

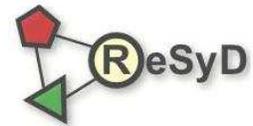
RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaïa

Faculté des Sciences Exactes
Département d'Informatique
École Doctorale d'Informatique

Mémoire de Magistère en Informatique

Option : Réseaux et Systèmes Distribués



Thème

L'auto-structuration dans les Réseaux de Capteurs Sans Fil

Présenté par :

LAOUID Abdelkader

Devant le jury :

Président	M ^r Ahmed AÏT SAÏDI	Professeur	U. A/Mira Béjaïa.
Rapporteur	M ^r Moussa KERKAR	Professeur	U. A/Mira Béjaïa.
Examineur	M ^r Abdelkamel TARI	MCA	U. A/Mira Béjaïa.
Examineur	M ^r Ali MELIT	Professeur	Université de Jijel.
Invité	M ^r Hamouma MOUMEN	MAA	U. A/Mira Béjaïa.

Remerciements

« Tout d'abord, je remercie Dieu, le tout puissant de nous avoir accordé le savoir, le droit chemin et l'opportunité de poursuivre mes études ».

Je tiens à exprimer tous mes vifs Remerciements et ma profonde Gratitude à mon promoteur : Pr M. KERKAR et Mr H. MOUMEN pour leurs disponibilités, aide et bonne humeur durant toutes les étapes de ce projet. Leurs dévouements, conseils scientifiques et suivis, m'ont permis de mener mon travail à terme.

Mes sincères Remerciements et ma profonde gratitude vont également au président et aux membres de jury pour l'honneur qu'ils m'accordent, en acceptant de juger mon travail.

Et enfin merci à tout ceux qui ont contribué de près ou de loin durant l'accomplissement de mes études.

Résumé

Résumé

L'apparition des applications de surveillances dans l'industrie ces dernières années, et l'impossibilité de l'être humain à intervenir pour maintenir ou renouveler l'énergie, ont poussé un grand nombre de chercheurs à proposer des solutions à économie d'énergie au niveau de la pile protocolaire dans la perspective de minimiser la dépense énergétique nodale et de maximiser la durée de vie du réseau global d'une manière complètement autonome.

Dans ce mémoire, nous avons proposé un protocole de structuration pour les réseaux des capteurs sans fil. En fait, il n'existe qu'un seul sujet dans la littérature pour la structuration décentralisée des nœuds capteurs, il a été proposé par un groupe de l'IRISA. Ces chercheurs ont pu développer un protocole d'auto-structuration des RCSFs, VINCOS, ce protocole offre beaucoup d'avantages aux réseaux installés dans des zones hostiles où l'intervention humaine n'est pas possible, tel que les coordonnées virtuelles. En revanche, ce protocole consomme beaucoup plus d'énergie surtout dans la phase « initialisation de la bordure », cette phase contient un nombre indéterminé d'itérations.

Pour cette raison, notre contribution comprend un protocole qui prend en compte la consommation d'énergie comme une ressource critique. Une seule itération suffit pour déterminer la bordure, mais nous avons abordé au centre. Ce dernier était déterminé au travers d'une technique simple et efficace, il s'agit de déterminer le nœud central en calculant le paramètre de convergence N . En plus, nous avons minimisé le nombre de capteurs avec la couverture maximale de la zone d'intérêt. L'idée maîtresse est de mettre en veille chaque capteur qui a une table de voisinage incluse dans (ou égale à) une table de l'un de ses voisins, et que tous les autres nœuds vont se mettre actifs jusqu'à la disparition de l'un de ces nœuds, un capteur voisin en veille va le remplacer afin de garder le réseau vivant le plus longtemps possible.

Mots clés : *Réseaux sans fil, Réseaux de capteurs, Découverte de voisinage, Auto-structuration, Systèmes autonomes.*

Abstract

The appearance of surveillance applications in industry in recent years, and the inability of humans to intervene to maintain or renew energy, this has led many researchers to propose solutions to saving energy on the level of the protocolar stack with a prospect to minimize the nodal energy expenditure and to maximize the lifespan of the total network in a completely autonomous way.

In this memory, we proposed a protocol structure for wireless sensor networks WSN. Indeed, there is one subject in the literature for structuring decentralized sensor nodes, was proposed by the group of "IRISA". They have developed a protocol for self-structuring of WSNs, VINCOS, this protocol offers many advantages to networks installed in hostile areas where human intervention is not possible, such as virtual coordinates. However, this protocol consumes much more energy especially in the phase "initialization of the border", this phase includes an indeterminate number of iterations.

From this reason, we proposed a protocol in our contribution taken into account the energy consumption as a critical resource. One iteration to determine the border (the leaves of the tree), but we approached the center. In addition we also minimized the number of sensors with the maximum coverage area of interest. The idea is standby each sensor has a neighbor table included or equal in a neighbor table of its neighbors, and all other nodes will remain active, to the disappearance of one these nodes. At the disappearance of one of the active nodes, a neighbor standby sensors will replace to keep the network live a long time possible.

keywords : *wireless networks, wireless sensor networks, neighbor discovery, Self-structuring, Autonomous systems.*

Table des matières

Remerciements	i
Résumé	ii
Table des matières	iv
Table des figures	vi
Liste des tableaux	vii
Introduction Générale	1
1 Introduction aux Réseaux des Capteurs	4
1.1 Introduction	4
1.2 Réseaux des capteurs	4
1.2.1 Définition d'un capteur	4
1.2.2 Définition d'un réseaux de capteurs	5
1.2.3 Les trois phases de la vie d'un réseau de capteurs	6
1.2.4 Réseaux des capteurs et Défis	7
1.3 Conclusion	9
2 L'auto-structuration dans les RCSFs	10
2.1 Introduction	10
2.2 La notion d'auto-structuration	10
2.2.1 L'auto-structuration et l'auto-organisation	11
2.2.2 Caractéristiques d'un système auto-structuré	11
2.3 LEGOS (Low Energy self-organization scheme for WSNs)	12
2.3.1 Hypothèses	12
2.3.2 Description	13
2.3.3 Procédures de construction et maintenance de LEGOS	14
2.3.4 Les apports de LEGOS	18
2.4 Conclusion	19
3 Protocoles d'auto-structuration dans les RCSFs	21
3.1 Introduction	21
3.2 L'auto-structuration avec GPS et puissance de signal	21
3.2.1 Historique du système GPS	21
3.2.2 Algorithmes d'auto-structuration centralisé	23
3.3 Quelques techniques de localisation dans les réseaux mobiles	26
3.3.1 Localisation collaborative	26
3.4 L'auto-structuration avec des informations locales	29
3.4.1 Comment évaluer une auto-structuration ?	30
3.4.2 Apports de l'auto-structuration	30
3.5 VINCOS(Virtual networked coordinate system)	32

3.5.1	Hypothèses	33
3.5.2	Contexte	33
3.5.3	Description	33
3.6	Les apports de VINCOS :	37
3.7	Conclusion	38
4	Un protocole simple et efficace pour l'auto-structuration	39
4.1	Introduction	39
4.2	Contexte et Motivation	39
4.3	Modèle du système	40
4.3.1	Le principe de base de la solution	42
4.4	Hypothèses	44
4.5	Un algorithme implémentant l'auto-structuration	45
4.5.1	Notation	45
4.5.2	Les phases de formation de l'arbre	45
4.5.3	Présentation de l'algorithme	48
4.6	Simulation	51
4.7	Cordonnées virtuelles pour les réseaux de capteurs	62
4.7.1	Évaluation des résultats	64
4.8	Conclusion	65
	Conclusion générale et perspectives	66
	Bibliographie	68

Table des figures

1.1	Capteur sans fil	5
1.2	Composants de base d'un capteur	6
1.3	Réseau de capteurs	7
1.4	La portée d'un capteur	8
2.1	Exemple de topologie LEGOS	13
2.2	Exemple de backbone LEGOS	14
2.3	Arrivée de nouveaux nœuds	15
2.4	Départ d'un gateway	16
2.5	Départ d'un leader	17
3.1	Principe de GPS	22
3.2	Coordonnées virtuelles	25
3.3	Intersection de signaux	25
3.4	La localisation Collaborative dans les RCSFs et Ad-hoc.	27
3.5	Exemple de coordonnées virtuelles	34
3.6	Les nœuds de bordure	35
3.7	Les voisinages des nœuds	36
4.1	Représentation graphique(non orienté) d'un RCSF	41
4.2	Les phases de formation du réseau	42
4.3	Représentation graphique de propagation des messages	44
4.4	propagation que les messages pertinents	44
4.5	Distribution rectangulaire de 100 capteurs	52
4.6	Nombre de voisins de chaque nœud (i, j) avec $R = 3$	53
4.7	Les voisins de chaque nœud (i, j) avec $R = 2$	53
4.8	Les nœuds veilles (+)	54
4.9	Le nœud central (*)	54
4.10	La propagation du message	55
4.11	Distribution circulaire	55
4.12	Les feuilles de la forme circulaire	56
4.13	L'arbre de la forme circulaire	56
4.14	La distribution aléatoire	57
4.15	Nombre de messages	58
4.16	Les capteurs en veilles	59
4.17	Le nœud central	59
4.18	Nombre de messages de 300 nœuds	60
4.19	Nombre maximum de nœuds mis en veille	61
4.20	Nombre maximum de message	61
4.21	L'arborescence du réseau	62

4.22 La segmentation du réseau	63
4.23 Structuration avec des segments équivalents	64

Liste des tableaux

- 4.1 Calcule du 1-degré et 1-densité 41
- 4.2 Les notations utilisées dans la proposition 46

Introduction Générale

De nos jours, les réseaux de télécommunications sont présents dans de très nombreux domaines. Que ce soit dans la production, l'espace ou la domotique, les réseaux sont devenus incontournables. Le multiplexage des données sur un bus unique permet de limiter considérablement les câbles, abaissant globalement les coûts, tant à l'installation qu'à la maintenance. Les câbles sont généralement inesthétiques, coûteux, difficiles à installer et peu pratiques à l'usage. De plus, dans des domaines industriels où l'appareil est soumis à des contraintes vibratoires très importantes, leur connectique représente une faiblesse qui augmente considérablement le risque de panne. Enfin, dans certains domaines comme l'aéronautique ou le spatial, les câbles représentent un poids non négligeable qu'il est absolument nécessaire de limiter par une infrastructure plus légère. Dans cette démarche de forte limitation des câbles, le sans fil permet d'aller encore plus loin que le bus de données. Le support de transmission sans fil étant immatériel, il permet de gagner largement en poids, il peut traverser des cloisons sans nécessité de perçage et permet une forte mobilité des appareils connectés. En revanche, si les gains sont importants sur ces aspects, l'utilisation du sans fil amène son lot de contraintes qui empêche la transposition directe des mécanismes utilisés sur les bus filaires au sans fil. Par exemple, le phénomène d'aveuglement empêche la détection des collisions en temps réel, la forte variabilité des caractéristiques du médium radio entraîne une grande incertitude sur la fiabilité des liens et la forte atténuation du médium immatériel implique des portées limitées qui ne sont pas toujours simples à appréhender, même avec beaucoup d'expérience. Les perspectives de recherches, tant sur la couche physique que sur les protocoles de niveaux supérieurs, restent toujours importantes dans ce domaine.

En parallèle, les applications de surveillance, de protection et d'observation de notre environnement se sont accrues ces dernières années, aussi bien dans le milieu médical qu'écologique ou militaire. Malheureusement, l'intervention humaine n'est pas réalisable dans tous les lieux d'investigation. Les années 2000 ont alors vu la naissance de « poussières intelligentes », ensemble de plate-formes matérielles miniaturisées couplées avec un module d'acquisition de données permettant de capter et de collecter des stimuli ou événements issus de l'environnement local de ce capteur. Les faibles possibilités technologiques d'un capteur unique ne lui permettent pas d'avoir une utilité propre, mais la mise en réseau d'un grand nombre de ceux-ci ouvre des opportunités de progrès et de fonctionnement palpitantes.

L'intérêt des communautés issues de la recherche et de l'industrie pour ces réseaux de capteurs sans fil (RCSFs) s'est accru par la potentielle fiabilité, précision, flexibilité, faible coût ainsi que la facilité de déploiement de ces systèmes. Les caractéristiques des RCSFs permettent d'envisager un grand nombre d'applications d'observation répartie dans l'espace. Ces dernières peuvent se déployer dans de nombreux contextes : observation de l'environnement naturel (pollution, inondation, . . .), suivi d'écosystèmes (surveillance de niches d'oiseaux, croissance des plantes, . . .), militaire (télésurveillance de champs de bataille, détection d'ennemis, . . .), biomédical et surveillance médicale (détection de cancer, rétine artificielle, taux de glucose, diabète, . . .), etc. La spontanéité, l'adaptabilité du réseau et la dynamique de sa topologie dans le déploiement des RCSFs soulèvent de nombreuses questions encore ouvertes. L'une des limitations contraignantes de ces réseaux est la faible ressource énergétique

des capteurs due essentiellement à leur taille minimaliste et leur indépendance filaire. Cette contrainte doit être considérée comme la préoccupation de premier ordre dans la conception et le déploiement d'un RCSF.

Motivation et Contexte

Contraintes et Objectifs Compte tenu du nombre potentiellement important de capteurs participants et de leurs ressources réduites, il apparaît indispensable de développer des solutions entièrement décentralisées, répartissant la charge entre les participants et permettant de passer à l'échelle en terme de nombre d'entités déployées. Par conséquent, dans le contexte des RCSFs, l'approche collaborative dans la conception de systèmes dédiés apparaît nécessaire. En effet, ces derniers sont la plupart du temps développés dans un cadre applicatif précis afin d'optimiser la consommation d'énergie locale et globale, dans le but d'optimiser la durée de vie du réseau. Enfin, cet ensemble de capteurs devant être capable de délivrer un service sans nécessiter d'intervention extérieure (humaine ou matérielle), la collaboration doit être conçue de manière auto-organisante. Cette dernière notion a pour optique l'émergence d'un comportement général et global du système, à partir d'acteurs indépendants et possédant uniquement des informations restreintes et locales du système. Il s'agit d'émerger une structure donnée à partir des informations locales c'est un facteur très important dans l'auto-organisation. Ainsi, la charge est répartie sur l'ensemble du système, et l'augmentation du nombre de participants n'entraîne pas la formation de goulot d'étranglement, ni de point de défaillance critique (par l'unicité du serveur), couramment observés sur les systèmes communicants dit centralisés.

Limitations des systèmes existants Depuis leur récente apparition, les RCSFs ont connu une effusion d'intérêt permettant en quelques années de réunir un ensemble de connaissances considérable. Nombreuses sont les applications déployées fournissant des services fondés sur un RCSF de nos jours. Par exemple, nous pouvons citer la surveillance de zone sécurisée, ou la domotique. Malheureusement, la grande majorité des résultats scientifiques et techniques dans ce domaine a été développée dans un contexte applicatif. à ce jour, encore très peu de travaux présentent des analyses fondamentaux et/ou théoriques des RCSFs. Une modélisation générique communément admise, permettant d'explorer les enjeux et les limitations de ce domaine, manque toujours à cette communauté en plein essor.

Plan du mémoire

Dans ce mémoire, on présente un protocole de structuration des capteurs sans fil denses. Nous commencerons par une brève description des réseaux de capteurs, leurs applications et leurs principaux défis. Néanmoins, avant d'atteindre la phase d'implantation industrielle à grande échelle des réseaux de capteurs, beaucoup de problèmes restent encore à traiter et beaucoup de solutions proposées dans la recherche demeurent encore à améliorer, telles que les solutions proposées aux problèmes d'optimisation de l'énergie dépensée, de routage des données, d'auto-structuration en cas de déploiement dans une zone hostile, etc.

Ce mémoire relate en second lieu les points essentiels caractérisant la notion d'auto-structuration (centralisée gérée par une station de base, GPS. . .et décentralisée), sa définition et les métriques qui l'évalue et expose certaines solutions proposées dont la littérature. À savoir quelques exemples de stratégies orientés dominant et des stratégies de sélection de liens.

A l'issue de cette recherche, on a présenté le protocole de type backbone LEGOS et le système de coordonnées virtuelles VINCOS. On a aussi noté que l'auto-structuration signifie la structuration d'un réseau pour rendre son exploitation plus aisée, et l'association d'un ensemble de fonctionnalités permettant de maintenir cette structure et faciliter son utilisation. Elle est le résultat d'application d'algorithmes qui organisent le comportement global d'un système. Donc, il faut penser à proposer des structures assurant la structuration d'un RCSF.

Finalement, nous avons proposé une solution simple et claire adaptée aux RCSFs de topologies denses de manière complètement décentralisée, sans aucune information globale sur le réseau et aucune connaissance externe. Cette solution cherche à augmenter la durée de vie du réseau et à assurer une couverture maximale avec un minimum de capteurs actifs, ce en construisant les ensembles "not include not equal " pour chaque nœud. Puis structurer le réseau en arbre initiée par le nœud central, construction de l'arbre et enfin la segmentation de réseau. Les résultats obtenus par simulation montrent clairement la structure obtenue. Cette solution peut être utilisée pour d'autres défis liés aux RCSFs tels que le routage ou la localisation.

Chapitre 1

Introduction aux Réseaux des Capteurs

1.1 Introduction

Les réseaux filaires ainsi que les réseaux sans fil classiques possèdent structurellement une hiérarchie. Celle-ci se manifeste par la présence d'entités dédiées au bon fonctionnement du réseau : routeurs, antennes-relais, point d'accès... Ces entités, situées au cœur du réseau, ont pour seule finalité de l'organiser : assurer le routage des paquets, la couverture radio, la sécurité... afin que les terminaux puissent utiliser ce réseau. Les réseaux de capteurs, par leur nature homogène (tous les capteurs sont considérés comme physiquement identiques), ne possèdent pas d'infrastructure préalable : chaque entité du réseau va potentiellement utiliser les fonctionnalités du réseau et en assurer le fonctionnement. De plus, de par les contraintes énoncées précédemment, les réseaux de capteurs doivent mettre en place des mécanismes d'organisation du réseau sans intervention humaine, efficaces et économes en énergie.

Durant ces dernières années, beaucoup d'applications de Réseaux des Capteurs Sans Fils (RCSFs) sont apparues. Pour mieux comprendre la différence entre les RCSFs et les autres réseaux, nous devons lister les propriétés et les capacités des capteurs. Les réseaux des capteurs se composent d'un nombre important de nœuds de capteurs. le contrôle de l'environnement de chaque capteur se fait d'une manière autonome [11]. L'objectif final est de voir le comportement global émergé du réseau, i.e. surveillance d'une large zone , en utilisant la méthodologie d'auto-organisation. Pour établir cette tâche, les nœuds de capteurs doivent pouvoir rassembler l'information de chaque capteur, pour les expédier au-dessus d'un réseau ad-hoc à une destination finale telle qu'une station de surveillance.

1.2 Réseaux des capteurs

1.2.1 Définition d'un capteur

Un capteur est un dispositif qui transforme l'état d'une grandeur physique observée en une information manipulable

Il est caractérisé par :

- Rayon d'émission : environ 100m
- Position : cordonné (x, y, z)
- Énergie : capacité de batterie
- Cout d'émission /puissance d'énergie : d^4

Un capteur est composé de quatre unités de base, à l'instar de la représentation en figure 1.2 : une unité de capture, une unité de traitement, une unité de communication et une unité d'alimentation



FIGURE 1.1 – Capteur sans fil

[17] [23]

L'unités de capture permettant la collecte de données issus de stimuli externes, est le plus souvent constitué de deux sous-composants : des capteurs classiques combinés à un convertisseur analogique-vers-numérique (ADC pour Analog-to-Digital Converter). Le signal analogique, généré par le phénomène observé perçu par les capteurs, est converti en un signal numérique par l'ADC et transmit à l'unité de traitement

L'unités de traitement est composé d'un processeur ou Unité Centrale de Traitement (UCT, ou CPU pour Central Processing Unit) généralement associé à un périphérique de stockage de taille réduite. Il exécute le processus permettant la collaboration des nœuds entre eux afin de mener à bien les tâches de capture dévolues.

L'unités de communication permet de connecter les nœuds du réseau, quels que soient le support et le média utilisé par celui-ci. Le plus souvent, il est constitué principalement d'un périphérique de communication sans fil de type Bluetooth, ZigBee ou même WiFi. Il représente la principale source de consommation d'énergie et doit ainsi être largement pris en compte.

L'unité d'alimentation est quant à lui l'un des composants les plus importants d'un nœud. Il assure la durée de vie du capteur en approvisionnant en énergie tous les modules internes et externes du capteur. De même que pour l'unité de communication, toute conception de systèmes sur des RSCSFs doit prendre en compte l'utilisation de ce module.

1.2.2 Définition d'un réseaux de capteurs

Un réseau de capteurs sans fil consiste en une grande population, relativement dense, de capteurs dans un territoire quelconque et dont le seul moyen de communication est l'utilisation des interfaces sans fil, sans l'aide d'une infrastructure préexistante ou administration centralisée [33].

Description et caractéristiques Les réseaux de capteurs sans fil -Wireless Sensor Networks (WSN)- sont considérés comme un type spécial de réseaux ad hoc. Les nœuds de ce type de réseaux

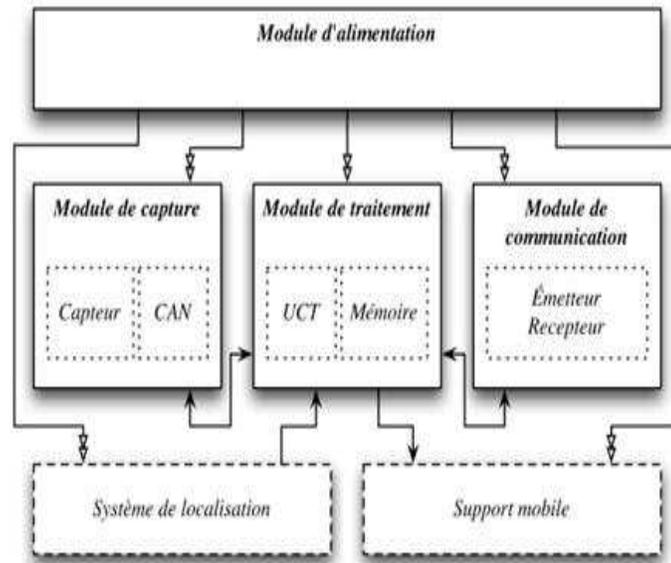


FIGURE 1.2 – Composants de base d'un capteur

consistent en un grand nombre de capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils sont dispersés aléatoirement à travers une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le phénomène capté. Les données captées sont acheminées grâce à un routage multi-saut à un nœud considéré comme un "point de collecte", appelé nœud puits (ou sink). Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou un satellite. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du nœud puits.

