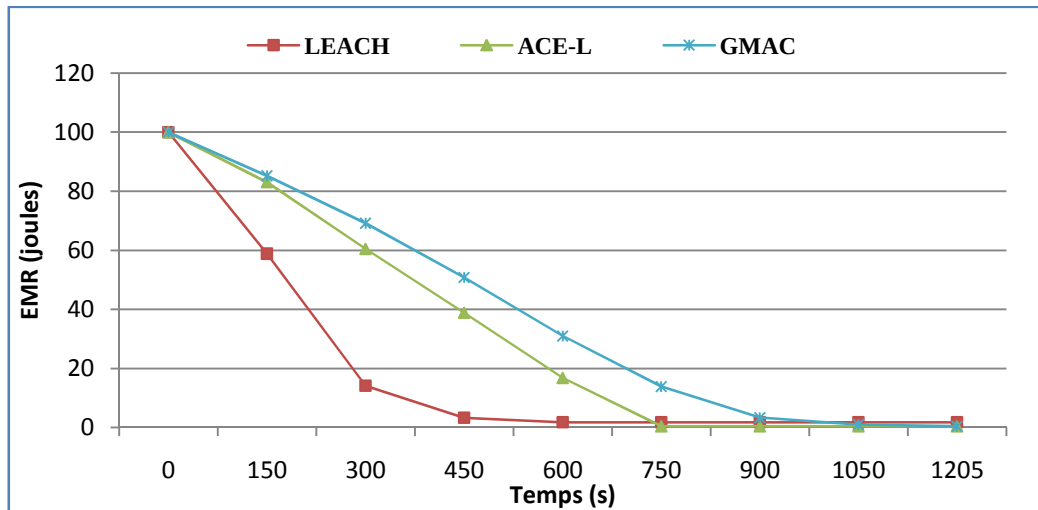


Figure 35. Energie moyenne restante en fonction du temps pour GMAC

b) La durée de vie moyenne

Nous remarquons sur la Figure 36 que notre protocole a une durée de vie plus grande que celle de LEACH et ACE-L pour les différents critères d'évaluation de la durée de vie. Cela s'explique par la surexploitation des ressources des cluster-heads, soit à cause de la taille importante des clusters générés dans le cas de LEACH, soit à cause des réélections successives d'un nœud comme cluster-head dans le cas d'ACE-L. Le critère de sélection des cluster-heads dans GMAC influence positivement sur la durée de vie du réseau du fait que les clusters générés sont plus stables et leurs nœuds membres sont plus proches de leurs cluster-heads. Ce qui permet de minimiser la consommation d'énergie lors des communications intra-clusters. D'autre part, le choix du cluster-head dépend de la métrique *MobilityGroup* liée aux positions des nœuds et de celles de leurs voisinages. Ainsi, l'alternance du rôle cluster-head entre les nœuds est assurée et donc les dépenses énergétiques sont équitablement distribuées sur le réseau.

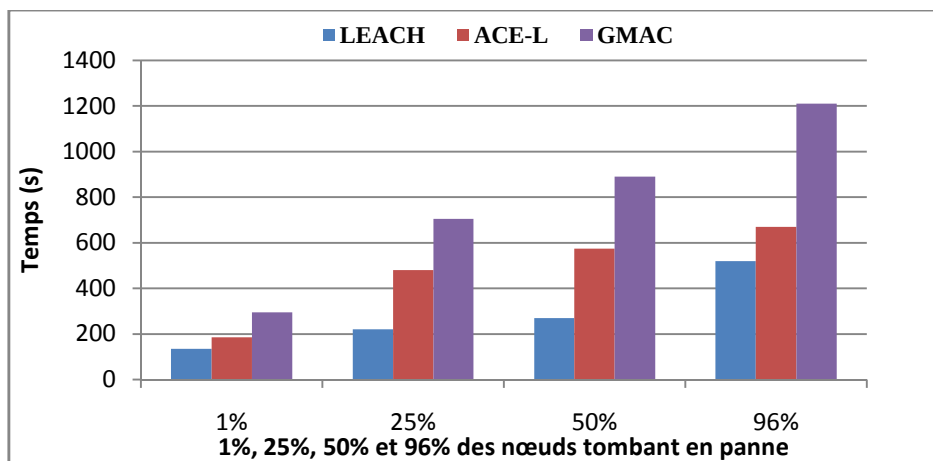


Figure 36. La durée de vie du réseau pour GMAC

c) Le nombre moyen de paquets livrés à la station de base (NMPL)