1.2.3 Les trois phases de la vie d'un réseau de capteurs

Quand on observe dans sa globalité un réseau de capteurs sans fils, il apparaît des phases distinctes avec des caractéristiques particulières : la naissance, le fonctionnement normal et la mort[20].

1. **La phase de naissance** : correspond à l'arrivée progressive des nœuds, à la fois durant le déploiement initial du réseau de capteurs et durant le redéploiement de nouveaux nœuds dans le réseau. Il en résulte une phase où les nœuds découvrent leur voisinage. Chaque nœud diffuse des paquets hello pour indiquer leur présence, transmettre des informations sur leur état, et diffuser leur table de voisinage. Des incohérences dans ces tables de voisinage apparaissent alors. En effet, à cause du déploiement progressif, les capteurs ont généralement une vision partielle de leur voisinage durant l'envoi de leurs premiers messages. Cela conduit à des erreurs durant le processus d'élection des nœuds dominants ou le processus de sélection de liens. Nous observons ces incohérences et le temps nécessaire avant la stabilisation de la structure logique. Nous déterminons ainsi la latence entre la naissance physique d'un réseau et sa naissance logique suivant son schéma d'auto-organisation.
2. **La phase de fonctionnement** : commence dès que la structure organisée est stabilisée. Cette phase correspond à ce que nous pouvons attendre de l'auto-organisation. Nous mettons en évidence la qualité d'une auto-organisation durant cette phase en observant la cardinalité de la

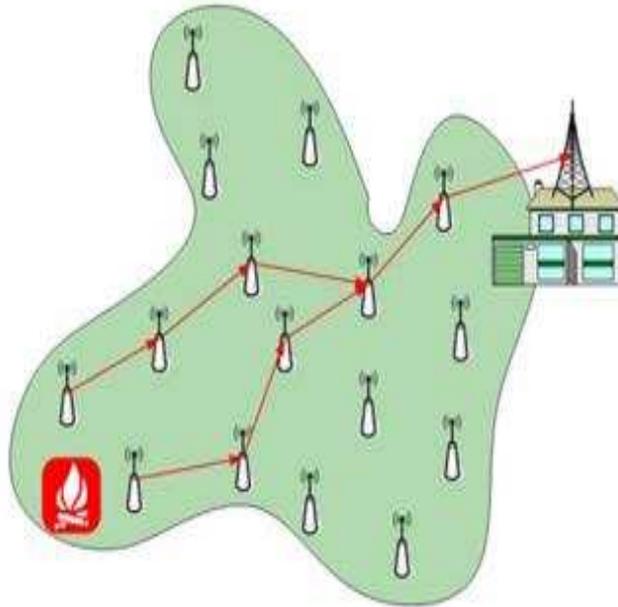


FIGURE 1.3 – Réseau de capteurs

structure dominante dans le cas de protocoles orientés dominants, le degré moyen du réseau dans le cas de protocoles à sélection de liens. La phase de fonctionnement normal se termine quand il y a trop de modifications dans le réseau pour que la structure soit maintenue.

3. **La phase de la mort** : Lorsqu'un ou plusieurs nœuds disparaissent, la reconstruction de la structure logique, globale ou localisée va être nécessaire. Là encore, les capteurs n'ont pas immédiatement la perception de la disparition d'un voisin, cela a pour conséquence une certaine inertie. Cette dernière phase peut être assimilée à un processus de maintenance et d'auto-réparation.

1.2.4 Réseaux des capteurs et Défis

Notre travail entre dans le cadre d'émerger une structure à partir des informations internes avec la prise en compte de consommation d'énergie possible, tel que l'objectif final est de résoudre le problème de routage dans les réseaux de capteurs sans fils. Le problème de routage au sein des WSN est très compliqué, cela est dû essentiellement à l'absence d'infrastructure fixe (forêt, mer,...), de toute administration centralisée et à d'autres défis qui doivent être vérifié qui sont les suivant :

Consommation d'énergie sans perte d'efficacité¹ Les capteurs utilisent leur réserve d'énergie à des fins de calcul et de transmission de données. La durée de vie d'un capteur dépend essentiellement de celle de sa batterie. Dans un WSN, chaque nœud joue le rôle d'émetteur et de routeur. Une défaillance énergétique d'un capteur peut changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier[8].

Tolérance aux pannes Certains nœuds peuvent être défaillants à cause d'un manque d'énergie, d'un dommage physique ou d'une interférence. Ceci ne doit pas affecter la globalité de la tâche du

1. Nous cherchons dans toutes nos contributions à limiter son utilisation et ainsi améliorer la durée de vie du nœud capteur.

réseau de capteurs. En cas de défaillance, de nouveaux liens et routes doivent être établis pour assurer la collecte des données. La redondance peut également être utilisée, tout en veillant à conserver une faible consommation d'énergie[18].

Couverture Dans les WSN, chaque nœud perçoit une vue locale de son environnement, limitée par sa portée et sa précision. La couverture d'un vaste espace déterminé est donc composée de l'union de nombreuses couvertures de petite taille. La figure 1.4 présente la portée d'un capteur et ses voisins.

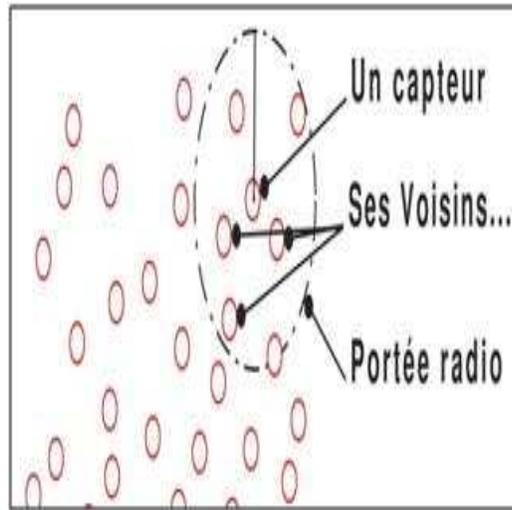


FIGURE 1.4 – La portée d'un capteur

Déploiement de nœuds C'est un facteur dépendant de l'application qui affecte grandement les protocoles de routage. Le déploiement peut être déterministe ou aléatoire. Dans la première stratégie, les capteurs sont placés manuellement et les données peuvent donc être acheminées via des chemins prédéterminés. En revanche, avec une approche aléatoire, les capteurs sont éparpillés (lâchés d'un avion). Dans le cas d'une répartition non uniforme, une stratégie de groupement (clustering) peut s'avérer nécessaire[18].

Modèle de couverture des données La capture d'information et la couverture des données requises dépendent essentiellement de l'application. Elles peuvent être classifiées selon différents modèles : en fonction du temps (surveillance périodique), des événements (réaction à l'occurrence d'un événement particulier), des requêtes (réponse à une demande d'une station de base) ou de manière hybride (combinaisons des précédentes approches)[8].

Hétérogénéité des nœuds /connections Dans de nombreuses études, tous les capteurs d'une application sont considérés homogènes (i.e. même capacité de calcul, de communication et d'énergie). Néanmoins, en fonction de l'application, certains capteurs peuvent avoir des rôles différents, générant une architecture hétérogène[18]. Par exemple, dans un WSN hiérarchique, certains capteurs sont déclarés « chef » de leur groupe. Le routage vers les stations de base est alors traité par ces derniers.

Agrégation de donnée De nombreux capteurs peuvent générer de la redondance. L'agrégation de paquets similaires en provenance de différents nœuds permet de réduire le nombre de transmissions ou un traitement du signal peut également être effectué, par fusion de données (conformation de faisceaux)[8].

Qualité de service Dans diverses applications, la donnée doit être transmise dans une certaine plage de temps, après quoi, elle n'aura aucune importance. Pourtant, dans la plupart des applications, la durée de vie du réseau est favorisée au détriment de la qualité d'émission des données. Les protocoles de routage assurant une qualité de service et prenant en compte la gestion de l'énergie, représentent un défi nouveau et stimulant[18].

1.3 Conclusion

Nous avons présenté dans ce chapitre une vision générale sur les réseaux des capteurs et leur intérêt. Nous constatons que les réseaux des capteurs possèdent un large domaine d'application, leurs propres caractéristiques et propres contraintes. L'importance des réseaux de capteurs apparaît principalement dans divers domaines, notamment, les réseaux de communication sans fil dont les performances et les applications ne cessent d'augmenter. Il s'avère que la distribution et l'organisation des capteurs dans l'espace est un grand défi.

L'auto-structuration est un mécanisme permettant aux réseaux de capteurs de s'organiser pour construire une structure virtuelle sur laquelle les algorithmes s'exécuteront plus efficacement que sur les réseaux à plat (structure réelle). Cette structure virtuelle représente une hiérarchie des nœuds où chacun d'eux aura un rôle précis dans l'acheminement des données vers le puits. Même avec son importance, elle est vulnérable face à la sécurité des réseaux de capteurs. Le chapitre suivant traite en détail le phénomène d'auto-structuration et son importance dans les réseaux sans fil.

Chapitre 2

L'auto-structuration dans les réseaux de capteurs sans fil

2.1 Introduction

Avant d'étudier l'auto-structuration dans les réseaux de capteurs sans fils et de mesurer son impact en termes quantitatifs (performances) et qualitatifs (facilité de déploiement de protocoles réseaux), il nous semble important de la définir plus précisément. Pour nous, la notion d'auto-structuration est corrélée à celle de structuration du réseau à l'opposé de la vision plus classique qui consiste à considérer un réseau à plat. L'idée maîtresse est qu'une fois structuré/organisé le comportement des protocoles tels que le routage, l'inondation, la localisation, etc. est plus efficace. Nous abordons donc par la notion d'auto-structuration sous le point de vue d'un comportement émergent, structurant le réseau à partir d'interactions locales uniquement. Selon notre point de vue, l'auto-structuration vise à tirer parti des propriétés des nœuds pour dégager une structure permettant de l'organisation du réseau. Cette structure se doit d'être autonome et dynamique tel qu'un changement local ne doit que faiblement impacter la structure globale. Le recours à cette structure auto-organisante doit faciliter le déploiement de protocoles de communication. La notion d'interactions et de décisions locales seront les principes fondamentaux de l'auto-structuration.

Par conséquent, nous attendons des propriétés d'adaptation dynamique à l'environnement, de robustesse et bien entendu que le concept de localisé entraîne un meilleur passage à l'échelle que le recours à des solutions distribuées et a fortiori centralisées.

2.2 La notion d'auto-structuration

L'auto-structuration se rapporte aux capacités d'un réseau d'émerger une structure spécifique, à partir de zéro, sans exigence d'une information externe. Une auto-structuration est une dimension importante de l'autonomie de système. Dans les réseaux des capteurs, l'auto-structuration représente une condition importante pour les opérations communes, tel que l'expédition, équilibrage, élection de leader et gestion de consommation d'énergie. Notre étude consiste à diviser un secteur en plusieurs zones pour surveiller une région donnée, par un ensemble de capteurs, et l'objectif est de générer des algorithmes spécifiques pour économiser l'énergie.

La complexité du mécanisme d'auto-structuration dépend fortement à la quantité de connaissance initiale fournies aux nœuds dans le réseau. Si par exemple tous les nœuds dans le système ont une connaissance complète du système, la structuration est donc inutile. D'autre part, si les nœuds ne

reconnaissent que leurs voisinages, s'assure que d'émerger une structure à partir des décisions individuelles. La connaissance externe est l'information fournie à un nœud par une entité ou un dispositif externe. Ceci est contraire à la connaissance intrinsèque, telle que chaque entité rassemble les informations à partir de sa observation de réseau. La connaissance externe est requise, la difficulté de système apparue dans les événements où l'intervention humaine est difficile. Au lieu de cela, des approches s'appuyant surtout dans les systèmes autonomes augmentent définitivement la connaissance intrinsèque

L'objectif de ce chapitre est donc de définir de façon plus précise les concepts inhérents à l'auto-structuration. Nous allons prendre dans cette section quelques domaines où l'auto-structuration est requise.

2.2.1 L'auto-structuration et l'auto-organisation

Les réseaux des capteurs sont généralement difficiles à gérer à partir des événements externes (station de base, satellite. . .) où l'intervention humaine est éventuellement impossible. Nous allons alors chercher des autres techniques pour rendre les RCSFS fonctionnent sans interventions externes. Il s'agit l'auto-organisation c'est une technique très répandue dans les RCSFS de tel sorte que le réseau fonctionne sans aucune information de l'extérieur. Cette technique nécessite d'abord de s'auto-structurer les nœuds pour gérer le routage et l'inondation. . .

- L'auto-organisation est un phénomène de mise en ordre croissant, et allant en sens inverse de l'augmentation de l'entropie ; au prix d'une dissipation d'énergie qui servira à maintenir cette structure.¹
- L'auto-organisation fait référence à un processus dans lequel l'organisation interne d'un système (habituellement un système hors équilibre) augmente automatiquement sans être dirigée par une source extérieure. Typiquement, les systèmes auto-organisés ont des propriétés émergentes (bien que cela ne soit pas toujours le cas).²
- L'auto-structuration Émergence "spontanée" d'une structure ordonnée, due aux interactions dynamiques entre les éléments composants du système considéré.³

Après ces définitions, nous constatons que la notion d'auto-structuration fait référence à l'organisation d'un système sans interaction avec une entité extérieure et sans contrôle centralisé. Ainsi, l'auto-structuration doit nécessairement être basée sur des interactions locales, de façon complètement distribuée. Un réseau auto-organisé est souvent perçu comme un réseau où les nœuds s'adaptent au changement de topologie de façon localisée ou distribuée comme un mécanisme de découverte de routes. Il n'y a pas là d'idée de structure, d'organisation du réseau.

2.2.2 Caractéristiques d'un système auto-structuré

Individuellement, chaque nœud possède une vue locale, peut communiquer avec ses voisins et capable d'actions dites de bas niveau. Globalement l'ensemble doit pouvoir résoudre une tâche complexe, en agissant de manière coordonnée tout en étant adaptable, robuste, et de dimension ajustable[13].

- **Interactions locales uniquement** : Sans contrôle extérieur.
- **Émergence d'une structure globale à partir d'informations locales** : L'auto-structuration doit émerger une vue globale. On peut considérer qu'un réseau de capteurs est, par essence (de nature), un système auto-organisé car tous les protocoles utilisés sont basés sur des interactions locales entre les nœuds et sont répartis sur l'ensemble des membres du réseau. Il ne s'agit pas de se limiter uniquement à la configuration d'adresses ou au routage mais bien de fournir une

1. fr.wikipedia.org/wiki/Auto-organisation

2. <http://www.techno-science.net/>

3. <http://www.definitions-de-psychologie.com/fr/definition/auto-organisation.html>

structure virtuelle dynamique et unique permettant de faciliter le déploiement de tout protocole réseau.

- Réactivité aux changements locaux et robustesse :

1. Adaptabilité : les systèmes auto-structuré sont adaptables à l'environnement et réagissent aux changements locaux. En fait, les entités s'adaptent continuellement aux changements d'une façon coordonnée, tel que le système se réorganise toujours. En faisant ainsi, il essaye de converger vers des structures avantageuses désirables en évitant d'autres structures.
2. Robustesse : Comme il n'y a pas d'entité centrale, dans un système auto-organisé il n'y a pas de nœuds critiques et le système peut être réparé sans aide extérieure et continue à fonctionner malgré la perte de fréquence de l'un de ses éléments (le système peut corriger des dégâts sans aide externe).

- Passage à l'échelle : Le système auto-structuré est capable d'adapter sa taille et sa forme aux circonstances, même avec un nombre de nœud très important, il doit continuer à fonctionner efficacement. Cette propriété découle de l'absence de contrôle centralisé et des interactions locales.

2.3 LEGOS (Low Energy self-organization scheme for wireless sensor networks)

LEGOS est un protocole auto-organisation des réseaux de capteurs proposé par f. Valois [13]

Il avait pour objectif de fournir un protocole d'auto-organisation simple, efficace en énergie, adapté aux topologies larges échelles qui permet de structurer le réseau et qui permet de s'adapter aux changements dans le voisinage. La structure du réseau devait répondre aux exigences suivantes :

- Seules des interactions locales doivent être utilisées pour faire émerger un comportement global et réagir aux changements dans le voisinage ;
- Eviter l'émission de paquets périodiques hello afin d'économiser de l'énergie,
- La complexité et la puissance de calcul requise doivent être efficaces pour être adaptés aux ressources limitées des capteurs.

2.3.1 Hypothèses

L'auteur [13] de LEGOS ciblait les réseaux de capteurs sans fil avec les hypothèses suivantes :

- Capteurs identiques : tous les capteurs ont les mêmes capacités en termes de mémoire, puissance de calcul, communication et d'énergie. Seul le puits peut avoir des capacités différentes mais il n'est pas pris en compte afin de garantir plus de flexibilité ;
- Energie comme ressource critique : l'optimisation de l'énergie doit être présenté systématiquement pour améliorer la durée de vie du réseau ;
- Aucune information de localisation géographique n'est connue de la part des nœuds et aucune information globale n'est requise.
- Aucune hypothèse sur la distribution des nœuds sur la zone de surveillance : les nœuds sont déployés aléatoirement.
- Aucune hypothèse en termes de synchronisation : les nœuds exécutent LEGOS indépendamment sans être nécessairement synchronisés.
- Pas d'infrastructure.
- Pas d'hypothèse sur la mobilité.
- Remontées de données vers un ou plusieurs puits, éventuellement mobile.

2.3.2 Description

LEGOS est un protocole asynchrone orienté évènement destiné à l'auto-organisation des réseaux de capteurs sans fil. Des règles locales sont appliquées en fonction des modifications du voisinage. La méthode de découverte du voisinage est fonction de l'existence ou non d'une auto-organisation déjà présente. Seul un sous ensemble des nœuds du réseau émet périodiquement un paquet de type hello pour annoncer un voisinage organisé ; la cardinalité de cet ensemble est assez faible. Une des différences principales avec les autres protocoles d'auto-organisation (LMST , CDS, ...) est que : le gain en robustesse, latence, efficacité est important puisque il ne nécessite pas de reconstruction périodique.

1. Les Points clefs de LEGOS

- Pas de message hello périodique,
- Procédures orientées évènement pour la construction et la maintenance (arrivée, départ, gestion de partitions),
- Réorganisation locale,
- Connaissance et interactions locales,
- Construction incrémentale.

2. La topologie d'auto-organisation générée par LEGOS

Dans LEGOS les nœuds capteurs peuvent prendre l'un des trois rôles suivant : leader, gateway ou member.

- Leader : un leader prend en charge les communications à un saut (voisin direct),
- Gateway : permet l'interconnexion de plusieurs leaders,
- Member : est un nœud passif sans fonction particulière en dehors du comportement applicatif et qui est rattaché à un seul leader nécessairement présent dans son voisinage à 1 saut. Des leaders et des gateways forment un ensemble connecté dominant. Seuls les leaders émettent périodiquement des paquets (Leader Broadcast Msg) dans le voisinage à 1 saut pour informer de l'existence d'une auto-organisation stable.

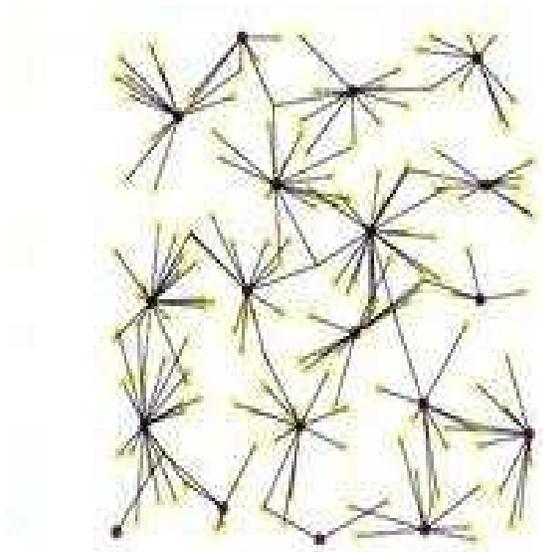


FIGURE 2.1 – Exemple de topologie LEGOS

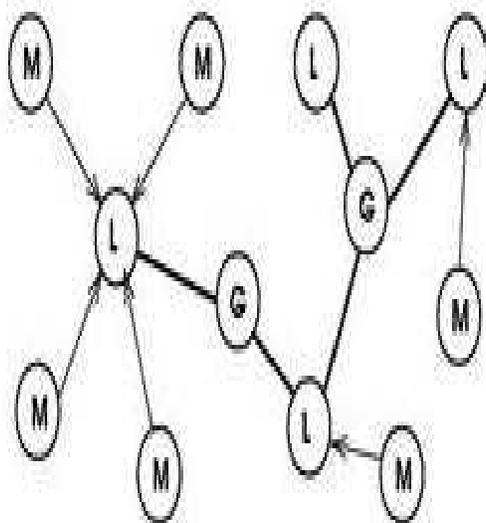


FIGURE 2.2 – Exemple de backbone LEGOS

2.3.3 Procédures de construction et maintenance de LEGOS

LEGOS est caractérisé par des procédures de construction et de maintenance orientées événement. En effet quatre événements locaux sont identifiés : L'arrivée de nouveaux nœuds, le départ des nœuds, la gestion de partitions et une procédure de réorganisation locale.

1. L'arrivée de nouveaux nœuds

Cet événement correspond au processus de déploiement du réseau, un nœud arrivant tire pleinement profit du voisinage déjà organisé. Donc lorsqu'un nœud N arrive il doit intégrer la structure d'auto-organisation suivant les trois étapes suivantes :

- Le nœud N exécute la procédure d'auto-organisation à un saut c'est-à-dire N écoute le médium pour détecter un (Leader Broadcast Msg). Si un leader est détecté grâce à ce message, N devient member et se rattache à ce leader.
- Si N ne détecte pas de message (Leader Broadcast Msg). Cela veut dire qu'il n'y a pas de leader dans le voisinage, N exécute la procédure d'auto-organisation à 2 - sauts : il émet un (Member Sollicitation Msg) pour la découverte de son voisinage. Si un nœud M de l'auto-organisation entend ce message, ça ne peut être qu'un member. Il répond alors avec (Member Advertissement Msg) : N devient alors leader et M devient gateway.
- Si les deux premières procédures sont infructueuses, N va se considérer comme le premier nœud d'une partition : il va se déclarer leader, générer un numéro aléatoire de partition et émettre périodiquement un paquet (Leader Broadcast Msg) qui inclut cet identifiant de partition. Les deux figures ci-dessous illustrent la procédure d'arrivée de nouveaux nœuds capteurs :

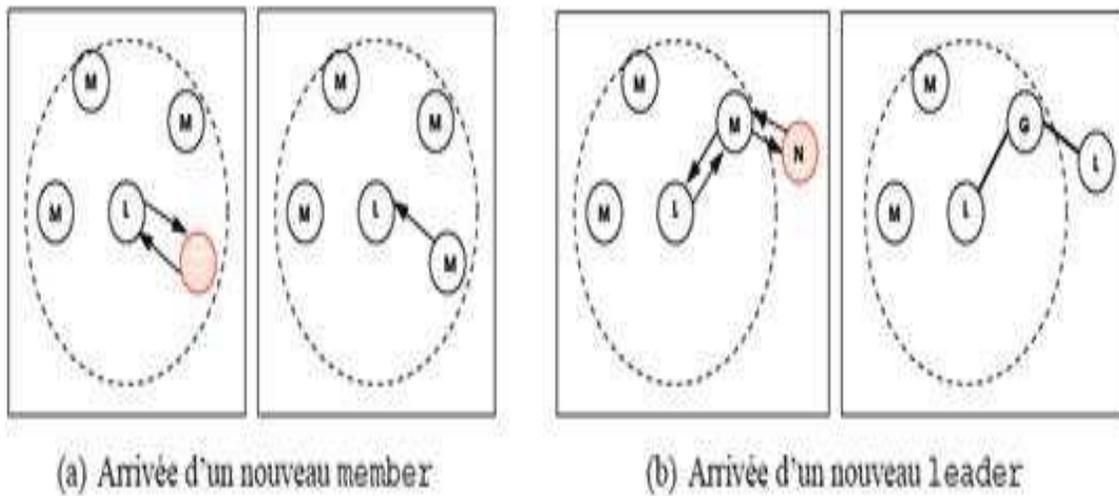


FIGURE 2.3 – Arrivée de nouveaux nœuds

2. Le départ de nœuds

Cet événement correspond au processus de mort et/ou mobilité des nœuds. Et comme c'est indiqué précédemment les nœuds leader, gateway ou member dans LEGOS ont des rôles différents donc leur départ respectif n'a pas le même impact :

- Le départ d'un member : étant donné que ce type de nœud ne prend pas un rôle vital dans l'auto-organisation, son départ n'a pas besoin d'être détecté. Cependant tous les paquets applicatifs qui devaient être transmis vers ce member seront perdus.
- Le départ d'un gateway : le départ d'un gateway est détecté soit par l'information explicite de chacun des leaders de son prochain départ, soit par un mécanisme de détection explicite des gateways par les leaders lors de l'absence de transmission de données. Pour faciliter la reconstruction locale, chaque leader doit connaître les identifiants des leaders connectés via les gateways donc lors du départ d'un gateway, les leaders impliqués demandent explicitement à leurs nœuds member d'entrer dans un mode de découverte du voisinage afin de déterminer quels members sélectionner pour Procéder à une reconstruction locale.

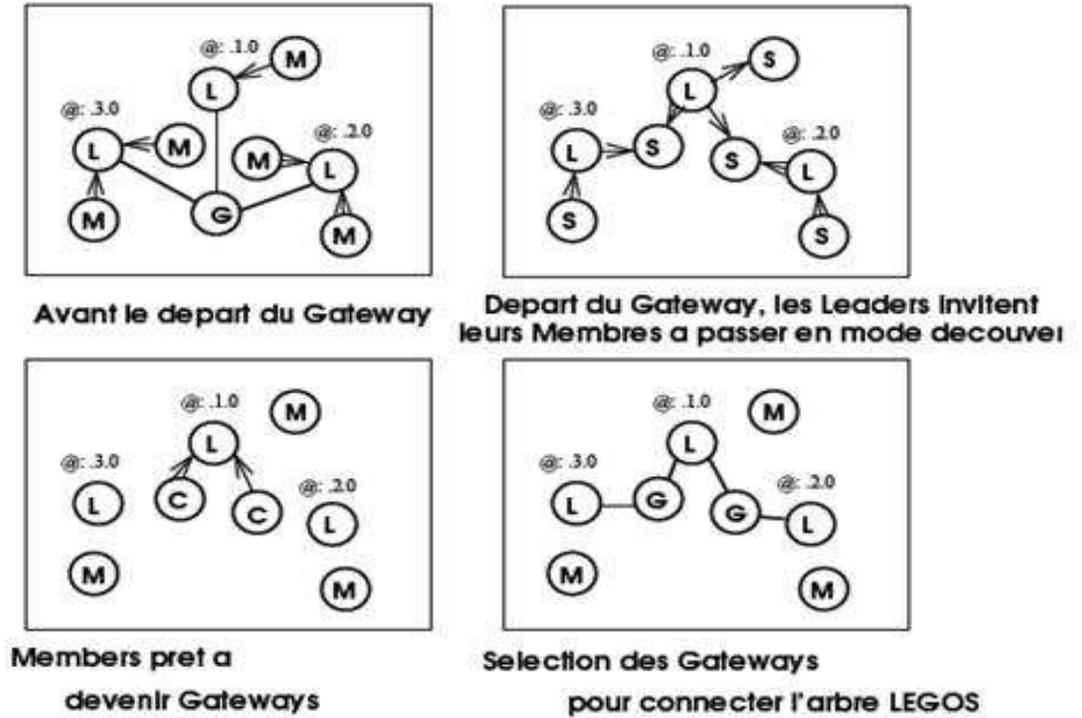


FIGURE 2.4 – Départ d'un gateway

- Le départ d'un leader : si plusieurs paquets (Leader Broadcast Msg) consécutifs ne sont pas reçus les autres nœuds de son voisinage à 1 saut (membres ou gateways). Les gateways qui ont détecté le départ du leader deviennent membres puis l'ensemble des membres se lance dans la découverte de voisinage comme si les nœuds venaient d'être déployés : chaque membre se rattache alors à un leader et certains membres deviennent gateways ou leaders suivant la configuration locale. La figure suivante montre un exemple de départ d'un nœud leader :

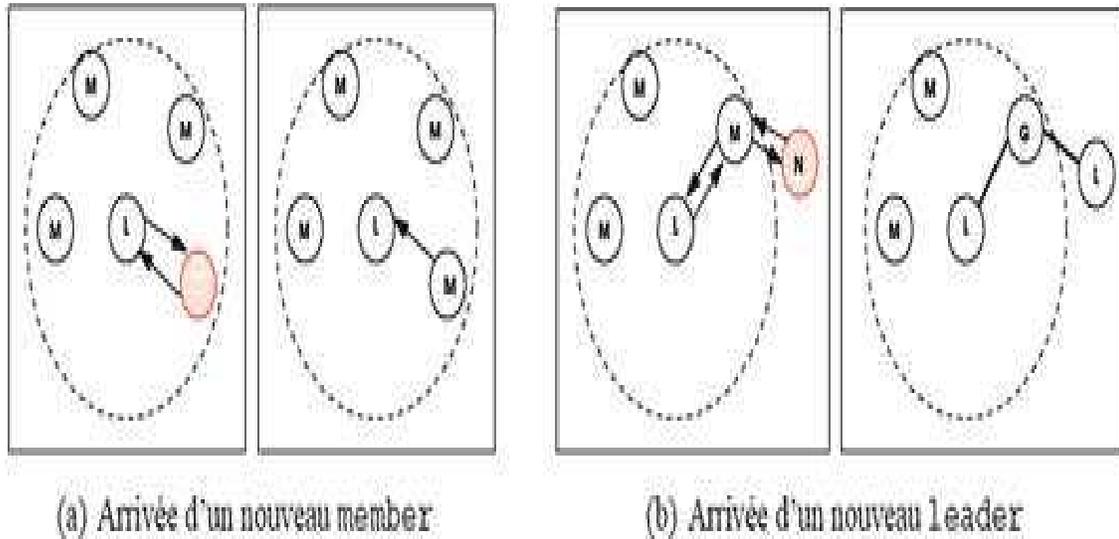


FIGURE 2.5 – Départ d'un leader

3. Gestion de partitions

à cause du caractère aléatoire de déploiement des nœuds, plusieurs partitions peuvent apparaitre : il est donc nécessaire d'appliquer des mécanismes de détection et de fusion de partitions ainsi que des mécanismes de scission dans le cas de la mort des nœuds.

(a) Scission de partition

La scission de partitions cherche à conserver les règles du backbone LEGOS et de maintenir la structure virtuelle de communication dans chaque partition :

- Si un member perd son leader, il réagira comme étant un nouveau nœud pour se rattacher à un autre leader.
- Les leaders restent ainsi après scission.
- Seuls les gateways demandent une réorganisation locale qui peuvent revenir dans l'état member s'ils ne sont plus connectés qu'à un seul leader.
- Après une scission, si un leader a perdu tous ses gateways, il génère un nouvel identifiant unique de partition et informe tous ses voisins du nouvel identifiant utilisé.

(b) Détection de partitions

Lors du déploiement du réseau, beaucoup de partitions peuvent être générées.

à chaque fois qu'un nouveau nœud se retrouve isolé, il devient leader et génère un nouvel identifiant de partition. Plus le réseau se densifie et plus des partitions sont amenées à fusionner. Ceux sont les nouveaux nœuds arrivant qui se charge de la détection de plusieurs partitions en écoutant le voisinage à 1 saut des paquets (Leader Broadcast Msg) émis par les leaders qui véhiculent l'identité des partitions ainsi que le nombre de leaders présents dans la partition.

(c) **Fusion de partitions**

Lors de la détection de plusieurs partitions, le nœud détecteur informe chacun des leaders de son voisinage et s'intègre dans la partition ayant le plus de leaders puis devient gateway. Il informe ensuite les autres partitions du nouvel identifiant résultant de la fusion. Cela nécessite une inondation dont le coût est contrôlé par la faible cardinalité du backbone LEGOS.

4. **Réorganisation locale**

Puisque l'énergie est au cœur de la problématique, cet événement a pour but d'optimiser sa consommation et d'augmenter la durée de vie du réseau, en faisant tourner les rôles de leader et gateway car dès qu'un nœud devient leader il prend en charge son voisinage, dépensant une grande partie de son énergie.

Les rôles de leader et gateway sont tournés suivant les propriétés de la topologie locale et une métrique locale à chacun des nœuds. Le but est de répartir la charge de communication entre les nœuds .

La métrique utilisée est basée sur l'énergie résiduelle et sur les propriétés du voisinage. Cette métrique M n'est appliquée que sur les leaders et gateways. M est défini par :

$$M(u) = \frac{E_{residual}}{E_{max}} * \frac{N_{member}}{N_{member} + 5} * \frac{N_{gateway} + 1}{N_{gateway}} \text{ pour les leaders} \quad (2.1)$$

$$M(u) = \frac{E_{residual}}{E_{max}} * \frac{N_{leader}}{N_{leader} + 1} \text{ pour les gateways} \quad (2.2)$$

Où :

- $E_{residual}$: représente l'énergie résiduelle pour le nœud,
- E_{max} : l'énergie maximum disponible,
- N_x : le nombre de nœuds de type x dans le voisinage.

L'idée de cette métrique est de tenir compte à la fois de la capacité en énergie qu'un leader ou gateway à tenir son rôle mais aussi de la capacité à trouver un autre leader ou gateway dans le voisinage (densité locale de leader et/ou de gateway). Dès que $M(u)$ atteint un seuil, le nœud (qu'il soit leader ou gateway) bascule dans l'état member après avoir diffusé, localement, une notification de réorganisation locale.

à la réception de cette notification, l'ensemble du voisinage à 1 saut de l'ancien leader bascule à nouveau dans l'état member pour une phase de découverte de voisinage, identification et/ou recherche de leaders. Tous les nœuds qui n'ont pu se rattacher à un leader informent l'ancien leader qui sélectionne ses successeurs à partir des informations locales qu'il possède

2.3.4 Les apports de LEGOS

1. **Auto-configuration d'adresses :**

L'auto-configuration :c'est un ensemble de mécanismes aboutissant à la reproduction d'adéquates configurations en fonction de la situation courante en termes des circonstances environnementales.

Deux stratégies sont à considérer :

- DHCP distribué où chaque nœud se retrouve en position d'allouer une adresse.
- Les solutions basées sur la détection d'adresses en conflit soit localement soit globalement.

Il s'agit alors de fournir un identifiant unique à un nœud du réseau pour ensuite mettre en place un routage, une localisation, etc. la difficulté de cette solution réside dans l'unicité de l'adresse. Ces solutions, entraînent un coût de vérification souvent trop important et une surconsommation en énergie, défavorable à la vie du réseau. C'est possible de développer une solution d'auto-configuration basée sur l'auto-organisation fournit par LEGOS et d'exploiter l'existence du backbone LEGOS stable pour diffuser des informations d'allocation d'adresses lorsque c'est nécessaire et tirer bénéfice de la faible cardinalité de l'ensemble des nœuds leaders. Le gain en latence, en complexité, en messages utilisés et, finalement, en énergie sera alors appréciable.

2. Gestion de puits mobiles et acheminement de données :

Le déploiement des nœuds est un facteur dépendant de l'application et de localisation du puits dans le réseau. L'introduction de puits multiples/mobiles dans les réseaux de capteurs permet de réduire le temps de réponse de l'application, de répartir la charge de trafic sur les nœuds du réseau, de router sur de plus courts chemins, et d'économiser l'énergie. De plus, différents puits peuvent avoir des intérêts de collecte de données différents.

Deux types de stratégies existent :

- Basée sur des rendez-vous : un ou plusieurs points de rendez-vous sont sélectionnés pour collecter les données, mais cela nécessite la connaissance d'informations de localisation.
- Orientée source : le puits diffuse une information de localisation lorsqu'il se déplace. Toutefois, la maintenance de cette information a un coût en messages important, et donc, entraîne une consommation énergétique forte. C'est possible d'apporter une réponse simple et efficace à la problématique des puits multiples et mobiles, en se basant sur l'auto-organisation LEGOS. En effet lors de la phase de déploiement du réseau, l'auto-organisation LEGOS se met en place, indépendante de toute notion de localisation, indépendante des puits, apparaît comme une structure non orientée. C'est à travers cette topologie logique que la prise en compte des puits va être faite : le support des requêtes, l'acheminement des informations depuis le puits vers les nœuds et depuis les nœuds vers le puits ainsi que la conservation des données remontées par les capteurs si aucun puits n'est connu.

3. Agrégation de données :

L'agrégation est un mécanisme clef pour les réseaux de capteurs. L'agrégation de données permet de réduire le nombre de transmissions et permet également une économie d'énergie et une robustesse du mécanisme de routage. Deux types d'agrégations sont à considérer :

- l'agrégation spatiale où l'on cherche à combiner des informations venant de plusieurs sources.
- l'agrégation temporelle venant de différentes périodes de temps. L'agrégation se fait sur le chemin allant du capteur vers son puits. Les solutions d'agrégation existantes reposent sur une topologie en arbre dont la construction est initiée par le nœud puits en inondant le réseau avec un message d'initialisation et la maintenance de l'arbre requiert l'utilisation de paquets hello ce qui rend ces solutions non efficaces en énergie.

LEGOS fournit une topologie en arbre initialement construite lors du déploiement du réseau. L'arbre LEGOS, peut être utilisé pour développer un nouveau mécanisme d'agrégation temporelle de données.

2.4 Conclusion

Dans ce chapitre, nous avons défini la notion de l'auto-structuration et montré son importance dans les réseaux mobiles et particulièrement dans les RCSFs. On a conclu que les capteurs sans fil sont généralement difficiles, ou plus tôt impossible à installer manuellement, parce que ils sont déployés dans des environs inaccessibles. Alors, les capteurs doivent collaborer entre eux et sans intervention externe pour émerger une structure donnée. Donc, le choix d'un système auto-structuré est alors primordial au

réseaux de capteurs sans fil pour garantir une grande autonomie de réseau. Par conséquence, on peut considérer qu'un réseau de capteurs, est un système auto-structuré car tous les protocoles utilisés sont basés sur des interactions locales entre les nœuds.

Pour bien comprendre son impact dans les réseaux mobiles, nous avons présenté un protocole d'auto-organisation. LEGOS est un protocole d'auto-organisation asynchrone orienté évènement, s'exploite sur trois catégories de nœuds(Leader, Gateway et Member) et destiné aux réseaux Ad-hoc et RCSFs. Son objectif principal est de fournir une auto-organisation simple, efficace et adapté aux topologies à larges échelles.

Chapitre 3

Protocoles d'auto-structuration dans les RCSFs

3.1 Introduction

Les réseaux de capteurs sans fil, peuvent être vus comme des systèmes distribués dont l'objectif principal est de collecter les données capturées afin de les exploiter. Ils sont destinés à être utilisés dans des environnements hostiles où l'intervention humaine est impossible ou très dangereuse. Dans ce contexte ; développer des applications autonomes, tolérantes aux différentes défaillances qui peuvent être produites, et avec un coût minimal présente une direction de recherche très attractive.

Quant à l'autonomie des applications qui fait l'objet de notre travail, elle se base sur l'auto-structuration et l'auto-organisation. L'auto-structuration représente la capacité d'un système à créer une structure spécifique, en partant de rien, et sans apport d'information extérieure. C'est une dimension importante des systèmes autonomes, par exemple pour le passage à l'échelle. La complexité d'une telle structuration dépend de la quantité d'information dont les entités (ou nœuds) disposent initialement. L'auto-structuration signifie la structuration d'un réseau pour rendre son exploitation plus aisée, et l'association d'un ensemble de fonctionnalités permettant de maintenir cette structure et faciliter son utilisation. Elle est le résultat d'application d'algorithmes qui organisent le comportement global d'un système.

3.2 L'auto-structuration avec GPS et puissance de signal

De nombreuses techniques ont été proposées dans la littérature pour calculer les positions des nœuds capteurs. Nous étudions une technique plus utilisée aujourd'hui, en se basant sur la puissance de signal GPS pour estimer la distance entre les nœuds.

3.2.1 Historique du système GPS

En 1968, le Pentagone imagine un système de localisation géographique composé d'une constellation de satellites en orbite autour de la Terre qui pourrait leur fournir la position d'un point sur la planète en temps réel et 24h sur 24. Le système NAV.S.T.A.R-G.P.S (navigation system time and ranging - Global Positioning System) a alors été conçu, financé et développé dès 1973 par le département de la Défense des Etats-Unis pour un usage strictement militaire. Le premier satellite a été lancé en 1978. Le système a été déclaré opérationnel en 1995, comportant alors 28 satellites en orbite quasi-circulaires autour de la Terre formant un angle de 55° avec l'équateur et de 60° avec chacune des autres orbites

à une altitude de 20200 Km, et un segment de contrôle. Depuis 1995, la constellation comporte 24 satellites « titulaires » et 4 satellites de réserves en cas de disfonctionnement, tous opérationnels et les performances du système sont stables. Ils ont une période de révolution de 1 jour sidéral (11 h 58 min). Ainsi, une constellation peut être observée 2 fois par jour et chaque jour 4 min plus tôt que le jour précédent.

Le principe de fonctionnement Le principe du positionnement GPS est très proche du principe de triangulation. La constellation a été conçue de telle manière que partout sur Terre, on puisse voir au moins 4 satellites à tous moments. Les récepteurs fonctionnent sur le principe de la télémétrie : On mesure la distance entre l'utilisateur et un certain nombre de satellites de positions connues. On définit ainsi des sphères centrées sur des satellites et dont l'intersection donne la position. Le récepteur GPS est capable d'identifier le satellite qu'il utilise à l'aide du signal pseudo aléatoire émis par chaque satellite. Il charge, à l'aide de ce signal, les informations sur l'orbite et la position du satellite. Pour mesurer la distance qui sépare le satellite du GPS, on mesure le temps T mis par le signal pour aller de l'un vers l'autre[29].

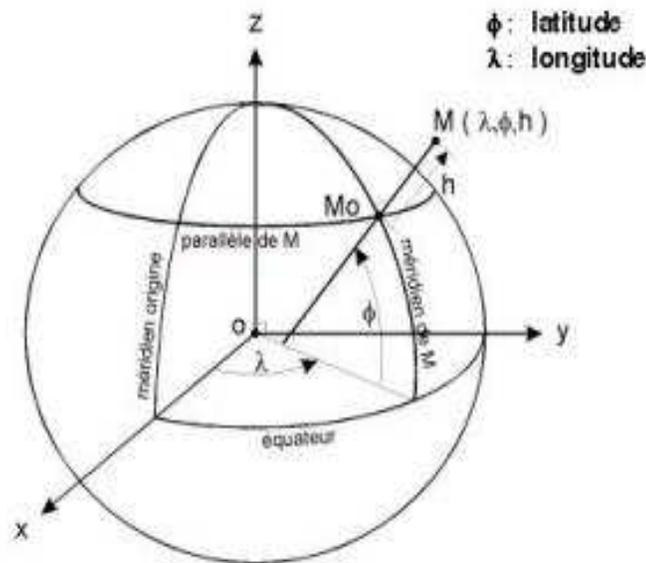


FIGURE 3.1 – Principe de GPS

Principe de calcul des coordonnées géographiques La position de points disposés à la surface terrestre et qui sont connues en coordonnées cartésiennes tridimensionnelles géocentriques ou géographiques, ne peut être représenté que sur un globe, ce qui est très encombrant et peu maniable sur le terrain. Aussi pour beaucoup d'opérations, il faut arriver à une représentation cartographique plane, permettant de travailler sur papier que l'on appelle projection plane. Le passage de ces coordonnées fastidieuses à des coordonnées rectangulaires planes en 2 dimensions, suppose de ramener tout d'abord tous les points de la surface réelle terrestre sur un ellipsoïde de référence, et au final, de l'ellipsoïde à un plan. L'ellipsoïde n'étant pas une surface développable sur le plan, il y aura nécessairement des déformations au niveau des distances, car toutes les projections déforment les longueurs. Dans la pratique, on définit une surface développable autour de l'ellipsoïde (plan, cône, cylindre) et on détermine

une transformation amenant le point de l'ellipsoïde sur cette surface : c'est le système de projection.