Les graphes obtenus sur la Figure 37 montrent que GMAC assure la délivrance d'un nombre plus important de paquets par comparaison à LEACH et ACE-L. Nous pouvons expliquer ces résultats, d'une part, par les résultats obtenus précédemment concernant la durée de vie de chaque protocole. D'autre part, nous trouvons l'explication par la stratégie de construction des clusters dans GMAC. Cette méthode minimise les pertes de paquets par apport aux autres protocoles. En effet, la sélection des cluster-head dans GMAC se fait par la métrique *MobilityGroup*, qui permet la formation de cluster contenant des nœuds membres se déplaçant en groupe ensemble ce qui minimise la perte des paquets.

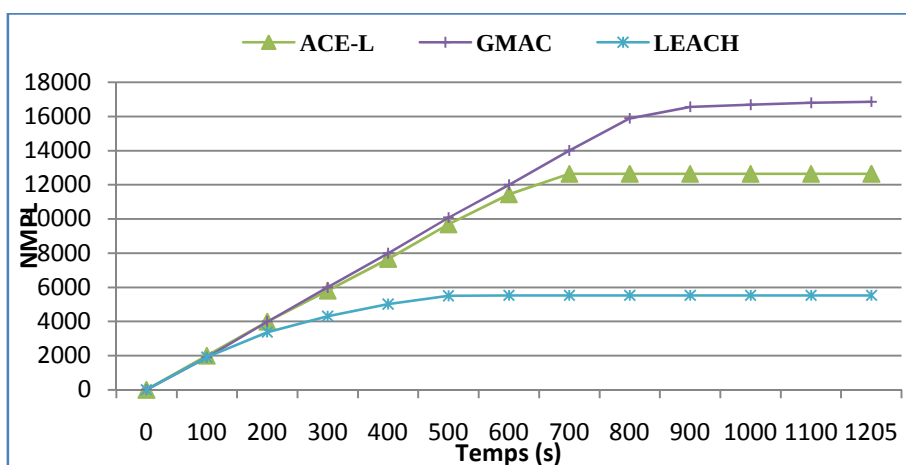


Figure 37. Nombre de données reçues par la station de base pour GMAC

d) Impact des caractéristiques des groupes sur les performances de GMAC

Dans cette partie nous allons évaluer l'influence des caractéristiques des groupes sur le fonctionnement de notre protocole GMAC. Dans ce but, nous avons généré des groupes de différentes tailles en variant leurs diamètres entre 20m et 70m. D'après les résultats montrés sur la Figure 38, nous remarquons que plus le diamètre des groupes augmente plus les performances du protocole en termes de consommation d'énergie diminuent. Cela s'explique par la variation des distances de communication intra-cluster. Plus les groupes sont compacts plus les distances de communication sont courtes et par conséquent une plus grande connectivité intra-cluster et une réduction de la consommation d'énergie. La taille des groupes a aussi un effet sur la consommation d'énergie par le protocole. Cette consommation d'énergie est maximale lorsque le nombre de groupes se rapproche de 1 ou de 100, cela est dû au fait qu'en se rapprochant de ces valeur la mobilité des nœuds dans le réseau ne sera pas en groupes mais plutôt plus proche de celle du modèle RWM ou un modèle aléatoire quelconque. D'autre part, nous constatons que plus le nombre de groupes générés se rapproche du nombre de points de référence plus l'efficacité en consommation d'énergie augmente pour notre protocole.

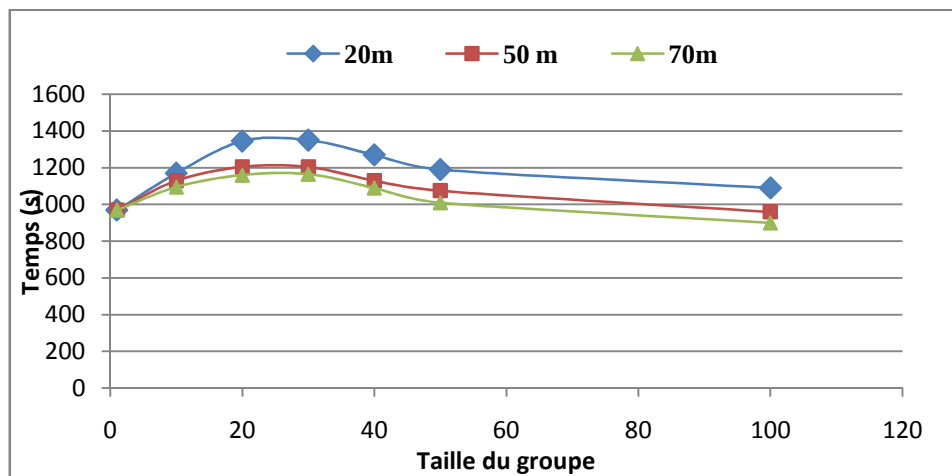


Figure 38. Durée de vie du réseau en fonction de la taille des groupes et leurs diamètres pour GMAC

Le protocole GMAC donne de bons résultats en termes de consommation d'énergie, de paquets livrés à la station de base et de durée de vie relativement à LEACH et ACE-L, mais la question qui se pose : est ce que ce protocole donne de bons résultats dans le cas où la mobilité suit le modèle RWM par exemple. Nous avons réalisé une série de simulations sur les protocoles

GMAC, ACE-L et LEACH avec ce modèle de mobilité. Les résultats obtenus sont illustrés sur le graphe qui représente l'énergie moyenne consommée.

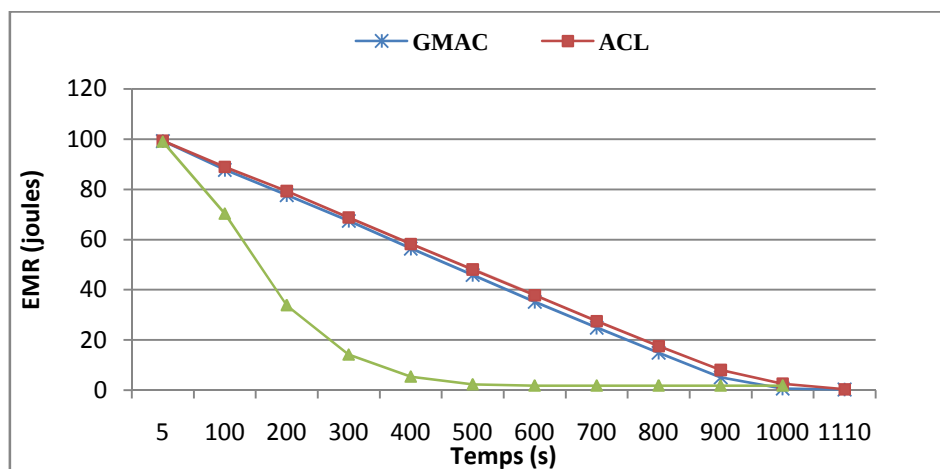


Figure 39. Energie moyenne restante en fonction du temps pour GMAC avec une mobilité suivant le modèle RWM

Sur le graphe ci-dessus, nous avons illustré les résultats fournis par notre proposition comparée à ACE-L et LEACH en termes de consommation d'énergie dans le cas d'une mobilité qui suit le modèle RWM. Malgré que notre proposition ait été conçue pour fonctionner dans le cas de mobilité en groupe, elle donne des résultats qui se rapprochent de ceux fournis par ACE-L. Cela est dû au fait que dans ACE-L, on exploite la mobilité des nœuds pour choisir les nœuds les plus proches des points de référence comme cluster-heads. Dans notre proposition, comme dans ACE-L la mobilité des nœuds va nous assurer la variation de la valeur du critère de choix des cluster-heads d'un instant à un autre ce qui permet d'attribuer en alternance le rôle du cluster-head aux les nœuds capteurs.

Conclusion

Dans ce chapitre nous avons fait une évaluation et une étude comparative entre nos protocoles DACE-L et GMAC et les protocoles LEACH et ACE-L où nous avons utilisé NS2 pour faire nos simulations. Nous avons commencé par faire des simulations sur ACE-L pour montrer l'effet du nombre de clusters sur la durée de vie du réseau. Ensuite nous avons fait une série de tests sur les différents protocoles qui ont montré que nos propositions améliorent les performances par rapport à LEACH et à ACE-L en termes de consommation d'énergie et de nombre de paquets livrés à la station de base, spécialement dans le cas où les nœuds se déplacent en groupe.