3.2.2 Algorithmes d'auto-structuration centralisé

Dans les réseaux ad hoc, la consommation de l'énergie a été considérée comme un facteur déterminant mais pas primordial parce que les ressources d'énergie peuvent être remplacées par l'utilisateur. Ces réseaux se focalisent plus sur la QoS que sur la consommation de l'énergie.

Le GPS [7] résout le problème de la localisation dans les environnements extérieurs. Cependant, pour les larges réseaux de capteurs où les nœuds doivent être de très petite taille, peu consommateurs et peu onéreux, équiper chaque unité avec une puce GPS est trop coûteux. En outre, la précision des puces GPS pour capteurs reste encore assez faible. De nombreuses techniques sont proposées pour permettre aux nœuds d'estimer leur position.

[19] distingue deux types de stratégie de localisation : les **localisations fines** et les **localisations approximatives**. Les localisations fines déterminent de manière précise la distance entre deux nœuds ou les coordonnées du nœud dans le réseau tandis que les localisations approximatives spécifient une surface, des coordonnées virtuelles, etc.

Localisations fines L'utilisation du GPS permet de résoudre le problème de la localisation en extérieur sous certaines conditions et hypothèses. En effet, son coût, à la fois économique et énergétique, pose soucis. Dans les réseaux de capteurs denses, sa précision (de l'ordre d'une dizaine de mètres) peut également poser problème. Quelques articles contournent le problème et proposent l'utilisation d'ancres : seulement certains nœuds connaissent leur position précise et permettent autres nœuds du réseau, par triangulation ou multi-latération de connaître la leur. Pour cela, plusieurs solutions sont proposées :

- ToA (Time of Arrival)[30]. Sachant la vitesse de la propagation du signal dans l'air et l'instant où il a été envoyé, il est possible de déterminer avec précision la distance entre un récepteur et un transmetteur. En utilisant plusieurs nœuds de références, le nœud est capable de déterminer sa position relative par triangulation. Ce mécanisme, utilisé dans la localisation GPS, présente cependant une faille : pour être précis, il nécessite une synchronisation forte et une horloge interne stable (un décalage de 1μ s entraîne une erreur de 300 m dans le cas d'un signal radio). Des solutions existent pour améliorer la précision en réduisant les imprécisions d'horloges mais elles nécessitent plus de nœuds de référence, des calculs importants ou des spécificités matérielles.
- TDoA (Time difference of Arrival)[1]. Pour pallier l'obstacle de la synchronisation, le TDoA utilise deux signaux de différentes natures (ultrasons et radio par exemple), pour estimer la distance entre 2 nœuds. Connaissant le temps de propagation de chacun des signaux, la différence de temps d'arrivée va permettre de calculer la distance sans que la synchronisation ne soit nécessaire. Cependant, chaque nœud doit être équipé d'une antenne pour chaque signal et d'une horloge précise ce qui n'est pas le cas pour les capteurs actuellement sur le marché. De plus, cette solution entraîne un surcoût énergétique causé par la réception et l'envoi des deux signaux et la gestion de deux modules radio.
- AoA (Angle of Arrival). Chaque capteur est équipé d'un ensemble d'antennes. Ils sont donc capables de connaître l'angle d'arrivée d'un signal reçu. Cette information, obtenue de plusieurs nœuds voisins, va permettre d'appliquer une triangulation et le nœud sera capable d'obtenir une distance relative. Là encore, la consommation des capteurs est augmentée et les spécifications des capteurs présents sur le marché ne correspondent pas aux hypothèses initiales : ils ne possèdent qu'une seule antenne omnidirectionnelle.

- Combinaison de TDoA et de AoA[5]. Afin d'améliorer la précision et d'étendre la localisation à des environnements 3D. Les auteurs conçoivent une solution hybride en utilisant l'AoA pour améliorer la technique utilisée en 2D.

L'ensemble de ces protocoles ne prend pas en compte la consommation énergétique et le coût des nœuds avec comme hypothèses des capteurs suréquipés. En outre, le système des ancres n'évite pas le problème de la localisation mais ne fait que le réduire à un sous-ensemble des nœuds du réseau. En outre, il fait apparaître d'autres problèmes comme le placement optimal de ces ancres dans le réseau, pour permettre une meilleure localisation des nœuds [2, 9, 19]. Enfin, de nombreuses hypothèses existent concernant la synchronisation et le besoin de connaître le temps exact d'émission qui sont difficilement conciliables avec un réseau de capteurs dense où la contention et les interférences peuvent être importantes.

Localisations approximatives D'autres stratégies consistent à trouver des coordonnées approximatives en déterminant imparfaitement la distance entre deux capteurs. Selon l'application, une localisation approximative des capteurs peut être un bon compromis et dans ce cas plusieurs approches sont possibles :

- Le système des badges actifs[3]. Dans ce système chaque nœud transmet un paquet hello périodique toutes les 10 secondes. Le signal est reçu par des ancres fixes placées à des endroits précis à l'intérieur d'un bâtiment. Ce sont ces ancres qui vont relayer l'information jusqu'au puits. La position d'un capteur est associée à la position de l'ancre la plus proche.
- L'intersection des signaux[26]. On considère également un réseau non homogène avec la présence régulière d'ancres. Chaque capteur compte le nombre d'ancres dont il reçoit le signal. En prenant comme hypothèse la connaissance de la portée radio des ancres et leur position, le capteur est en mesure de déterminer une zone de localisation (voir figure 3.3). Cette zone va correspondre à l'intersection théorique des disques de portée radio des ancres. Il est donc nécessaire ici d'avoir un nombre suffisant d'ancres connaissant leur position afin que tous les capteurs soient à portée directe d'au moins une ancre et de plusieurs pour avoir une meilleure précision.
- L'algorithme de localisation Monte-Carlo (MCL)[21]. Les auteurs se servent de la mobilité des nœuds dans le réseau pour affiner leur localisation. Ils utilisent la méthode séquentielle de Monte-Carlo pour calculer des probabilités de distribution des positions possibles des nœuds. En fonction de la position précédente des nœuds et des observations provenant d'ancres déployées dans le réseau, l'algorithme est capable de filtrer les positions impossibles.
- Le système de coordonnées virtuelles[28]. Chaque nœud détermine sa distance aux ancres en nombre de sauts et détermine ainsi ses propres coordonnées dans un système, dont la dimension dépend du nombre d'ancres. Le routage devient possible en suivant le système de coordonnées ainsi établi (voir figure 3.2).

Ces protocoles sont peu adaptés aux réseaux de capteurs car ils nécessitent soit une architecture fixe préalable, ce qui interdit le déploiement des capteurs dans des zones inaccessibles ou dangereuses soit des calculs centralisés qui requièrent des communications régulières avec le puits de la part de l'ensemble de la topologie. Ce dernier mode de fonctionnement va être potentiellement très consommateur en énergie et peu résistant à un passage à l'échelle.

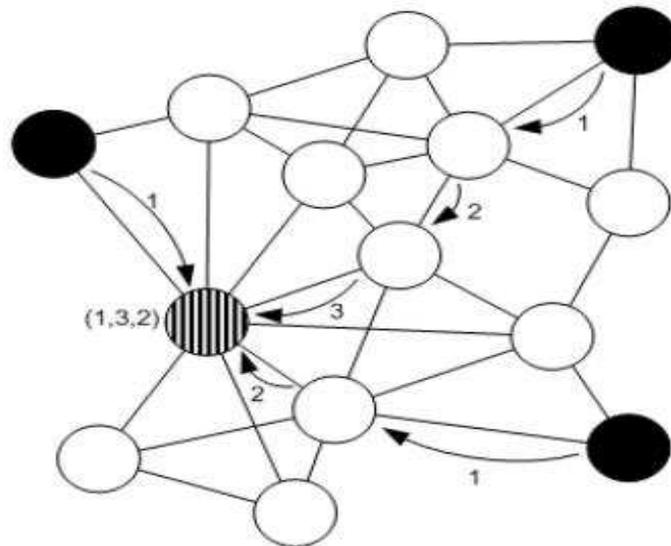


FIGURE 3.2 – Exemple d'une localisation approximative : coordonnées virtuelles

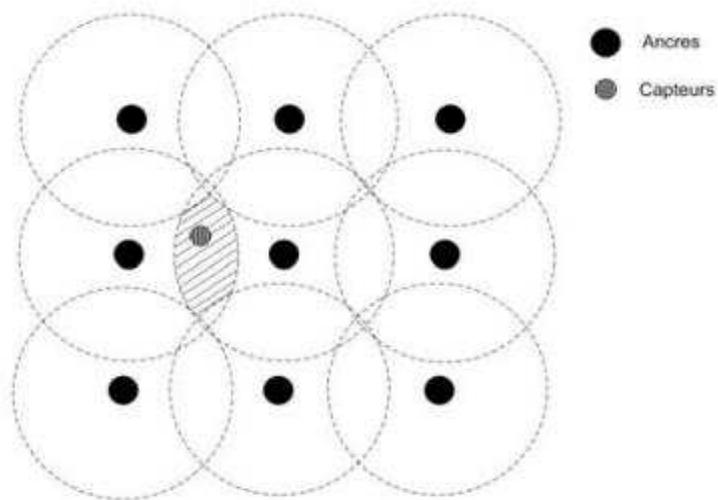


FIGURE 3.3 – Exemple d'une localisation approximative : intersection de signaux

3.3 Quelques techniques de localisation dans les réseaux mobiles

Le rôle des réseaux de capteurs étant souvent la surveillance d'un espace donné. De ce fait un des points important est celui de la localisation entre cellules plus précisément les méthodes de localisation. Dans la majorité des cas les cellules sont repartis dans l'environnement de manière aléatoire ou sont en mouvement quasi constant. Il faut qu'une cellule puis savoir l'emplacement des autres dans cet environnement pour qu'il puisse transmettre de manière localisée ou qu'il puisse savoir de quel nœud vient les flux d'information reçus.

Il est vrai qu'il existe déjà des systèmes et des méthodes de localisation comme par exemple le GPS mais il n'est pas adapté aux applications visé par les RCSFs. Car il a des handicaps non négligeables. Il est disponible qu'a l'extérieur et est très sensible aux obstacles et aux zones étroites. Sans parler des problèmes liée au coût du au nombre important de capteur que demande certain application. Enfin le paramètre le plus problématique est la consommation d'énergie lors des transmissions qui est bien trop importante ce qui ne correspond pas à la mentalité des RCSFs. Donc une localisation par des méthodes et algorithmes les plus optimisés possibles est nécessaire.

3.3.1 Localisation collaborative

Comme nous avons mentionné précédemment pour le GPS, la position absolue d'un appareil sans fil peut être déterminée par des satellites. La même approche peut être utilisée pour estimer la localisation dans les systèmes sans fil plus petite échelle : Si un nœud peut estimer la distance de trois ou plusieurs nœuds d'ancres (les position de ces nœuds sont déjà connues), sa position peut être calculée en utilisant un système similaire à celle utilisée par les récepteurs GPS. Cette approche est conceptuellement bien défini, mais elle peut échouer dans les applications à large échelle où les nœuds ont une capacité limitée de communication et capture. Par exemple, dans les applications sans fil à faible consommation d'énergie et d'autres applications ad-hoc, il pourrait être impossible de garantir une couverture suffisante avec un tel nombre d'ancres pour estimer la position de chaque appareil dans le réseau. Dans tels scénarios, les nœuds peuvent compenser le manque de dispositifs de référence en prenant des mesures avec d'autres nœuds homologues en position inconnue. Si tous les nœuds participent à cet effort de collaboration, les données collectées peuvent être utilisées pour obtenir la position de chaque nœud dans le réseau (voir la figure 3.4).

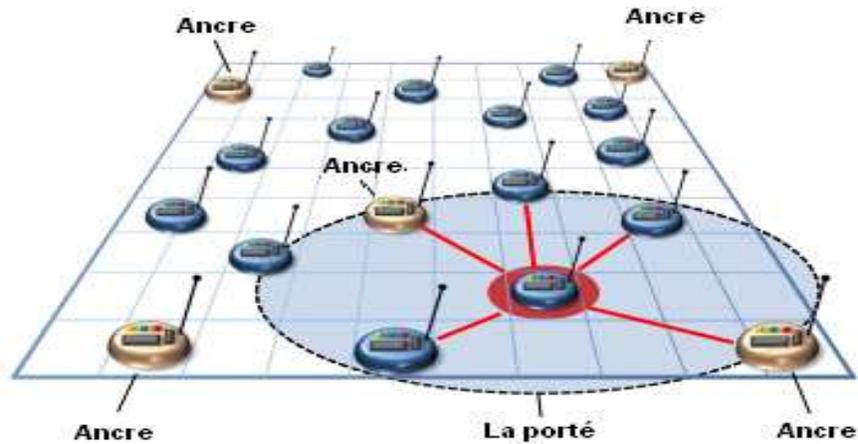


FIGURE 3.4 – La localisation Collaborative dans les RCSFs et Ad-hoc.

Les techniques de localisations range-based et range-free

Les solutions existantes de collaboration peuvent être regroupés dans *range-based*[27][24][32] et *range-free*[26][31][12] en fonction du type de mesures utilisées . Avec **range-based**, en calculant la distance entre les paires de nœuds(l'angle de l'arrivé) pour mettre en œuvre la localisation physique, cette information est généralement obtenue en augmentant chaque nœud avec matériel spécialisé comme les émetteurs-récepteurs à ultrasons pour la mesure de distance, ou des antennes directionnelles pour estimer l'angle d'arrivé. La technique **range-free** s'appuie seulement sur des informations de proximité, c'est à dire savoir si deux nœuds sont proches dans l'espace ou non. Bien que cette solution puisse seulement fournir des résultats à grain grossier, les contraintes de proximité sont aisément disponibles par la détection des phénomènes physiques communs tels que le bruit et la lumière, ou par l'échange des messages radio. D'autre part, la technique range-based utilise des mesures d'arrangement telles que le temps d'arrivé (ToA), angle d'arrivée (AoA), indicateur de la force de signal reçu (received signal strength indicator RSSI) et le temps différé d'arrivée (time difference of arrival TDoA) pour mesurer les distances entre les nœuds afin d'estimer la localisation des nœuds. Ces différentes techniques sont décrites comme suit.

Temps d'arrivé(Time of Arrival ToA)

La technique de temps d'arrivé(ToA), tous les nœuds transmettent un signal avec une vitesse prédéfini à leurs voisins. Puis, chacun nœud envoie un signal d'acquiescement à ses voisins, et en utilisant les temps de transmission et de réception, chaque nœud estime la distance entre lui et son voisin[16] .

Indicateur de la force signal reçu (Received Signal Strength Indicator RSSI)

RSSI est défini comme la quantité de puissance présenté dans le signal radio reçu. la force de signal reçue (RSS) diminue à cause de l'augmentation de la distance de propagation(l'atténuation de signal). Par conséquent, la distance entre deux nœuds capteurs peut être comparée en utilisant la valeur de RSS au récepteur, en supposant que la puissance de transmission à l'expéditeur est fixée ou connue[16].

Un avantage de cette technique est : que les nœuds capteurs exigent des dispositifs standards et n'acceptent aucun matériel additionnel, aucun impact significatif à la consommation d'énergie ou taille du réseau. L'inconvénient majeur de cette technique est l'inexactitude. Par exemple, si le réseau de capteur est déployé à l'intérieur des murs et d'autres obstacles, ces derniers vont réduire sévèrement la précision de cette méthode due aux non-linéarités, au bruit, à l'interférence et à l'absorption.

Le temps différé d'arrivée (Time Difference of Arrival TDoA)

La technique de TDoA exige aux nœuds de transmettre deux signaux de différentes vitesses. Dans cette technique, chaque nœud est équipé d'un microphone et d'un parleur. La plupart des systèmes utilisent l'ultrason tandis que certains utilisent des fréquences audibles. Dans TDoA, un message radio est envoyé par l'émetteur, qui attend alors un certain intervalle fixe de temps et produit après un échantillon fixe des gazouillements sur son parleur. En mode d'écoute, les nœuds entendent le signal radio et notent le temps courant, et après ils mettent en marche leurs microphones pour détecter l'échantillon de gazouillements et notent encore le temps courant. Par cet intervalle de temps, les nœuds peuvent calculer la distance entre eux et l'émetteur en profitant le fait que les ondes radio traversent beaucoup plus rapidement que le bruit en air[16]. L'inconvénient de cette technique est qu'elle exige le matériel spécial qui doit être doté dans les nœuds capteurs. En outre, la vitesse du son en air change avec la température de l'air et l'humidité, qui peuvent présenter des inexactitudes. Enfin, il est très difficile de rencontrer des conditions de line-of-sight (vue libre) dans beaucoup d'environnements tels que à l'intérieur des bâtiments ou dans les terrains montagneux.

angle d'arrivée (AoA)

la technique AoA recueillent les données en utilisant les rangées de radio ou de microphone. Ces rangées permettent à un nœud récepteur de déterminer la direction du nœud transmetteur. Des techniques de communication optique peuvent également être employées pour recueillir des données d'AoA. Dans ces techniques, un seul signal transmis est entendu par plusieurs microphones séparés. Donc la phase ou le temps différé entre l'arrivée des signaux dans les différents microphones est calculée et ainsi l'angle d'arrivée du signal est trouvé.

Cette technique est généralement précise, mais l'inconvénient est que le matériel d'AoA est plus grand et plus cher que le matériel de TDoA, puisque chaque nœud doit avoir un parleur et plusieurs microphones. Un autre facteur important est le besoin de séparation spatiale entre les parleurs elle sera difficile à adapter, car la taille des nœuds de capteur très petite[16].

En conclusion, les techniques range-based peuvent fournir des résultats très précis mais exigent le matériel cher, tel que des dispositifs d'ultrasons pour TDoA et des rangées d'antenne pour AoA. Un inconvénient des techniques range-based est que l'information de distance peut être difficile à obtenir dans la pratique telle que la présence des obstacles qui empêchent le line-of-sight.

Les techniques de localisations Anchor-Based versus Anchor-Free

Une autre classification des algorithmes de localisation pour les RCSFs est basée sur des nœuds de référence (les références de ces nœuds est obligatoirement non externes). Ces nœuds, appelés des nœuds d'ancre (ou simplement les ancres comme abréviation), habituellement ont un récepteur de GPS installé sur eux ou savent leur position par configuration manuelle. Ces nœuds sont utilisés par d'autres nœuds comme des nœuds de référence afin de fournir des coordonnées dans le réseau.

Les algorithmes de Anchor-based [35][34][4][22] utilisent les nœuds d'ancre pour tourner, traduire et des fois balancer les coordonnées relatifs du système de sorte qu'il coïncide avec les coordonnées absolu

du système. Dans tels algorithmes, une fraction des nœuds doit être des nœuds d'ancre ou au moins un nombre minimum de nœuds d'ancre sont exigés pour à résultats adéquats. Pour les espaces à deux dimensions, au moins trois nœuds d'ancre non situés sur la même direction et pour les espaces à trois dimensions, au moins quatre nœuds non coplanaires d'ancre sont exigés. Les coordonnées finales attachées aux nœuds capteurs sont valides et conformes aux coordonnées globales du système. L'inconvénient de ces algorithmes est qu'un autre système de positionnement est exigé pour déterminer les positions des nœuds d'ancres. Par conséquent, si l'autre système de positionnement est indisponible, par exemple, si les ancres de GPS-based (basé au principe de GPS) situent dans des secteurs où il n'y a une vue claire du ciel, l'algorithme peut ne pas fonctionner correctement. Un autre inconvénient dans les algorithmes anchor-based est que les nœuds d'ancres présentent un coût non négligeable car ils exigent habituellement d'un récepteur de GPS. Par conséquent, les algorithmes qui exigent beaucoup de nœuds d'ancre sont très coûteux et consomme beaucoup plus d'énergie. L'information de localisation peut également être difficile à coder par les nœuds d'ancre, cependant, dans ce cas-là un déploiement rigoureux des nœuds d'ancre est exigé, qui peuvent être très chers ou même impossibles dans les terrains inaccessibles.

En revanche, les algorithmes de localisation ancre-libres[25][15] n'exigent pas des nœuds d'ancre. Ces algorithmes fournissent seulement des positions relatifs des nœuds. Pour quelques applications, les coordonnées relatifs sont suffisantes, par exemple dans les protocoles de routages géographiques, le prochain nœud d'expédition est choisi facilement, en se basant sur la métrique de distance qui exige au prochain hôte d'être physiquement plus près de la destination, qui peut être parfaitement exprimée avec des coordonnées relatives.

3.4 L'auto-structuration avec des informations locales

Dans les réseaux de capteurs, les problèmes sont principalement liés à l'autonomie résultant du coût des communications. Ces coûts sont importants lors de l'envoi des données collectées par un capteur à la station de base ou par l'envoi d'une requête à tous les capteurs par la station de base. Ainsi, le choix d'une technique d'auto-structuration est alors primordial pour garantir une grande autonomie pour ces réseaux qui sont généralement déployés dans des zones hostiles. Cette technique doit être capable de diffuser et acheminer une information sans perdre trop d'énergie. De ce fait, pour le traitement du problème de la structuration dans les réseaux ad hoc et de capteurs, plusieurs approches ont été décrites dans la littérature. Ces approches se basent essentiellement sur le clustering, l'ensemble dominant connecté (CDS) et la diffusion dépendant de la source. Cependant, les techniques d'acheminement de l'information dédiées particulièrement aux réseaux de capteurs doivent tenir compte de la spécificité de ces réseaux et du type de communications induit par l'application.

Dans les architectures à plat, la majorité des protocoles de routage et de diffusion conçus pour les réseaux de capteurs ou les réseaux ad hoc de petite ou moyenne taille avec une faible mobilité de nœuds fournissent de bonnes performances. Cependant, lorsque le nombre de nœuds augmente ou que les nœuds sont mobiles, le trafic de contrôle prédomine les communications réelles. Ce qui conduit à une augmentation de la latence et à une explosion des tables de routage. Afin de pallier à ces limites, l'architecture hiérarchique s'est considérée comme l'une des solutions communément efficaces pour la diffusion et le routage dans les réseaux ad hoc et de capteurs. Elle consiste à regrouper les nœuds proches géographiquement en groupes aussi appelés "clusters" et d'établir des schémas de routage différents à l'intérieur des clusters (intra-clusters) et entre les clusters (inter-clusters). Chaque cluster est représenté par un nœud particulier appelé cluster-head. Ce nœud est élu comme cluster-head selon une métrique spécifique ou une combinaison de métriques. Il est responsable de la coordination entre les différents membres de son cluster, pour agréger leurs données collectées et les transmettre vers la

station de base. Dans un cluster, chaque nœud stocke la totalité des informations de son cluster et une partie des informations des autres clusters; ce qui minimise considérablement la taille des tables de routage et le nombre de messages échangés dans le réseau.

La notion d'auto-structuration prend tout son sens dans le cadre des réseaux ad hoc et de capteurs. Elle offre un cadre pour faciliter la configuration des nœuds et la mise en place de protocoles de communications. Dans un réseau auto-organisé, l'intervention humaine est réduite au minimum ce qui facilite son déploiement. L'auto-structuration doit donc faire émerger une structure globale. On peut considérer qu'un réseau de capteurs, est un système auto-structuré car tous les protocoles utilisés sont basés sur des interactions locales entre les nœuds et sont répartis sur l'ensemble des membres du réseau. De notre point de vue, il ne s'agit pas de se limiter uniquement à la configuration d'adresses ou au routage mais bien de fournir une structure virtuelle dynamique et unique permettant de faciliter le déploiement de tout protocole réseau.

Dans cette section, nous présentons et nous analysons ces approches pour tirer profit de leurs avantages et d'éviter leurs limitations.

3.4.1 Comment évaluer une auto-structuration ?

Dans cette section nous allons évaluer un système auto-structuré, puisque la restructuration consiste à s'adapter face à des situations imprévisibles. Alors la question qui se pose dans cette partie est "comment évaluer une auto-organisation"[13]. Pour évaluer un système on doit prendre quelques métriques pour son évaluation :

Au sens réseau et protocoles de communication : il est possible de caractériser une stratégie d'auto-organisation par le surcoût protocolaire, voisinage, etc.

Au sens algorithmique distribué : Complexité en messages, en temps, en mémoires, etc.

Auto-stabilisation : s'intéresse à la convergence d'algorithmes distribués et permet de montrer qu'en présence de fautes, les algorithmes convergent vers un objectif identifié, etc.

La capacité : en termes de flots induite par dans le réseau.

L'entropie : L'entropie en thermodynamique, permet de mesurer l'ordre : (2.1)

$$\xi = k_b \sum_i p_i \log p_i$$

Dans le cas des RCSFs on s'intéresse aux liens actifs, L'entropie d'un lien est donnée par : (2.2)

$$E = -p(u, v) \log p(u, v)$$

Où $p(u, v)$ est la probabilité qu'un lien entre u et v existent. L'entropie globale est donnée par : (2.3)

$$E = \sum_{u, v \in X} -p(u, v) \log p(u, v)$$

Plus l'entropie est importante plus il y'a de désordre et plus l'entropie est faible plus il y'a de l'ordre.

3.4.2 Apports de l'auto-structuration

Nous situons dans cette section quelques avantages apportés de l'auto-structuration. La localisation, collaboration, auto-stabilisation, passage à l'échelle et dynamique, sont des propriétés communes à tous les réseaux, et la persistance, adaptation et flexibilité d'utilisation, sont des propriétés spécifiques aux concepts des réseaux mobiles Ad-hoc ou capteurs.

Localisation

Commençons par la distinction entre réseaux centralisés, distribués et localisés. Un réseau est dit centralisé s'il possède a priori une connaissance complète. Par exemple, un protocole de routage centralisé possède une vue complète de la topologie et des informations sur toutes les arêtes et nœuds. Il peut donc prendre une décision optimale. Les réseaux distribués permettent de prendre une décision en distribuant la charge, en échangeant des messages pour partager l'information requise. Ils sont par définition beaucoup mieux adaptés aux réseaux ad hoc, possédant un grand nombre de nœuds devant chacun prendre des décisions cohérentes. Enfin, un réseau localisé est un réseau distribué n'échangeant des informations qu'avec des nœuds à une distance bornée de lui, et éventuellement avec un nombre borné de nœuds quelconques du réseau.

Coopération

Un ensemble de décisions localisées doit amener à un comportement global émergent. Ainsi, si nous reprenons une définition empruntée à la cybernétique : le tout est plus que la somme des parties [14]. Chaque nœud va appliquer une règle sur l'ensemble des informations desquelles il dispose afin d'en tirer une décision. à l'échelle du réseau, nous pouvons dire que le réseau prendra une décision globale permettant de faire émerger une structure d'auto-organisation.

Auto-stabilisation

Un réseau est dit auto-stabilisant si, partant d'un état quelconque, il converge vers un état légal en un temps fini. Une telle propriété d'auto-stabilisation est requise dans les réseaux mobiles(ad hoc ou capteur) afin que le réseau converge vers un espace d'états légaux même en cas de faute permanente . Ainsi, si un algorithme auto-stabilisant est exécuté sur un système lui aussi auto-stabilisant, le concepteur peut être certain que l'ensemble des nœuds converge vers un état stable en un temps borné. Le but est naturellement de minimiser ce temps de convergence.

Passage à l'échelle

Tout système pour être efficace ne doit pas voir ses performances baisser de façon drastique si le nombre de nœuds constituant le réseau augmente. Soit λ_i un paramètre (la mobilité, le nombre de nœuds...), un protocole P , et $P(\lambda_1)$ les performances associées à la valeur λ_1 du paramètre λ_i . Alors, le facteur de passage à l'échelle de P par rapport à λ_i est [9] :

$$\psi \leq \lim_{\lambda_i \rightarrow \text{inf}} \frac{\log p(\lambda_i, \lambda_2 \dots)}{\lambda_i}$$

On dit d'un protocole qu'il passe à l'échelle si $\psi_{\lambda_i} = 0$, quelles que soient les performances mesurées. Le but étant idéalement de concevoir un protocole passable à l'échelle vis à vis de la cardinalité du réseau en fonction de tout paramètre. De façon pratique, si nous observons sur un graphe les performances de l'algorithme en fonction du paramètre λ_i , la courbe ne devrait pas s'approcher asymptotiquement de l'axe des abscisses.

Robustesse à la dynamique du réseau

Puisque chaque mobile est libre de se déplacer indépendamment des autres, la topologie est continuellement changeante. En conséquence, un algorithme doit s'adapter aux changements de topologies. Il peut fonctionner en mode dégradé pendant le laps de temps nécessaire à la mise à jour de ses données : plus cet intervalle de temps est faible, plus le protocole réagit rapidement aux changements, et meilleure est sa robustesse. Puisqu'un terminal ne possède, comme décrit précédemment, que des

informations localisées, des incohérences dans la vue du réseau peuvent survenir. C'est pourquoi un protocole de routage par exemple peut voir une de ses routes casser du fait de l'absence d'une mise à jour immédiate. Le but du concepteur est donc de réduire ces incohérences et/ou son impact. Il est important de noter qu'un algorithme peut optimiser sa robustesse en proposant des règles permettant de bien réagir aux incohérences : même si la vue est incohérente, la décision prise n'engendrera aucun conflit.

Persistance

La structure d'auto-organisation doit être stable dans le temps. La structure virtuelle, la hiérarchie sera utilisée ultérieurement pour l'adressage, le routage. . Il est donc fort probable qu'un nœud nouvellement élu soit contraint d'échanger des informations avec les autres nœuds du réseau pour mettre à jour des informations de routage par exemple. De même, il se peut qu'un changement dans la structure virtuelle crée des changements de routes. Les nœuds constituant la structure virtuelle doivent dans la mesure du possible rester inchangés.

Flexibilité d'adaptation

Un réseau ad-hoc se meut dans un environnement variable. Ainsi, la densité, la mobilité ou le nombre de nœuds peut évoluer avec le type d'application ou même avec le temps. Les algorithmes proposés doivent donc pouvoir être aisément paramétrables afin de les adapter aux conditions environnementales. Idéalement, une telle adaptation devrait être automatique et dynamique. [?] identifie 3 niveaux d'adaptation :

1. 1^{er} niveau : l'algorithme doit s'adapter aux changements de topologie (comme décrit dans un des paragraphes précédents).
2. 2^{ieme} niveau : l'algorithme doit adapter la valeur de ses propres paramètres. Par exemple, lorsque le degré du réseau augmente, un protocole peut par exemple choisir de minimiser les informations échangées avec les voisins directs.
3. 3^{ieme} niveau : l'algorithme doit être remplacé par un autre algorithme plus efficace dans ces conditions. Par exemple lorsque les changements de topologie sont trop importants, un mécanisme d'inondation aveugle peut être mis en place.

Flexibilité d'utilisation

Comme nous le décrivions dans le paragraphe précédent, cette structure d'auto-organisation nous sera utile pour déployer tout service réseau. En conséquence, elle doit être assez générique pour remplir tous les objectifs de routage, d'allocation de bande passante. . . Nous verrons dans les chapitres suivants que les concepts proposés dans ce chapitre peuvent être utilisés pour l'économie d'énergie, pour le routage dans un réseau ad-hoc ou RCSF, pour la localisation d'un nœud dans un réseau cellulaire multi-sauts.

3.5 VINCOS(Virtual networked coordinate system)

VINCOS est un système adaptatif de coordonnées virtuelles proposé dans [2]. Ce système définit est un système de coordonnées auquel les entités du réseau pourront se référer pour obtenir leur position. Il est léger et complètement décentralisé et repose uniquement sur l'exploitation de données locales.

3.5.1 Hypothèses

L'étude s'effectue sur les scénarios où la région surveillée ne permet pas d'intervention humaine. Pour des raisons de simplicité, l'analyse des coûts est effectuée soit :

- Sur une distribution de nœuds uniforme dans une zone de forme rectangulaire, ou
- En supposant les nœuds répartis régulièrement sur une grille. Le principe demeure le même sur d'autres distributions. Dans le cas d'une grille de N nœuds, chaque rangée et colonne contient \sqrt{N} nœuds.

Nœuds : Le système est composé d'un ensemble fini de N capteurs répartis de façon uniforme sur une zone géographique. Chaque nœud possède un identifiant unique i .

Communication : Chaque nœud peut communiquer avec tout autre nœud situé à portée radio, qui est modélisée par un disque de diamètre R . Le réseau est supposé non partitionné et les communications bidirectionnelles.

Connaissance initiale : Initialement, un nœud connaît seulement son identité, le fait qu'il soit le seul à posséder cette identité dans le réseau, et un paramètre d , définissant la dimension de l'espace de coordonnées.

3.5.2 Contexte

La majorité des approches existantes reposent sur des hypothèses spécifiques (de la disponibilité d'informations *GPS* à la connaissance de l'angle d'arrivée du message radio), et témoignent de la nécessité d'une approche permettant plus d'autonomie. Cela fut traité dans certaines solutions n'utilisant aucune ancre (nœud connaissant sa position). Il en résulte des systèmes de coordonnées virtuelles (par opposition aux systèmes de coordonnées géographiques).

3.5.3 Description

Ce système constitue une brique de base pour la structuration du réseau, et compose ainsi le cœur d'un système autonome. En plus il ne repose sur aucune ancre, ne suppose aucun nœud conscient de sa position, et n'utilise aucun système de mesure du signal. Les nœuds obtiennent des coordonnées d'une manière complètement décentralisée. Pour cela ils exploitent uniquement des informations locales ou obtenues par communications entre nœuds voisins pour :

- identifier les zones denses du système, et
- utiliser ces zones denses pour détecter la périphérie (ou bordure) du système.

Ces informations peuvent ensuite être exploitées pour implémenter différentes fonctionnalités.

Principe des coordonnées

Le positionnement est une information clé dans la construction et le maintien de systèmes autonomes. Un système de coordonnées procure à chaque nœud une position qui est à la fois localement et globalement cohérente. VINCOS considère un système à d dimensions. Ainsi, les coordonnées virtuelles d'un nœud i sont représentées sous la forme d'un tuple (x_1, \dots, x_d) , où x_j est la projection du nœud i sur le j ème axe de l'espace virtuel à d dimensions. L'unité de distance utilisée est le saut.

L'espace virtuel est défini comme suit :

La bordure de la zone géographique couverte par les nœuds est partitionnée en d segments. Ces segments de bordure peuvent avoir soit la même taille, soit des tailles différentes. Soit un axe j , $1 \leq j \leq d$, du système de coordonnées. La coordonnée x_j est alors la distance, en sauts, du plus court chemin du nœud i au segment j (i.e. au nœud le plus proche appartenant au segment). La figure suivante illustre ceci :

- L'espace de coordonnées y est de dimension $d = 3$.

- Les coordonnées virtuelles de trois nœuds y sont portées : les coordonnées $(2, 4, 2)$ signifient que le nœud est à distance 2 des bordures 1 et 3, et à distance 4 de la bordure 2. Ce système ne nécessite ainsi pas d'ancre ou de nœud particulier.

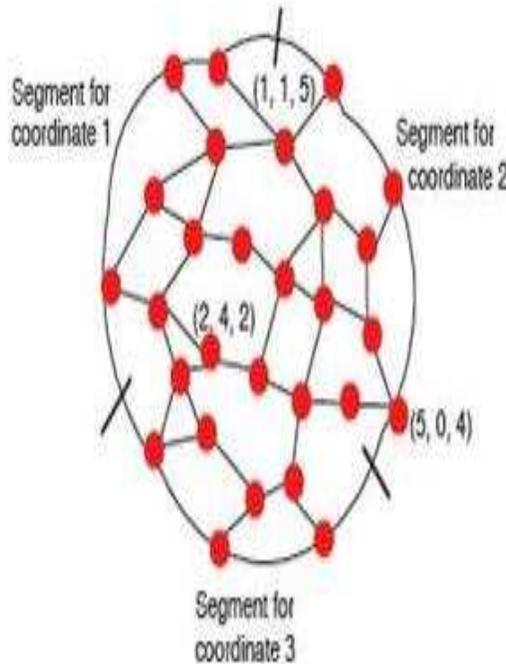


FIGURE 3.5 – Exemple de coordonnées virtuelles

Les coordonnées de *VINCOS* dépendent d'une définition précise des segments de bordure, i.e. les d segments qui composent la bordure du système. Afin de définir de tels segments, *VINCOS* repose sur la construction d'une ceinture de processus autour de la bordure (les nœuds de la bordure).

VINCOS : De l'anarchie aux coordonnées virtuelles

Afin de construire un système de coordonnées cohérent de façon décentralisée, *VINCOS* comporte quatre phases qui sont les suivantes :

1. Phase 1 : détection des initiateurs

Cette phase utilise les densités pour identifier un petit ensemble d'initiateurs. Ce sont des nœuds qui joueront un rôle important dans les phases suivantes :

- Découverte des voisins : Chaque nœud i émet un message contenant son identité. Ainsi chaque nœud i peut établir la liste des nœuds dans son voisinage : l'ensemble $\text{Neighbors } i$. Chaque nœud i émet ensuite son nombre de voisins, $|\text{neighbors}_i|$. Cela permet à chaque nœud d'apprendre le nombre de voisins qu'ont ses voisins.

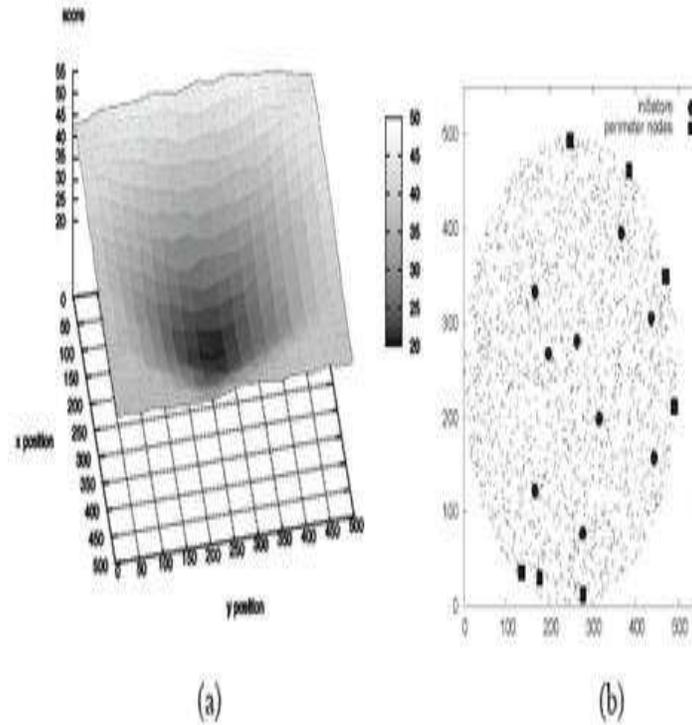


FIGURE 3.6 – (a) Les nœuds de bordure ont un score plus élevé. (b) Initiateurs et initiateurs de bordure pour un réseau de 2000-nœuds.

- Découverte des initiateurs : Un initiateur est défini comme un maximum local de densité. Ceci se traduit intuitivement par le nœud ayant plus de voisins que tous ses voisins. Formellement, un nœud i est initiateur s'il vérifie le prédicat suivant :

$$\forall j \in |neighbors_i| \quad |neighbors_j| < |neighbors_i| \quad (3.1)$$

Puisque la densité est supposée uniforme, les nœuds sont supposés avoir un nombre de voisins proportionnel à l'aire du système présent à portée radio. Ainsi les nœuds de bordure, qui ont la moitié de leur disque radio hors du système, sont supposés avoir moitié moins de voisins que les nœuds dans le système. Ainsi, le prédicat précédent permet de sélectionner des voisins qui ne sont pas sur la bordure, ce qui a son importance pour les phases suivantes.

- Phase 2 : Définition du score de bordure L'objectif est de procurer à chaque nœud un score de bordure. Pour cela, chaque nœud détermine sa distance à chaque initiateur.
 - Calcul de la distance aux initiateurs : Chaque initiateur i émet un message m contenant son identité et un compteur de sauts, initialisé à 0. Après avoir reçu le message m , et si m améliore sa distance à i , un nœud j incrémente le compteur de sauts de m et le réémet ; sinon il ne tient pas compte du message m . La distance d'un nœud j à un initiateur i est :

$$dist(i, j) = \min_{l \in neighbors_j} dist(i, l) + 1. \quad (3.2)$$

- Le score de bordure : La valeur, notée $score_j$, du score de bordure d'un nœud j est définie ainsi :

$$score_j = \sum_{i \in initiators} dist(i, j) \quad (3.3)$$

Il est intéressant d'observer qu'en plus de connaître son score, chaque nœud connaît aussi le score de ses voisins (puisque chaque nœud réémet les messages provenant des initiateurs). Ce

score peut être vu comme la distance à un « initiateur moyen ». La figure 3.6-(a) représente, en échelle de gris, le score de chaque nœud en fonction de sa position. On peut constater que le minimum (i.e. le plus proche de l'initiateur moyen) est situé au centre du système.

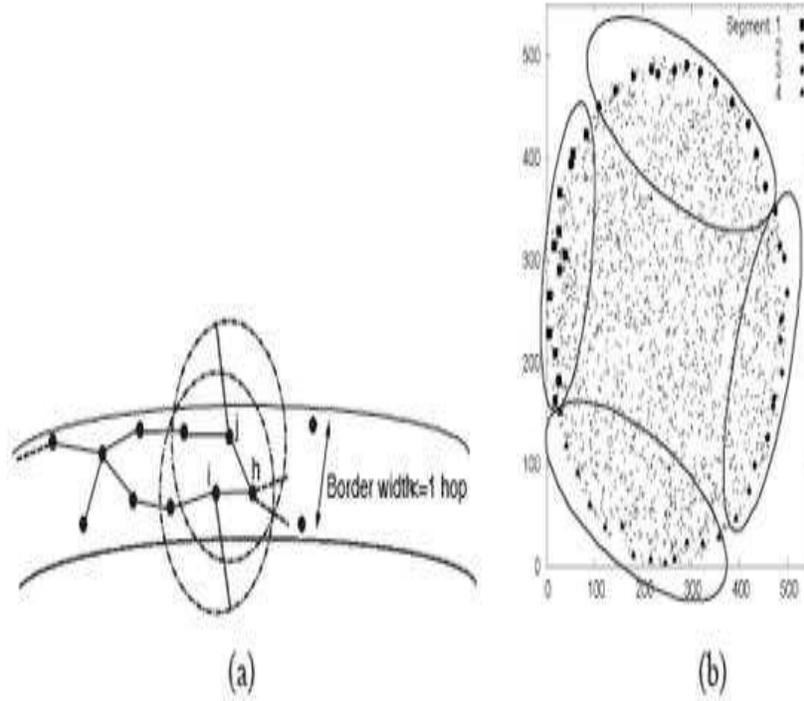


FIGURE 3.7 – (a) Les voisinages des nœuds i et j s'intersectent, ce qui permet au nœud h de ne pas croire qu'il a reçu des sondes provenant de côtés différents. (b) Les 4 segments définis pour un réseau de 2000 nœuds.

- Quelques nœuds de la bordure : Le deuxième objectif de la phase 2 est de découvrir quelques nœuds qui sont de façon certaine sur la bordure. Nous les appellerons initiateurs de bordure. Ces nœuds sont les nœuds responsables de l'initiation de la construction de la ceinture, lors de la phase 3. Un nœud j se découvre en bordure par rapport à un initiateur i s'il vérifie le prédicat suivant :

$$\forall l \in neighbors_j : dist(i, j) \geq (i, l). \quad (3.4)$$

Soit x un pourcentage d'initiateurs. Le protocole considère qu'un nœud est initiateur de bordure s'il est en bordure par rapport à $x\%$ d'initiateurs. Autrement dit, un nœud est initiateur de bordure s'il vérifie le prédicat pour $x\%$ initiateurs. Nous avons expérimentalement constaté que $x = 60\%$ permettait un bon compromis entre faux positifs (nœuds détectés à tort comme en bordure) et faux négatifs (nœuds non détectés comme en bordure). La figure 3.6-(b) illustre un exemple de sélection d'initiateurs et d'initiateurs de bordure.

3. Phase 3 : Construction de la ceinture

C'est la phase essentielle de *VINCOS*. L'objectif est d'établir une liste de nœuds qui appartiennent tous à la bordure. Cette liste est une structure mono-dimensionnelle. La détection d'une telle bordure est complexe s'il faut aussi un coût raisonnable. C'est la phase essentielle de *VINCOS*. L'objectif est d'établir une liste de nœuds qui appartiennent tous à la bordure. Cette liste est une structure mono-dimensionnelle. La détection d'une telle bordure est complexe s'il faut aussi un coût raisonnable.