Conclusion générale

La conception des réseaux de capteurs est fortement influencée par la limitation des ressources énergétiques disponibles au niveau des nœuds capteurs. Actuellement, la plupart des travaux de recherche sur ce type de réseaux sont consacrés à la conception des protocoles visant à minimiser l'énergie inhérente aux communications afin d'optimiser la durée de fonctionnement du réseau.

Dans cette optique, le clustering s'est présenté comme étant une solution prometteuse pour conserver l'énergie des nœuds et faciliter la transmission des données capturées dans le réseau vers la station de base. Cette structure hiérarchique permet d'alléger le trafic du réseau en effectuant des traitements au sein de chaque cluster, avant de faire parvenir les données à la station de base. L'obtention de meilleures performances des réseaux de capteurs avec l'approche de clustering résulte d'une organisation efficace des nœuds en clusters, tout en essayant de produire un nombre de clusters stable et équilibré.

La majorité des techniques de clustering proposées dans la littérature pour les réseaux de capteurs supposent que les nœuds sont stationnaires et peu d'entre elles utilisent des capteurs mobiles. Mais pour certaines applications, il est important de supporter la mobilité des nœuds. Sous de telles conditions, le clustering devient une tâche très ardue. Les nœuds changent de positions et les clusters se voient obligés de s'adapter en continue aux variations dynamiques de configurations du réseau. Lorsqu'un nœud membre se déplace en dehors du cluster, les données captées par ce nœud peuvent être perdues et une éventuelle affectation de ces nœuds est nécessaire pour maintenir la structure en cluster du réseau.

Afin de réduire la consommation énergétique des nœuds capteurs dans le réseau tout en prenant en compte les limites rencontrées dans les solutions étudiées dans le cas de nœuds mobiles, nous avons apporté, en premier lieu, une amélioration au protocole de clustering ACE-L. Cette amélioration (DACE-L) consiste à ajouter des mécanismes pour que le protocole résultant s'adapte à la variation de la densité des nœuds dans le réseau. En second lieu, nous avons conçu une nouvelle technique de clustering (GMAC) qui supporte la mobilité des nœuds. Elle se base sur une métrique de mobilité des nœuds en groupe pour sélectionner leur responsable de cluster (cluster-head).

L'évaluation des performances des protocoles proposés a été réalisée en utilisant le simulateur de réseaux NS2. Les simulations effectuées ont permis de montrer l'efficacité de nos solutions concernant la réduction de l'énergie consommée par les nœuds capteurs, participant ainsi au prolongement de la durée de vie du réseau.

Comme perspectives à notre travail, nous envisageons d'améliorer le critère de mobilité en groupe et d'adapter notre mécanisme de sélection des clusters heads à des environnements où la localisation des nœuds n'est pas une information indispensable.

Références bibliographiques

- [1] B. Wang, “Coverage Control In Sensor Networks”, Edition Springer, ISBN 978-1-84996-058-8, 2010.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, “A Survey on Sensor networks”, IEEE Communications Magazine, pages 102-114, August 2002.
- [3] J.M.Kahn, R.H.Katz, and K.S.J.Pister, “Next century challenges: Mobile networking for Smart dust”, in Proc Mobicom, pages 483-492, 1999.
- [4] Xbow: Crossbow technology inc. <http://www.xbow.com/>, dernière visite 22/04/2010.
- [5] Crossbow: Micaz data sheet http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf, dernière visite 28/04/2010.
- [6] M. Youssef, N. El-Sheimy, “Wireless Sensor Network: Research vs. Reality Design and Deployment Issues”, Communication Networks and Services Research (CNSR), Mai 2007.
- [7] <http://www.alertsystems.org>, dernière visite 28/04/2010.
- [8] I.A. Essa, “Ubiquitous sensing for smart and aware environments”, IEEE Personal Communications, pages 47–49, October 2000.
- [9] Y. P. Chen, A. L. Liestman, J. Liu, “Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing”, Volume 2, volume 75, chapter Clustering Algorithms for Ad Hoc Wireless Networks, pages 145–164. Nova Science Publishers, 2005.
- [10] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Micro Sensor Networks”. IEEE Proc of the Hawaii International Conference on System Sciences (HICSS), 2000.
- [11] A. D. Amis et al, “Max-Min D-cluster Formation in Wireless Ad Hoc Networks”. IEEE Proc of INFOCOM, pages 32 – 41, 2000.
- [12] S. Basagni, “Distributed Clustering Algorithm for Ad Hoc Networks”. Proc Int, Symp, Parallel Architectures, Algorithms and Networks, pages 310 – 315, 1999.
- [13] C.Li, M.Ye, G.Chen, J.Wu, “An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks”, IEEE International Conference on Mobile Adhoc and Sensor Systems, 2005.