- Lancement des sondes : Chaque initiateur de bordure identifié à la phase précédente envoie un message, appelé sonde, qui va «coller» à la bordure du système en s'appuyant sur les scores de bordure.
- Transmission de la sonde : Une sonde contient deux ensembles de nœuds : destinators et exclude. Un nœud i considère qu'il reçoit une sonde si l'ensemble destinators de la sonde contient i . Chaque nœud connaît son score de bordure, et le score de tous ses voisins. Lorsque le nœud i reçoit une sonde, il sélectionne k candidats à qui il va relayer la sonde (k est un paramètre de l'application). Ces k nœuds sont les nœuds ayant le score de bordure le plus élevé parmi l'ensemble candidates i tel que :

$$candidate_{s_i} = neighbor_{s_i} - (destinators(exclude)). \quad (3.5)$$

Ces k nœuds constituent l'ensemble destinators de la sonde que i va émettre. L'ensemble exclude de la sonde émise par i est l'ensemble destinators de la sonde que i a reçu. La sonde n'étant relayée que par les nœuds ayant un score de bordure localement maximal, elle va «coller» la bordure. Chaque sonde contient aussi un champ path, qui contient la liste des nœuds qui ont relayé la sonde : chaque nœud relayant la sonde y ajoute son identité. Enfin, chaque sonde contient aussi la liste des voisins du deuxième nœud la relayant. Cette liste est appelée stamp. Elle sert à identifier de manière unique chaque «coté» de la ceinture : deux sondes pourraient emprunter le même coté de la bordure avec des chemins différents (voir Figure 3.7-(a) mais l'intersection de leurs champs stamp serait alors non nulle.

- Définition des segments : Lorsqu'un nœud j reçoit deux sondes issues du même initiateur de bordure, il vérifie que ces deux sondes ont des chemins et des champs stamp différents. Si c'est le cas, l'union des champs path des deux sondes est une ceinture du système. Le nœud j , alors appelé définisseur de segment, va découper cette ceinture en d segments, et émettre un message de segment contenant la liste des nœuds composant chaque segment. Ce message de segment va alors emprunter le chemin inverse, relayé par les nœuds de la ceinture. Chaque nœud relayant un tel message apprend alors à quel segment il appartient (voir figure 3.7-(b)).
 - Combattre l'ambiguïté : Un mécanisme de propagation-extinction est utilisé sur les identités des initiateurs de bordures afin de ne garder qu'une segmentation pour tout les systèmes.
4. Phase 4 : Définition des coordonnées
- Obtention des coordonnées : Lorsqu'un nœud de la ceinture a appris à quel segment s auquel il appartient, il émet un message contenant ce numéro s et un compteur de sauts initialisé à 0. Ces messages sont relayés et utilisés par tous les nœuds pour découvrir à quelle distance ils sont de chaque segment de bordure. Chaque nœud i possède finalement un ensemble de coordonnées (x_1, \dots, x_d) dans l'espace de coordonnées d -dimensionnel.

3.6 Les apports de VINCOS :

1. Des coordonnées virtuelles à la structuration géométrique du réseau : Les coordonnées sont principalement utilisées pour faire du routage, et aussi elles constituent entre autres un outil puissant d'organisation du réseau. Afin d'organiser un réseau, il est intéressant de pouvoir assigner aux nœuds différents comportements selon leur position. Dans ce cas on peut utiliser les coordonnées assignées par VINCOS. L'objectif d'utiliser les coordonnées VINCOS est de faire émerger une structuration géométrique du réseau. Cette structuration pourra être utilisée pour des fins multiples, par exemple :
 - Le clustering,
 - L'agrégation de données,
 - La gestion des consommations d'énergie.

2. Définition de la fonction de structuration géométrique : C'est une fonction transformant des coordonnées en un numéro de partition. Formellement, soit K l'espace des coordonnées (avec $VINCOS, K = N^d$), et p le nombre de partitions différentes souhaitées, alors une fonction de structuration géométrique est une fonction f telle que :

$$f : k \rightarrow \{0, \dots, p\}$$

$$f(c_i) \rightarrow p_i. \tag{3.6}$$

Une telle définition permet à un nœud de calculer le numéro de partition de n'importe quel nœud dont il connaît les coordonnées : chaque nœud a une idée de la structuration du système.

3.7 Conclusion

Nombreuses méthodes ont été proposées dans la littérature pour estimer la localisation des nœuds fixes dans les réseaux de capteurs. Cependant, les méthodes de localisation pour les nœuds mobiles dans les réseaux de capteurs sont devenus un autre défi pour ces réseaux. Les classifications usuelles des méthodes de localisation de nœuds fixes pour les réseaux de capteurs peuvent être décrites comme suit :

1. Techniques centralisées basées sur la puissance du signal (GPS ou autres),
2. Techniques hybrides basées sur la localisation de quelques nœuds dans le réseau (Anchor-based).
3. Techniques complètement distribuées collaboratives (que des interactions locales entre les nœuds pour obtenir la position)

Dans ce travail, on ne considère pas l'existence des nœuds mobiles mais une contrainte forte sur le passage à l'échelle, la gestion de l'énergie et un contexte embarqué conduisant à une exigence de faible complexité algorithmique. Pour cela, nous avons banni l'utilisation d'informations de localisation (GPS ou assimilé) et nous n'avons aucune exigence au regard de la synchronisation des nœuds.

Enfin, nous avons étudié en détail le système de coordonnées virtuelles. On a pu conclure que $VINCOS$ est un système de coordonnées virtuelles auquel le réseau est léger, complètement décentralisé, économique en énergie et repose uniquement sur l'exploitation de données locales. L'algorithme $VINCOS$, assigne des coordonnées virtuelles aux nœuds, et l'exploitation de ces coordonnées permet de structurer un réseau. $VINCOS$ repose sur des hypothèses très faibles (les nœuds ont des identités distinctes et comparables), et ne nécessite aucune ancre ou système de mesure de signal. Le mot clé de $VINCOS$ est issu par l'algorithme de détection de bordures. Une fois la bordure est détectée, la segmentation du réseau se fait par un sondage entre les nœuds de la bordure et chaque nœud prend sa position dans le réseau selon les plus courts chemins entre lui et les nœuds de bordure de chaque segment. Cependant, la consommation d'énergie dans cet algorithme est non négligeable, un nombre indéterminé de messages de propagation envoyé par les initiateurs de bordure est requis pour découvrir toute la bordure. Pour cette raison, nous allons proposer dans le chapitre suivant un protocole d'auto-structuration en prenant la consommation d'énergie comme un facteur primordial.

Chapitre 4

Un protocole simple et efficace pour l'auto-structuration géométrique dans les RCSFs

4.1 Introduction

La complexité des algorithmes d'auto-structuration dépend fortement de la quantité de connaissance initiale de chaque capteur (chaque nœud ne connaît que son identité), ce qui rend le développement des algorithmes d'auto-structuration un peu complexe. D'autre part, si un réseau repose sur la connaissance externe (précision du GPS, régularité de la zone de déploiement...etc.), la structure devient moins robuste et irréaliste dans certains environs. Donc la gestion centralisée de ces réseaux pose un risque, puisque la disparition des nœuds critiques souvent rencontrée dans ces réseaux est due aux limitations énergétiques de leurs batteries et l'impossibilité de les renouveler, alors la mise en défaut des nœuds critiques empêche la réussite de la structuration.

Nous abordons l'étude de développement d'un algorithme efficace répondant à ces contraintes, en se basant uniquement sur des interactions locales et chaque nœuds ne connaît initialement que son identité. L'émergence d'une structure des capteurs sans fils d'une manière autonome et avec la prise en compte de la contrainte énergétique est importante pour de nombreuses opérations telles l'élection de leader, l'équilibrage de charge, l'économie d'énergie et le routage.

4.2 Contexte et Motivation

Le principal objectif de notre contribution est de proposer un algorithme pour auto-structurer les capteurs dans les réseaux RCSFs. Pour que cet algorithme soit efficace, il fallait qu'il prenne en considération les spécificités des réseaux de capteurs et qu'il soit réalisable dans des conditions qui se rapprochent de la réalité. En effet, quelques facteurs ont été pris en compte à savoir :

- La distribution aléatoire des capteurs dans la zone d'intérêt,
- Nous considérons aussi que les capteurs sont stables durant une période raisonnable pendant l'exécution du processus de structuration.

En plus, certains paramètres sont considérés fondamentaux pour mettre en œuvre cet algorithme. Ces paramètres concernent :

- La position des capteurs : les capteurs n'ont aucune connaissance de leurs positions physique dans le réseau.
- Connaissance du voisinage : chaque capteur ne puisse communiquer qu'avec ses voisins de 1-saut

pour que la diffusion ou le routage soit effectué normalement après la structuration du réseau. Les échanges des messages sont utilisés pour découvrir les voisinages.

- Connectivité : est un facteur déterminant dans les réseaux ad hoc et de capteurs. Elle permet d'assurer que tout nœud destinataire est joignable par tout nœud source du réseau. En particulier, dans les réseaux de capteurs, il devrait exister au moins un chemin entre tout capteur et la station de base pour garantir l'acheminement de l'information en tout moment et que tous les nœuds soient joignables à partir de la station de base.
- Couverture de zone d'intérêt : est un élément important pour les réseaux de capteurs. Tout algorithme conçu pour les réseaux de capteurs devrait assurer que l'occurrence d'un tel événement dans la zone de déploiement des capteurs pourrait être détectée par au moins un capteur (sensing fidelity).

4.3 Modèle du système

Considérons un réseau de capteur sans fils où tous les nœuds coopèrent dans le but d'émerger une structure finale. Un tel réseau sans fil peut être représenté par un graphe connexe $G = (V, E, P)$; où V est l'ensemble des nœuds (capteurs), $E \subseteq V^2$ l'ensemble des arcs reflétant les communications directes possibles entre les nœuds : la paire orientée (u, v) appartient à E si et seulement si le nœud u peut envoyer directement un message au nœud v , et nous disons que v est voisin de u ou également u couvre v . P est une fonction associant à chaque sommet $u \in V$ un poids $W(u) \in R$ représentant sa capacité d'être cluster-head. Les couples appartenant à E dépendent de la position des nœuds et de leur portée de transmission.

Nous supposons que tous les nœuds ont la même portée de transmission R_{Tx} , et tous les liens dans le réseau sont bidirectionnels, c'est-à-dire que si u est un voisin de v alors v est un voisin de u . Pour chaque nœud u , nous attribuons une valeur unique qui le caractérise, appelée identifiant et notée $NodeId(u)$. La distance $d(u, v)$ entre deux nœuds u et v est exprimée en nombre de sauts i.e. le nombre minimal de sauts qu'un message doit parcourir pour se rendre de u à v alors que $dist(u, v)$ représente la distance euclidienne entre u et v .

Nous avons utilisé le modèle du disque unitaire (Unit Disc Graph)[6], avec R_{Tx} comme rayon de transmission. Ce modèle est très répandu pour modéliser les communications entre les nœuds dans les protocoles de diffusion conçus pour les réseaux ad hoc et de capteurs. Dans ce modèle, il est supposé que deux nœuds peuvent communiquer entre eux si la distance euclidienne $dist(u, v)$ qui les sépare n'est pas supérieure à une portée de transmission donnée R_{Tx} , et les messages sont toujours reçus sans aucune erreur. D'où, l'ensemble E peut être défini comme suit :

$$E = \{(u, v) \in V^2 \mid dist(u, v) \leq R_{Tx}\} \quad (4.1)$$

L'ensemble des voisins $N_1(u)$ d'un nœud u est défini par l'équation 4.1 et son degré $\delta_1(u)$ (ou 1-degré) représente le nombre de ses voisins c'est-à-dire le cardinal de l'ensemble $N_1(u)$.

$$N_1(u) = \{v \in V \mid v \neq u \wedge (u, v) \in E\} \quad (4.2)$$

$$\delta_1(u) = |N_1(u)| \quad (4.3)$$

et l'ensemble étendu des voisins $N_1[u]$ est représenté par :

$$N_1[u] = N_1(u) \cup \{u\} \quad (4.4)$$

L'ensemble des voisins à deux sauts $N_2(u)$ d'un nœud u représente l'ensemble des nœuds qui sont les voisins des voisins du nœud u et qui ne sont pas les voisins de u . Il est défini comme suit :

$$N_2(u) = \{w \in V \mid (v, w) \in E : w \neq u \wedge w \notin N_1(u) \wedge v \in N_1(u)\} \quad (4.5)$$

La réunion des ensembles $N_1(u)$ et $N_2(u)$ représente l'ensemble de tous les nœuds présents à une distance inférieure ou égale à deux sauts de u . Elle est notée soit $N_{12}(u, v)$ ou $N_2(u)$ et définie comme suit :

$$N_{12}(u) = N_1(u) \cup N_2(u) = \{v \in V | v \neq u \wedge d(u, v) \leq 2\} \quad (4.6)$$

D'une manière générale, l'ensemble des k-voisins $N^k(u)$ (ou voisins à k sauts) d'un nœud u contient tous les nœuds se retrouvant à une distance inférieure ou égale à k sauts de u . Il est défini par l'équation suivante :

$$N^k(u) = \{v \in V | v \neq u \wedge d(u, v) \leq k\} \quad (4.7)$$

et l'ensemble étendu des k-voisins $N^k(u)$ du nœud u est défini par :

$$N_k[u] = N^k(u) \cup \{u\} \quad (4.8)$$

Le k-degré d'un nœud u noté $\delta_k(u)$ est le nombre de ses k-voisins (le cardinal de $N^k(u)$) :

$$\delta_k(u) = |N^k(u)| \quad (4.9)$$

La k-densité d'un nœud u représente le rapport entre le nombre de liens dans son k-voisinage (liens entre u et ses voisins et liens entre deux k-voisins de u) et le k-degré de u ; formellement, elle est représentée par l'équation suivante :

$$\text{k-densité}(u) = \frac{|(v, w) \in E : v, w \in N^k[u]|}{\delta_k(u)} \quad (4.10)$$

Pour illustrer la métrique k-densité, nous proposons l'exemple représenté par la figure 4.1 . Le tableau 4.1 illustre le calcul de la 1-densité des nœuds composant le réseau présenté dans la figure 4.1.

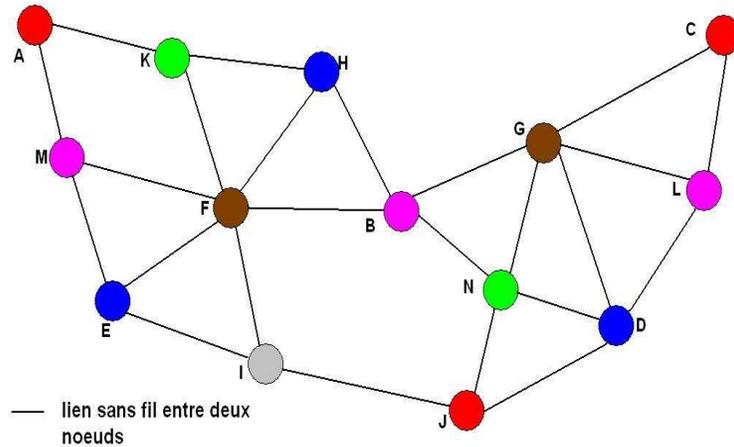


FIGURE 4.1 – Représentation graphique(non orienté) d'un RCSF

Noeud	A	B	C	D	E	F	G	H	K	L	M	N
1-degré	2	4	2	4	3	6	5	3	3	3	3	4
1-densité	2.5	3.0	2.5	1.75	2.66	1.66	2.66	3.0	2.33	2.0	2.33	2.25

TABLE 4.1 – Calcul de 1-degré et 1-densité de la figure 4.1

Dans ce travail, nous supposons que chaque capteur possède une antenne omnidirectionnelle lui permettant par une transmission simple de couvrir tous les capteurs se trouvant dans son voisinage, et que les capteurs sont déployés dans un espace bidimensionnel.

4.3.1 Le principe de base de la solution

Après une étude approfondie de VINCOS dans le chapitre précédent, nous avons conclu ses avantages comme :

- Fonctionnement complètement décentralisé,
- Des interaction locales uniquement,
- Des cordonnées virtuelles à chaque nœud de réseau.

Mais on a aussi découvert que VINCOS a des inconvénients majeurs surtout en facteur de consommation énergétique. Un nombre indéterminé d'itération requiert pour découvrir toute la bordure dans la phase de détection de bordure(voir le section 3.5), ça demande une forte consommation d'énergie. Pour cette raison, nous avons posé la question *"pour quoi la détection de la bordure et pas du centre"*. Cette question ouvre un autre espace dans la localisation dans les réseaux mobiles, tel que : comment trouver le centre, construire le réseau et mettre les cordonnées virtuelles.

L'algorithme que nous allons proposer doit prendre en compte les inconvénients de VINCOS et répondre aux exigences suivantes, et en plus simple, efficace et adapté aux topologies dense(déploiement aléatoire, consommation d'énergie. . .), La figure illustré ci-dessous 4.2 montre ce principe.

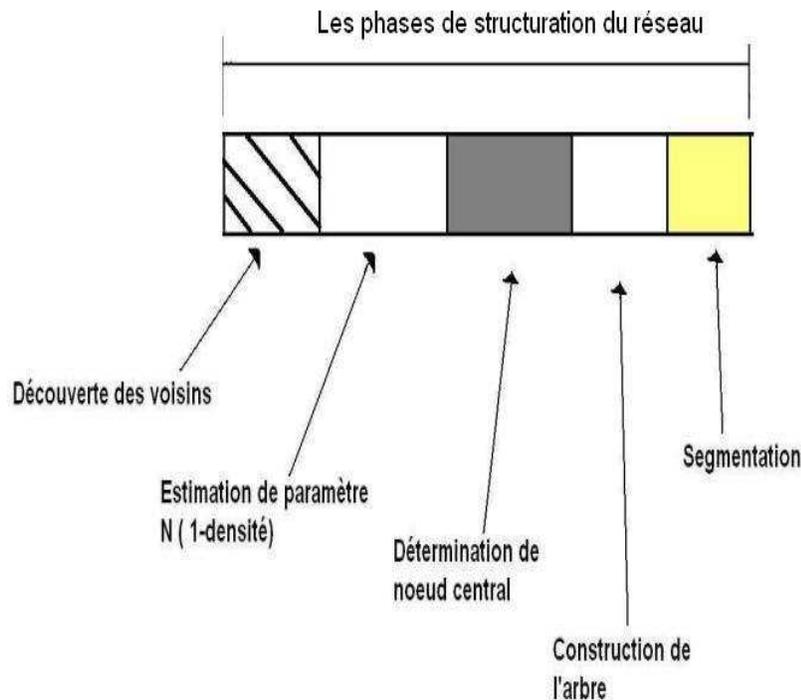


FIGURE 4.2 – Les phases de formation du réseau

1. La distribution initiale : Nous allons étudier dans notre proposition deux distributions de capteurs, l'un uniforme et l'autre aléatoire. On suppose que la distribution dans un espace de deux dimensions (X, Y) , pas des obstacle physique entre les capteurs, les capteurs sont identiques(capacité de mémoire, d'énergie, porté...etc.), chaque capteur a une porté de radio de longueur R sous forme de disque, une communication bidirectionnelle et le réseau non partitionné.
2. Optimisation du nombre de nœuds participants dans la structure :

- Pour assurer une longue durée de surveillance du réseau, il fallait assurer la couverture maximale de région avec un nombre minimal de nœuds capteurs. Puisque les capteurs n'ont pas des connaissances de leurs positions dans le réseau, nous allons alors minimiser le nombre de capteurs avec une technique simple et efficace.
 - Chaque capteur i envoie un message M (message de couverture de leur voisinage).
 - Si un nœud capteur j reçoit le message M , répond par un message qui contient son identité j .
 - Après de la réception des messages de ses voisins, le nœud capteur i construit une liste qui contient tous les identités des capteurs en voisin.
 - Le nœud capteur i envoie un message M_{id_i} à tous ses voisins qui contient la liste de ses voisins.
 - Pour chaque capteur j reçoit le message M_{id_i} , compare M_{id_i} avec son liste M_{id_j}
 - (a) Si M_{id_j} inclus dans M_{id_i} le capteur j va mettre en veille
 - (b) Sinon, si $M_{id_j} = M_{id_i}$, alors l'un de ces capteurs(i ou j) va mettre en veille
 - (c) Si non, alors le capteur i va mettre en veille

Avec cette technique, nous allons assurer une couverture maximale avec un nombre minimale de capteurs. Et les nœuds capteurs qu'ils sont mis en veilles vont mettre en actif après la disparition d'un capteur(par exemple l'épuisement de leur batterie), pour assurer une longue durée du réseau.

Cette technique offre un autre avantage dans la phase suivante. C'est difficile de trouver le nœud central avec une distribution aléatoire, où le k-densité de réseau ne donne pas la localisation exacte du centre de notre réseau. L'avantage de cette technique permet d'équilibrer la charge des nœuds participants dans le réseau.

3. Déterminer le nœud central : Le défi de cette étape est de déterminer le nœud central tel que les nœuds capteurs n'ont pas aucune information de leurs positions. Alors il fallait de chercher des solutions converge vers le nœud central réel et simple pour économiser la consommation d'énergie. Nous allons proposer deux techniques l'un très simple et pas précise et l'autre nécessite beaucoup d'envoi de messages mais converge vers le central réel :
 - (a) Pour déterminer le central, chaque nœud capteur du réseau fait le calcul de 1-densité(voir la figure 4.1 et le tableau 4.1). Après le calcul les capteur partagent les valeurs de 1-densités et le nœuds qu'il a la grande valeur est le nœud central
 - (b) La deuxième technique consiste deux étape :
 - i. Estimation de paramètre N
 - ii. Détermination du nœud central
4. La formation de l'arbre : Les contraintes imposées par les capteurs rendent la conception d'une technique efficace pour la diffusion et le prolongement de la durée de vie dans les réseaux de capteurs comme un vrai défi. Pour traiter ce défi, nous allons proposer une technique (simple, très efficace et adaptée aux RCSFs) d'auto-structuration basée sur l'approche d'arborescence pour optimiser la consommation de l'énergie dans ces réseaux. Le principe de base de cette technique est, le nœud central C_{node} envoie un message de propagation contient son identité, pour chaque nœud reçoit le message va considérer la source de message comme père et envoie encore un message qui contient son identité, si un nœud reçoit un message va ignorer tous les autres messages reçus par la suite. Les figures 4.3 4.4 présentent des graphes simplifiés de propagation des messages.
5. Segmentation de l'arbre : Nous allons après la construction de l'arbre segmenter le réseau en segment. Cette technique consiste à regrouper les nœuds proches virtuellement en segments.

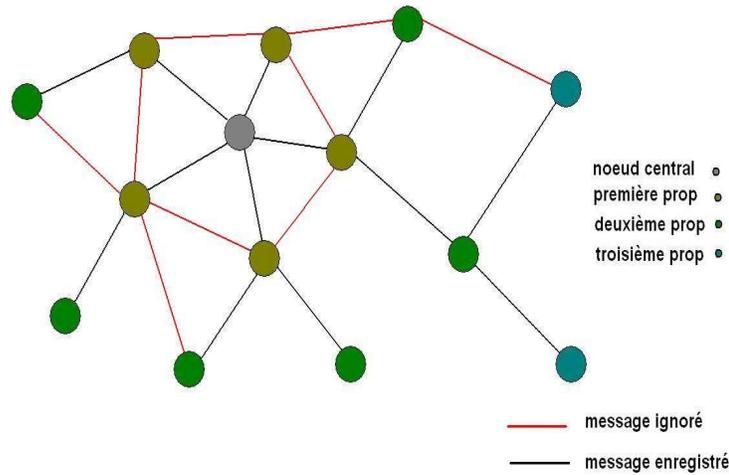


FIGURE 4.3 – Représentation graphique de propagation des messages

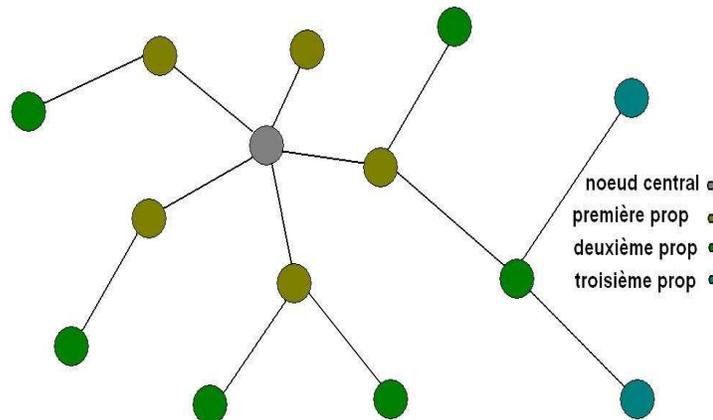


FIGURE 4.4 – Représentation graphique de propagation que les messages pertinents

4.4 Hypothèses

Ce mémoire traite principalement la problématique de structuration du réseau en optimisant la consommation d'énergie et le prolongement de la durée de vie dans les réseaux de capteurs sans fil. L'objectif principal est de proposer un algorithme efficace de structuration des capteurs afin de maximiser la durée de vie du réseau, tout en prenant les hypothèses suivantes :

- Le déploiement aléatoire des nœuds : le déploiement des nœuds dans la zone est complètement aléatoire de telle sorte que n'y a aucune information connue auparavant sur le réseau et les voisins d'un nœud ne sont pas connus au début. Mais nous discutons aussi un déploiement uniforme pour valider les résultats du déploiement aléatoire,

- La connaissance initiale : tous les nœuds capteurs possèdent des identifiants uniques avant le déploiement,
- Liaisons bidirectionnelles : chaque capteur peut envoyer ou recevoir des messages directement avec ses voisins de 1-saut, et nous supposons que la communication entre les nœuds est fiable,
- L'homogénéité : tous les capteurs sont physiquement homogènes (ont la même capacité en termes de mémoire, puissance de calcul, communication et d'énergie) et que l'identité qui diffère un nœud d'un autre,
- Les capteurs opèrent d'une manière asynchrone et sans contrôle centralisé.

4.5 Un algorithme implémentant l'auto-structuration

Dans ce travail, nous avons visé la réduction de la consommation d'énergie et le prolongement de la durée de vie des réseaux de capteurs. Pour atteindre ces objectifs, nous avons proposé une technique de segmentation, qui implique la 1-densité ou le paramètre N dans le calcul du nœud central afin de garantir la bonne arborescence et segmentation.

L'idée de base est de construire une structure virtuelle adaptée aux RCSFs de manière complètement décentralisée, basée uniquement sur les données locales et sans aucune information de localisation géographique ni information globale concernant le réseau. La topologie ainsi générée est un arbre construit à partir d'un nœud appelé nœud central. C'est une différence par rapport autres solutions basées sur la construction d'un arbre initiée par le nœud puits en inondant le réseau avec un message d'initialisation et la maintenance requiert l'utilisation de paquets hello périodiques ce qui rend ces solutions non efficaces en énergie.

Avant d'aborder les détails techniques de l'algorithme, nous présentons quelques définitions et notations qui s'avèrent nécessaires à sa compréhension.

4.5.1 Notation

Le processus de formation de l'arbre est exécuté en cinq phases : Découverte des voisins, Estimation du paramètre N (ou calcul de 1-densité), Détermination du nœud central, construction de l'arbre, et segmentation. Pour bien comprendre ces phases nous utilisons les notations de la table 4.2.

4.5.2 Les phases de formation de l'arbre

Pour bien comprendre l'algorithme de construction de l'arbre, nous le coupons en cinq phases.

1. Phase 1 : Découverte des voisins

Après le déploiement, chaque nœud capteur S_i découvre son voisinage en envoyant un message contenant son identité, ce même nœud reçoit des messages provenant d'autres nœuds qui seront rajoutés à son ensemble de voisins (v_i). De cette façon chaque nœud S_i connaît l'ensemble de ses voisins v_i .

Notre proposition se base sur l'optimisation du nombre de nœuds participants en excluant certains, dont le voisinage est équivalent ou inclus dans un autre. Cette étape a un intérêt surtout pour les RCSFs de densité importante où c'est possible de trouver des nœuds dont les voisinages sont équivalents ou inclus, donc ça serait intéressant d'en sélectionner que les nœuds critiques c'est-à-dire ceux dont l'exclusion entraîne une perte d'information. Pour cela chaque un des capteurs S_i diffuse l'ensemble v_i à tous ses voisins et attend jusqu'à réception des ensembles V de

Variable	Description
S_i	L'identificateur de l' $i^{\text{ème}}$ noeud capteur
V_i	L'ensemble des voisins du noeud S_i
$m(S_i)$	Message de découverte des voisins
$q(V_i)$	Message d'acquiescement envoyé par S_i à tous ses voisins V_i
$Nine_i$	L'ensemble des voisins non inclus non égal de S_i
$Include_i$	L'ensemble des voisins dont le voisinage est inclus dans V_i
$Equal_i$	L'ensemble des voisins dont le voisinage est égal à V_i
N	Un paramètre de convergence utilisé pour déterminer le noeud central
X_i	Variable utilisée pour obtenir le paramètre de convergence N
$central$	Ensemble des derniers noeuds convergeons à zéros
$Noeud\ central$	Le noeud central
$(explore, Z)$	Message d'exploration du réseau pour assurer la terminaison
(acq, X)	Acquiescement
$Nine_{central}$	L'ensemble des voisins non inclus non égal du noeud central
$reçu$	Booléen utilisé pour vérifier la réception du message d'exploration
$terminer$	Booléen utiliser pour vérifier la terminaison
$Pere_i$	Prédécesseur de S_i
$Fils_i$	Ensemble des descendants
$Bordure$	Ensemble des noeuds situés sur la bordure (feuilles)
Z	ensemble des noeuds ayant déjà reçus le message d'exploration
Y	ensemble des noeuds n'ayant pas reçus le message d'exploration
$Nbacq_i$	Nombre d'acquiescement attendus par le noeud S_i
NB	Nombre de noeuds de la bordure
$Seg_{message}$	Message annonce la segmentation du réseau
Seg_i	Le numéro du segment d'un noeud S_i

TABLE 4.2 – Les notations utilisées dans la proposition

tous ses autres voisins pour commencer à comparer ces ensembles reçus avec son propre ensemble. S'il trouve que certains de ses voisins ont un voisinage inclus dans le sien lui seul participera dans les prochaines phases et les autres seront exclus. Un autre cas est que S_i trouve d'autres dont le voisinage est le même que le sien alors le nœud dont l'identité est la plus petite participera et les autres seront exclus. Pour le reste des nœuds c'est-à-dire ceux dont le voisinage n'est ni inclus ni équivalent à celui de S_i ils sont considérés comme l'ensemble des nœuds $Nine_i$ (not include not equal) de S_i .

2. **Phase 2** : Pour déterminer le nœud central nous proposons deux techniques

(a) Estimation du paramètre N

Il s'agit ici d'estimer un paramètre N nécessaire pour la prochaine étape qui est la détermination du nœud central. Ceci est réalisé par échange de messages en deux étapes entre tous les capteurs et leurs voisins non inclus non équivalents $Nine$ respectifs,

– **L'étape A**

Les messages échangés dans cette étape contiennent les cardinalités des ensembles $Nine$. à la réception des messages attendus de tous ses voisins $Nine_i$, un nœud S_i additionne tous les nombres reçus avec son propre $|Nine_i|$ puis émet le résultat obtenu à tous ses voisins $Nine_i$.

– **L'étape B**

Dès qu'un nœud s_i reçoit la somme depuis un voisin il la compare à la sienne, dans le cas où la valeur reçue est plus grande il met à jour sa somme et réémet. Sinon il attend jusqu'à avoir reçue de tous ses voisins $Nine_i$ s'il n'y a aucune amélioration de la valeur il s'arrête. à la fin de cette étape tous les nœuds auront la même valeur qui est la plus grande somme obtenue et qui représente le paramètre N .

(b) Calcul de la densité

Dans les réseaux de capteurs déployés dans des zones hostiles et inaccessibles, nous favorisons le paramètre énergie sur les autres pour assurer une longue durée de vie au réseau. Pour cela nous proposons une technique simple, facile et efficace. Chaque nœud capteur du réseau fait le calcul de 1-densité (voir la figure 4.1 et le tableau 4.1).

3. **phase 3** : détermination du nœud central

Cette étape est importante car c'est à partir du nœud central on détermine l'arborescence et les segments de l'arbre.

– détermination du nœud central avec le paramètre N :

Pour se faire, chaque nœud S_i divise la somme obtenue en étape A de la phase 2 par le paramètre N , le résultat est émis à tous ses voisins $Nine_i$. A la réception, un capteur ajoute la valeur reçue à sa valeur locale puis divise par N ; le résultat est encore émis. Cette opération continue jusqu'à obtention d'un résultat très proche de zéro et le dernier à converger vers zéro constitue le nœud central.

Remarque : c'est possible dans certains cas de trouver plus qu'un nœud central, dans ce cas on choisit un seul parmi eux; par exemple : celui du plus petit identité.

– détermination du nœud central avec 1-densité

Après le calcul de 1-densité, les capteurs partagent les valeurs de 1-densités et le nœud qu'il a la grande valeur est le nœud central.

Remarque : c'est possible dans plusieurs cas de trouver plus qu'un nœud central, dans ce cas on choisit un seul parmi eux; par exemple : celui du plus petit identité.

4. **Phase 4** : Construction de l'arbre

Pour segmenter le réseau, nous construisons un arbre dont la racine est le résultat de la phase 3 et qui est le nœud central. Ce nœud est considéré comme la racine de l'arbre il commence à diffuser un message d'exploration à ces voisins *Nine* central. Ceci établit l'ensemble fils central (les fils du nœud central). Pour chaque nœud S_i , il prend en compte uniquement le premier message reçu et renvoi un acquittement positif à l'émetteur et des acquittements négatifs pour tous les autres. Ceci établit le *Père_i* (le père du nœud S_i). La réponse positive est envoyée au niveau supérieur que S_i , tous les acquittements attendus sont arrivés contrairement à la réponse négative qui est envoyée immédiatement. A la fin les feuilles de l'arbre c'est-à-dire les nœuds qui ne possèdent pas de fils sont les nœuds de la bordure.

5. Phase 5 : Segmentation

Nous allons après la construction de l'arbre segmenter le réseau, on prend en toujours considération le facteur d'énergie. L'exécution de cette phase procède de la façon suivante : les nœuds feuilles du réseau envoient un message de segmentation (contient son identité et le message de segmentation *Seg_{messg}*) à son père, chaque nœud reçoit le message rajoute son identité et le renvoie à son père jusqu'à ce que le message atteigne le nœud central (c'est l'inverse de la propagation). Le nœud central informe ses voisins que chacun est considéré avec ses fils comme un segment, sauf que les nœuds qu'ils n'ont pas des fils (les nœuds qu'ils ont la même valeur de convergence avec le nœud central), ils vont remplacer le central après sa disparition (épuisement de sa batterie, défaillance...etc.).

4.5.3 Présentation de l'algorithme

Algorithme 1 : Découverte des voisins

Construction de l'ensemble V_i de S_i :

```

1) for each nodei do
2)  $V_i \leftarrow \{S_i\}$ ;
3) send  $m(S_i)$ ;
4) while (receive  $m(S_j)$  from any Sensor  $S_j$ ) do  $V_i = V_i \cup \{S_j\}$ ;
5) send  $q(V_i)$  to all Sensor  $S_j \in V_i$ ;
6) wait until receive  $q(V_j)$  from all Sensor  $S_j \in V_i$ ;
7) Compute ( $Nine_i$ );
8) end for.

```

Fonction de construction de l'ensemble $Nine_i$:

```

9) Function Compute( $Nine_i$ )
10) Init :  $Include_i \leftarrow \{\}$ ,  $equal_i \leftarrow \{\}$ ,  $Nine_i \leftarrow \{\}$ ;
11) For each Sensor  $S_j \in V_i$  do
12)   If( $V_j \subset V_i$ )then  $Include_i = Include_i \cup \{S_j\}$ ;
13)   ElseIf( $V_j == V_i$ )then  $equal_i = equal_i \cup \{S_j\}$ ;
14)   Else  $Nine_i = Nine_i \cup \{S_j\}$ ;
15)   End If
16) End for;
17) If(( $equal_i \neq \{\}$ ) and ( $\exists k \mid \forall S_j \in equal_i$  and  $k < j$ ))Then
   Send(Stop) to all Sensor  $S_j \in equal_i - S_k \cup S_i$ ;
18) End If;
19) If( $Include_i \neq \{\}$ )then Send(stop) to all  $S_j \in Include_i$ ;
20) End If;
21) If receive(stop) from any sensor  $S_j \in V_i$  then No action();
22) End If;

```

Algorithme 2 : Estimation du paramètre N

Construction de l'ensemble des voisins de S_i :

————— Phase A —————

```

1) init :  $X_i \leftarrow 0$ ,  $N \leftarrow 0$ ;
2) For each Sensor  $S_i$  do
3)   Send ( $|Nine_i|$ ) to all Sensors  $S_j \in Nine_i$ ;
4)   Wait until receive ( $|Nine_j|$ ) from all Sensors  $S_j \in Nine_i$ ;
5)    $X_i = \sum |Nine_j| \setminus S_j \in Nine_i$ ;
6)   Send( $X_i$ ) to all Sensors  $S_j \in Nine_i$ ;
7)    $N = X_i$ ;

```

————— Phase B —————

```

8) When receive( $X_j$ ) from any Sensor  $S_j \in Nine_i$  do
9)   If( $X_j > X_i$ ) do  $N = X_j$ ;
10)  Send( $X_j$ );
11)  End If;
12) End for;

```

Algorithme 3 : Détermination du nœud central

```

1) init :  $Central \leftarrow \{\}$ ,  $Noeud\ central \leftarrow \{\}$ ;
2) For each Sensor  $S_i$  do
3) While( $X_i \geq 10^{-12}$ )do
4)    $X_i = X_i \setminus N$ ;
5)   Send( $X_i$ ) to all Sensors  $S_j \in Nine_i$ ;
6)   When receive( $X_j$ ) from any Sensor  $S_j \in Nine_i$  do
7)      $X_i = (X_i + X_j) \setminus N$ ;
8)     Send( $X_i$ ) to all Sensors  $S_j \in Nine_i$ ;
9)   End while;
10)  Send(0) to all Sensor  $S_j \in Nine_i$ ;
11)  If(For all Sensors  $S_j \in Nine_i$  ( $X_j < X_i$ )) then  $Central = Central \cup X_i$ ;
12)  End If;
13) End for;
14) If( $|Central| \neq 1$ ) then
15)  If( $\exists k | \forall S_i \in central_k \leq i$ ) then  $Noeud\ central \leftarrow S_k$ ;
16)  Else  $Noeud\ central \leftarrow Central$ ;
17)  End If ;
18) End If ;

```

Algorithme 4 : Construction de l'arbre

```

1) init :  $Reçu_i \leftarrow false$ ,  $Terminer \leftarrow false$ ,  $Père_i \leftarrow NULL$ ,  $Fils_i \leftarrow \{\}$ ;
    $Bordure \leftarrow \{\}$ ,  $Z \leftarrow \{Noeud\ central\}$ ,  $y \leftarrow \{\}$ ,  $NB \leftarrow 0$ ,  $Nbacq_i \leftarrow 0$ ;
   ————— Pour le nœud central, racine de l'arbre —————
2) If( $S_i == Noeud\ central$ ) then
3)  Send(explore,  $Z$ ) to all  $S_i \in Nine_{Ncentral}$ ;
4) End If;
5) For each  $S_i$  do
6)  When receive (explore,  $Z$ ) from any Sensor  $S_j \in Nine_i$  do
7)  If  $Reçu_i$  then Send(acq, no) to  $S_j$ ;
8)  Else
9)     $Reçu_i = true$ ;
10)    $Père_i = S_j$ ;
11)    $y = Nine_i - Z$ ;
12)   Case  $y == \{\}$  Send(acq, yes) to  $S_j$ ;
13)      $y \neq \{\}$ 
14)      $Nbacq_i = card(y)$ ;
15)      $Fils_i = y$ ;
16)     Send(explore,  $z \in y$ ) to all Sensors  $S_k \in y$ ;
17)   End Case;
18) End If;

```

```

19) When receive(acq, x) from any Sensor  $S_k \in \text{Nine}_i$  do
20) If  $x = \text{no}$  then  $\text{Fils}_i = \text{Fils}_i - \{S_k\}$ ;
21) End If;
22)  $\text{Nbacq}_i - -$ ;
23) If  $\text{Nbacq}_i == 0$  then
24)   Case  $S_i \neq \text{Noeud central}$  Send(acq, yes) to  $\text{Père}_i$ ;
25)    $S_i = \text{Noeud central}$ ;  $\text{Terminer} = \text{true}$ ;
26)   End Case;
27) End If;
28) If ( $\text{Terminer}$  and  $\text{Fils}_i == \{\}$ ) then  $\text{Bordure} = \text{Bordure} \cup \{S_j\}$ ;
29) End If;
30) End For

```

Algorithm 4 : Segmentation de l'arbre

```

1) For each Sensor  $S_i \in \text{Bordure}$  do Send( $S_i$ ) to Noeud central;
   _____ Pour le noeud central _____
2) If ( $S_i == \text{Noeud central}$ ) then
3)  $\text{NB} = 0$ ;
4) While (receive( $S_j$ , Segmessage) from Sensor  $S_j$ ) do
5)  $\text{NB} = \text{NB} + \text{Feuilles}_{S_j}$ ;
6)  $\text{Seg}_j = j$ ;
7) send( $S_j$ , Segj)
8) End while;
9) If ( $S_i == \text{Nine}_{\text{central}}$ ) then
10) If ( $\text{NB} > 1$ ) then  $S_j$  is a segment

```

4.6 Simulation

L'objectif principal de ce mémoire est de contribuer au développement de protocoles (sur la couche MAC) permettant d'éviter les sources inutiles de consommation d'énergie tout en tenant compte des contraintes des ressources en mémoire et capacité de calcul dans les réseaux de capteurs sans fils (RCSFs).

Nous avons utilisé MATLAB afin de simuler des déploiements aléatoires des RCSFs, et cela pour montrer la complexité en communication et le passage à l'échelle qui sont des métriques très importantes lorsqu'on désire proposer des solutions pour les RCSFs.

Dans cette contribution, nous allons simuler le réseau en différents types de déploiements. L'un aléatoire et l'autre uniforme sous forme d'un rectangle, et une distribution uniforme sous forme d'un cercle.

Nous étudions dans cette section deux distributions de capteurs, l'une uniforme et l'autre aléatoire. On suppose dans la proposition que

- la distribution est sur un espace de deux dimensions (X, Y) ,
- il n'y a pas d'obstacle physique entre les capteurs,
- les capteurs sont identiques (capacité de mémoire, d'énergie, porté...etc.),
- chaque capteur a une portée radio de longueur R sous forme de disque,
- la communication est bidirectionnelle et le réseau est non partitionné.

La distribution uniforme

En préconfigurant le nombre d'entités du système de la zone en forme carré tel que le nombre des nœuds $= n^2$ (n est entier), la superficie de la zone $= (n - 1)^2$ et la diamètre de la portée $= R$. On supposant que chaque nœuds initialement n'a aucune connaissance que, son identité, le fait qu'il soit le seul à posséder cette identité dans le réseau et un paramètre, d , définissant la dimension de l'espace de coordonnées.

1. Forme rectangulaire

- la distribution initiale : la figure 4.5 montre une distribution uniforme de $n = 10$ ($N = 100$ nœuds) dans un espace de 9 x 9.