- [14] M. Lehsaini, H. Guyennet, M. Feham, "CES: Cluster-based Energy-efficient Scheme for Mobile Wireless Sensor Networks". Chapter in the book *Wireless Sensor and Actor Networks II* (Boston Springer), vol.264, pages13-24, 2008.
- [15] I.Jung and Al, "An Energy Efficient Clustering Method for Wireless Sensor Networks", *Proceeding of optical Communications*, Corfu Island, Greece, Feb 2007.
- [16] C. Liu, C. Lee, L. Wang, "Distributed clustering algorithms for data-gathering in wireless mobile sensor networks", *Journal of Parallel Distrib. Comput*, pages 1187-1200, 2007.
- [17] G. Gupta, M. Younis, "Load-balanced clustering in wireless sensor networks", *Proceedings of the International Conference on Communication (ICC 2003)*, Anchorage, Alaska, May 2003.
- [18] G. Gupta, M. Younis, "Fault-tolerant clustering of wireless sensor Networks", *Proceedings of the IEEE Wireless Communication and Networks Conference (WCNC 2003)*, New Orleans, Louisiana, March 2003.
- [19] M. Younis, K. Akkaya, A. Kunjithapatham, "Optimization of task allocation in a cluster-based sensor network", *Proceedings of the 8th IEEE Symposium on Computers and Communications (ISCC'2003)*, Antalya, Turkey, June 2003.
- [20] V. Mhatre, C. Rosenberg, "Design guidelines for wireless sensor networks: communication, clustering and aggregation". *Ad Hoc Networks Journal*, vol.2, no.1, pages 45-63, 2004.
- [21] A.A. Abbasi, M. Younis, "A survey on clustering algorithms for wireless sensor networks", *Computer Communications* 30, pages 2826-2841, 2007.
- [22] M. Perianu, "Wireless Sensor Networks in Motion: Clustering Algorithms for Service Discovery and Provisioning", *PHD thesis, Univertsiry of Twente*, 2008.
- [23] D.S. Kim and Y.J. Chung, "Self-organization routing protocol supporting mobile nodes for wireless sensor network," presented at the 1st Int. Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006
- [24] M. Ye, C. Li, G. Chen, J. Wu, "EECS: an energy efficient cluster scheme in wireless sensor networks". In *proceedings of the IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks*, Phoenix, Arizona, 2005.
- [25] M. Gerla, T. J.Kwon, G. Pei, "On-demand routing in large ad hoc wireless networks with passive clustering". In *Wireless Communications and Networking Conference (WCNC)*, pages 100–105. IEEE Computer Society, 2000.
- [26] C. Bettstetter, "Mobility Modeling, Connectivity, and Adaptive Clustering in Ad Hoc Networks". *PHD thesis, University of M"unchen, Germany*, October 2003.

- [27] W.R. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, "Application-specific protocol architecture for wireless microsensor networks". IEEE Transactions Wireless Communications, vol.1, no.4, pages 660-670, 2002.
- [28] O. Younes, S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks". IEEE Transactions on Mobile Computing, vol.3, no.4, pages 366-379, 2004.
- [29] S. Lindsey, C.S. Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems". IEEE Transaction on Parallel and Distributed Systems, vol.13, no.9, pages 924-935, September 2002.
- [30] A. Manjeshwar, D.P. Agrawal, "TEEN: A Protocol for Enhanced Efficiency in Wireless Sensor Networks". 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, 2001.
- [31] A. Manjeshwar, D. P. Agrawal, "An Efficient Sensor Network Routing Protocol (APTEEN) with Comprehensive Information Retrieval," Proc. Second Int'l Workshop Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2002.
- [32] T. J. Kwon, M. Gerla, "Efficient flooding with passive clustering (pc) in ad hoc networks, SIGCOMM Comput. Commun. Rev., vol. 32, no. 1, pages 44–56, 2002.
- [33] A. Hac, "Wireless Sensor Network Designs", West Sussex PO19 8SQ, England, Available: cs-books@wiley.co.uk, 2003.
- [34] M.M. Rashid, M.M. Alam, C.S. Hong, "Energy conserving passive clustering for efficient routing in wireless sensor network", in 9th International Conference on Advanced Communication Technology, pages 982–986, 2007.
- [35] C. Intanagonwiwat et al., "Directed diffusion for wireless sensor networking", IEEE/ACM Transactions On Networking (TON), vol. 11, no. 1, pages 2–16, 2003.
- [36] H. Zeghilet, N. Badache, M. Maimour, "Energy Efficient Cluster-based Routing in Wireless Sensor Networks", published in 14th IEEE Symposium on Computers and Communications, ISCC'09, Sousse: Tunisia, 2009.
- [37] B. Liu, P. Brass, and O. Dousse, "Mobility Improves Coverage of Sensor Networks", MobiHoc'05, 25- 7, Mai 2005.
- [38] S. Munir, B. Ren, W. Jiao, B. Wang, "Mobile wireless sensor network: Architecture and enabling technologies for ubiquitous computing". International Conference on Advanced Information Networking and Applications Workshops, 2007.

- [39] R. Shah, S. Roy, S. Jain, "Data mules: modeling a three-tier architecture for sparse sensor networks". Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.
- [40] L. Fang, P. Antsaklis, L. Montestruque, M. McMickell, "Design of a wireless assisted pedestrian dead reckoning system", the NavMote experience. IEEE Transactions on Instrumentation and Measurement, 2005.
- [41] Q. Wang, M. Hempstead, W. Yang, "A realistic power consumption model for wireless sensor network devices", 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON), 2006.
- [42] C.M. Liu, C.H. Lee, "Power efficient communication protocols for data gathering on mobile sensor networks", Proceedings of the 60th IEEE Vehicular Technology Conference, Fall, 2004.
- [43] M. Liliana, C. Arboleda, N. Nasser, "Cluster-based routing protocol for mobile sensor networks, 3rd ACM international conference on Quality of service in heterogeneous wired/wireless networks, Waterloo, Ontario, Canada, vol. 191, 2006.
- [44] X. He, Q. Hao, Y. Song, "An Event-aware Clustering Algorithm of Mobile Wireless Sensor Network Nodes", ISECS International Colloquium on Computing, Communication, Control, and Management, 2009.
- [45] S. Gandham, M. Dawande, R. Prakash, S. Venkatesan. "Energy efficient schemes for wireless sensor networks with multiple mobile base stations", IEEE Global Telecommunications Conference, 2003.
- [46] G.S. Kumar, M.V. Vinu, K.P. Jacob, "Mobility Metric based LEACH-Mobile Protocol", 6th Intl Conference on Advanced Computing and Communication (ADCOM), 2008.
- [47] S.A.B. Awwad, C.K. Ng et al. "Cluster Based Routing protocol for Mobile Nodes in Wireless Sensor Network", 2009 International Symposium on Collaborative Technologies and Systems, 2009.
- [48] T. Chan, C. Chen, Y. Huang, and al, "Optimal Cluster Number Selection in Ad-hoc Wireless Sensor Networks", Wireless Transactions On Communications, Issue 8, Volume 7, 2008.
- [49] W. Andrew, "K-means and hierarchical clustering", 2001.
- [50] D. Davies, "Evaluating mobility models within an ad hoc network", Master's Thesis, Colorado School of Mines, 2000.
- [51] E. Royer, P.M. Melliar-Smith and L. Moser, "An analysis of the optimum node density for ad hoc mobile networks", Proc IEEE International Conference on

Communications (ICC), 2001.

- [52] D. Davies, “Evaluating mobility models within an ad hoc network”, these de Master, Colorado School of Mines, 2000.
- [53] G. Pei, M. Gerla, X. Hong, C.C. Chiang, “A wireless hierarchical routing protocol with *group mobility*”. In Proceedings of IEEE WCNC'99, Sep 1999.
- [54] V. Raghunathan, C. Schurgers, S. Park, M.B. Srivastava, “Energy-Aware Wireless Microsensor Networks”, IEEE Signal Processing Magazine, vol.19, pages 40-50, 2002.
- [55] BonnMotion, a mobility scenario generation and analysis tool, <http://iv.cs.uni-bonn.de/wg.cs/applications/bonnmotion/>.
- [56] NS2 (Network Simulator 2): Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, The ns Manual. February 27, 2006. <http://www.isi.edu/nsnam/ns/ns-documentation>.