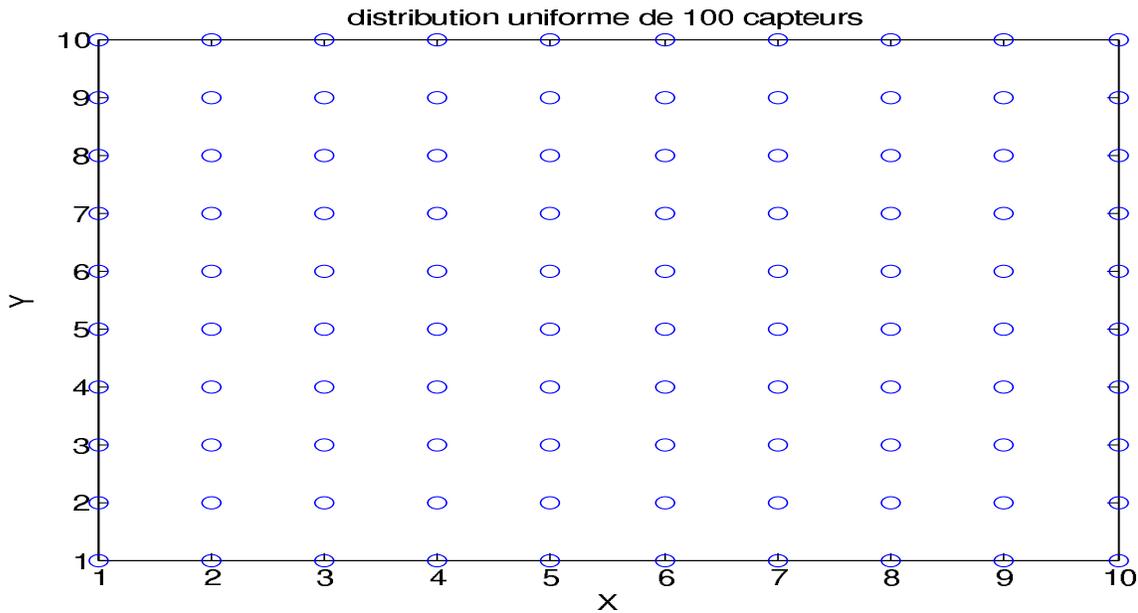


FIGURE 4.5 – Distribution rectangulaire de 100 capteurs

- élimination des nœuds du même voisinage : Il est intéressant de garder le réseau vivre longtemps possible, l'idée consiste à prolonger les capteurs qu'ils ont les mêmes voisins ou inclus dans un autre capteur. Alors les veilles capteurs vont mettre en actif lors de la disparition l'un de ses capteurs actifs. Donc ça serait intéressant d'en sélectionner que les nœuds critiques c'est-à-dire ceux dont l'exclusion entraîne une perte d'information. Pour cela chaque'un des capteurs S_i diffuse l'ensemble V_i à tous ses voisins et attend jusqu'à réception des ensembles V tous ses autres voisins pour commencer à comparer ces ensembles reçus avec son propre ensemble. S'il trouve que certains de ses voisins ont un voisinage inclus dans le sien lui seul participera dans les prochaines phases et les autres seront exclus.

Un autre cas est que S_i trouve d'autres dont le voisinage est le même que le sien alors le nœud dont l'identité est la plus grande participera et les autres seront exclus. Pour le reste des nœuds c'est-à-dire ceux dont le voisinage n'est ni inclus ni équivalent à celui de S_i ils sont considérés comme l'ensemble des nœuds $None_i$ (Not include not equal) de S_i . La figure 4.6 montre une distribution uniforme dans l'espace de 100 capteurs et une portée radio $= 3$. Il s'agit d'une matrice $voisin(i, j)$ représente le nombre de voisins de chaque nœuds. Puisque la représentation de les identifiants des voisins de 100 capteurs est énorme, la matrice $voisinIJ(i, j)$ montre dans La figure 4.7 les identifiants des voisins des 9 nœuds. A ce moment là chaque nœud i va comparer l'ensemble de ses voisins V_i avec l'ensemble V_j de chaque voisin j . La figure 4.8 représente les nœuds en veille(+) et les nœuds actif(o).

Le nombre de voisins de chaque nœud (i, j) est

voisin =

11	14	17	18	18	18	18	17	14	11
14	18	22	23	23	23	23	22	18	14
17	22	27	28	28	28	28	27	22	17
18	23	28	29	29	29	29	28	23	18
18	23	28	29	29	29	29	28	23	18
18	23	28	29	29	29	29	28	23	18
18	23	28	29	29	29	29	28	23	18
17	22	27	28	28	28	28	27	22	17
14	18	22	23	23	23	23	22	18	14
11	14	17	18	18	18	18	17	14	11

FIGURE 4.6 – Nombre de voisins de chaque nœud (i, j) avec $R = 3$

les voisins de chaque nœud sont

voisinsIJ =

111213212231	11121321222332	111213222333
11122122233132	111213212223313233	12132122233233
112122313233	12212223313233	132223313233

FIGURE 4.7 – Les voisins de chaque nœud (i, j) avec $R = 2$

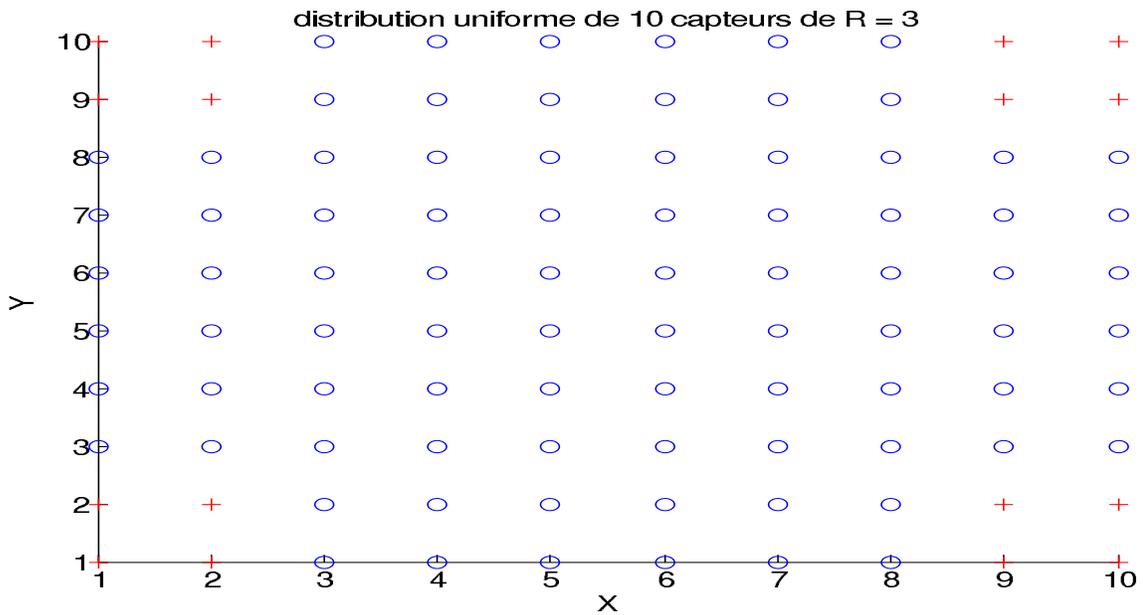


FIGURE 4.8 – Les nœuds veilles (+)

- détermination du nœud central : La figure 4.9 montre la position du nœud central par la méthode d'estimation de paramètre N .

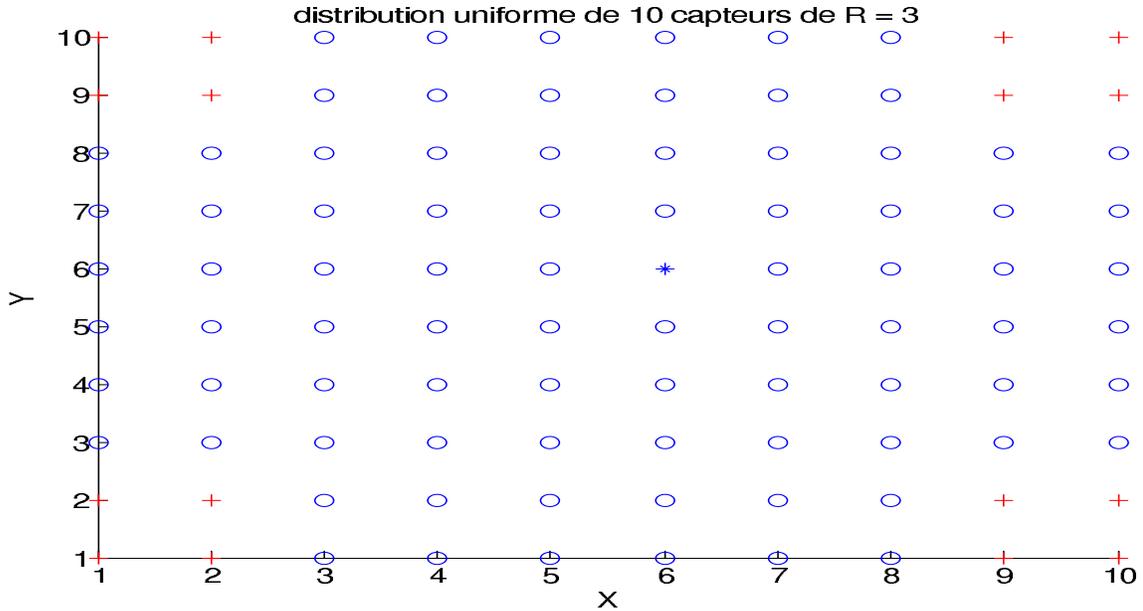


FIGURE 4.9 – Le nœud central (*)

- construction de l'arbre : La figure 4.10 représente une matrice d'arborescence, tel que $R = 2$, les zéros sont des nœuds mises en veilles, le un présente la racine(nœud central), et 4, 5 sont les feuilles de l'arbre.

Arbor =

0	5	4	4	3	4	4	5	0
5	4	4	3	3	3	4	4	5
4	4	3	3	2	3	3	4	4
4	3	3	2	2	2	3	3	4
3	3	2	2	1	2	2	3	3
4	3	3	2	2	2	3	3	4
4	4	3	3	2	3	3	4	4
5	4	4	3	3	3	4	4	5
0	5	4	4	3	4	4	5	0

FIGURE 4.10 – La propagation du message

2. Forme circulaire

Nous prenons cette fois une forme circulaire, la figure 4.11 présente la distribution initiale, la figure 4.12 présente les feuilles de l'arbre et la figure 4.13 présente l'arborescence.

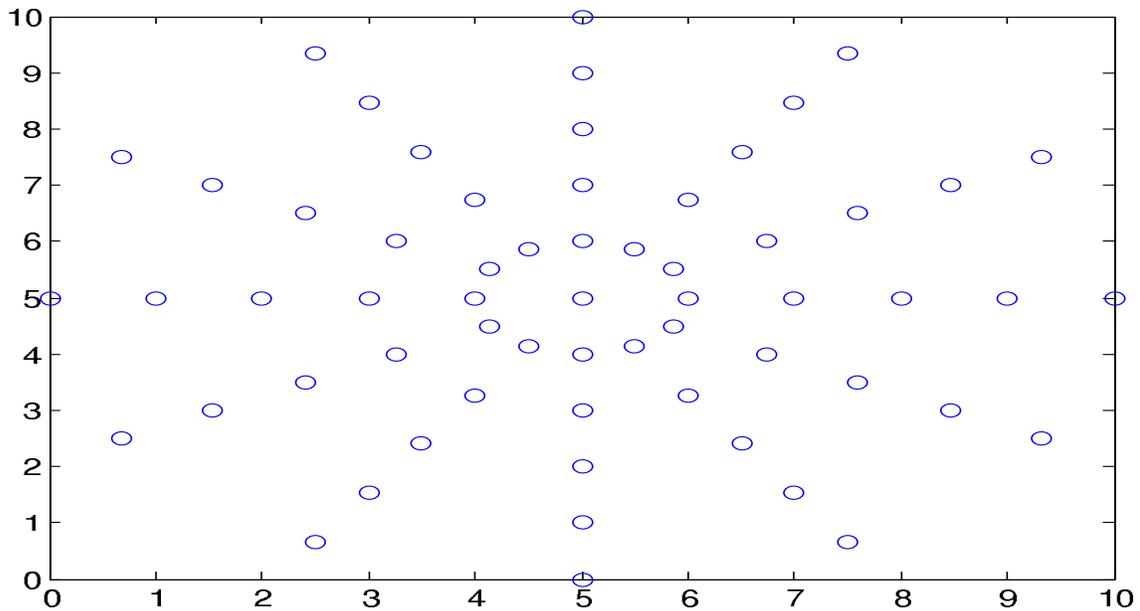


FIGURE 4.11 – Distribution circulaire

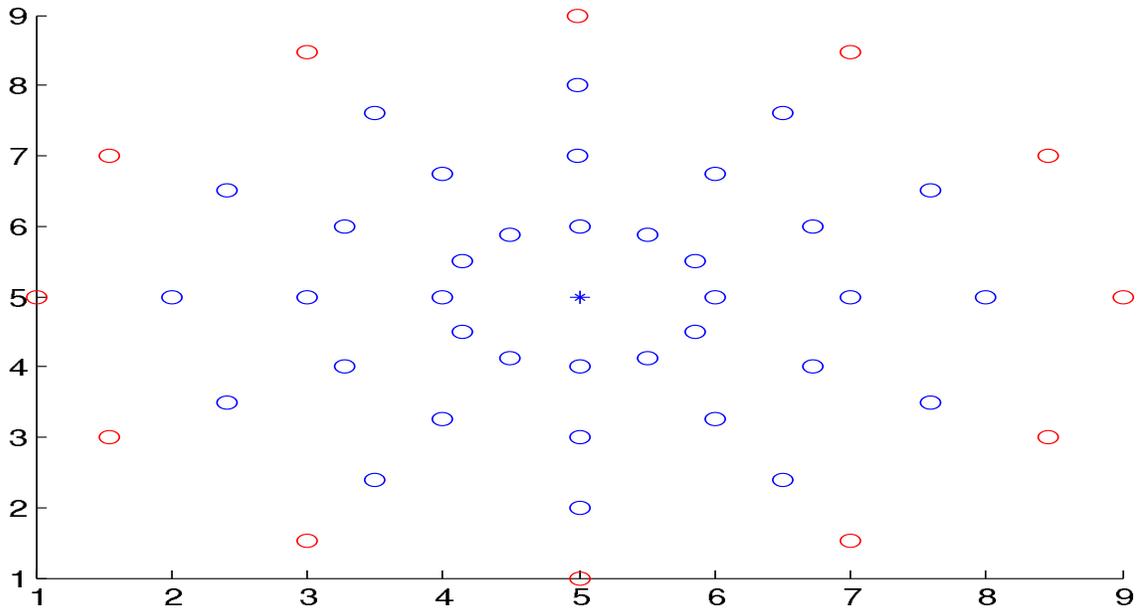


FIGURE 4.12 – Les feuilles de la forme circulaire

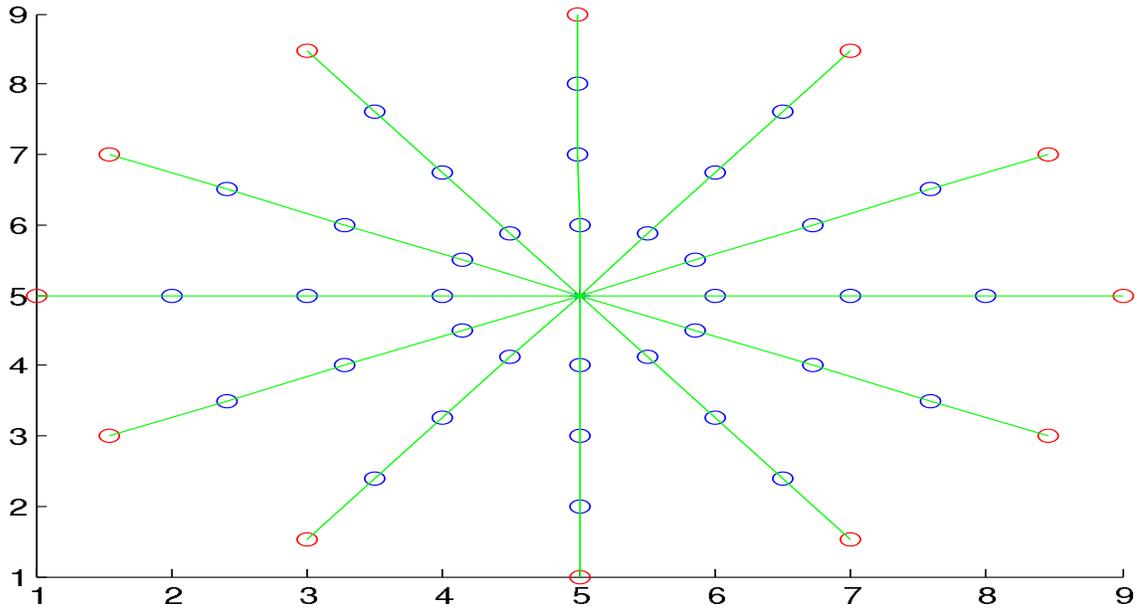


FIGURE 4.13 – L'arbre de la forme circulaire

La distribution aléatoire

Dans la section précédente nous avons étudié deux types uniformes, l'arborescence apparaît facile. Dans cette section nous allons prendre des distributions aléatoires.

en préconfigurant le nombre d'entités du système de la zone aléatoirement tel que le nombre des nœuds = n (n est entier) et la diamètre de la portée = R . On supposant que chaque nœuds initialement n'a aucune connaissance que, son identité, le fait qu'il soit le seul à posséder cette identité dans le réseau et un paramètre d , définissant la dimension de l'espace de coordonnées.

– la distribution initiale : la figure 4.14 montre une distribution aléatoire de $N = 100$ nœuds.

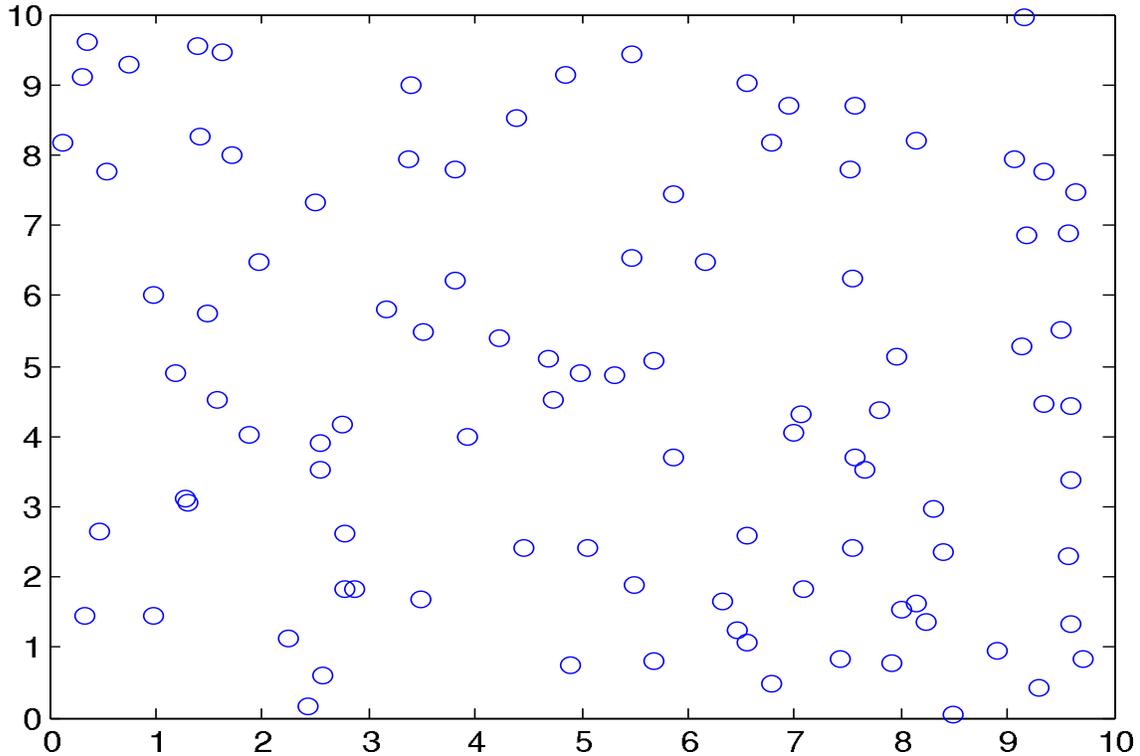


FIGURE 4.14 – La distribution aléatoire

L'histogramme suivant 4.15 montre la complexité en communication (nombre de messages) par chaque nœud capteur en fonction du nombre de voisins, à travers ce résultat, nous constatons que le nombre maximum de messages qui peuvent être reçus par un nœud du réseau est 37 messages. Si la valeur d'énergie consommée par chaque réception de message est connue, nous pouvons calculer le total d'énergie consommée.

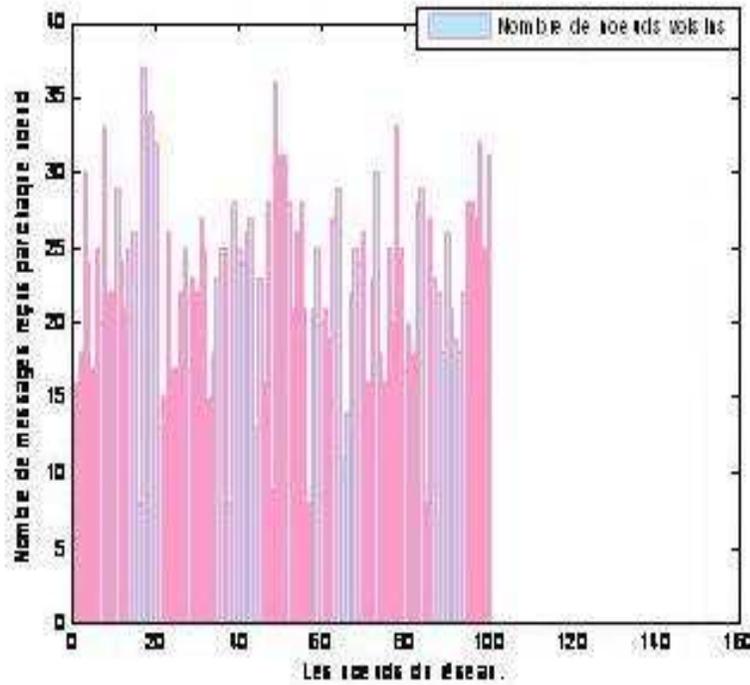


FIGURE 4.15 – Nombre de messages pour chaque nœud d'un réseau de 100 nœuds.

- élimination des nœuds du même voisinage : Pour cela chaque un des capteurs S_i diffuse l'ensemble V_i à tous ses voisins et attend jusqu'à réception des ensembles V tous ses autres voisins pour commencer à comparer ces ensembles reçus avec son propre ensemble. S'il trouve que certains de ses voisins ont un voisinage inclus dans le sien lui seul participera dans les prochaines phases et les autres seront exclus.

Un autre cas est que S_i trouve d'autres dont le voisinage est le même que le sien alors le nœud dont l'identité est la plus grande participera et les autres seront exclus. Pour le reste des nœuds c'est-à-dire ceux dont le voisinage n'est ni inclus ni équivalent à celui de S_i ils sont considérés comme l'ensemble des nœuds $Nine_i$ (Not include not equal) de S_i . La figure 4.16 montre une distribution aléatoire dans l'espace de 100 capteurs et une portée radio = 2.

A ce moment là chaque nœud i va comparer l'ensemble de ses voisins V_i avec l'ensemble V_j de chaque voisin j . la figure 4.16 présente les nœuds en veille(+) et les nœuds actif(o).

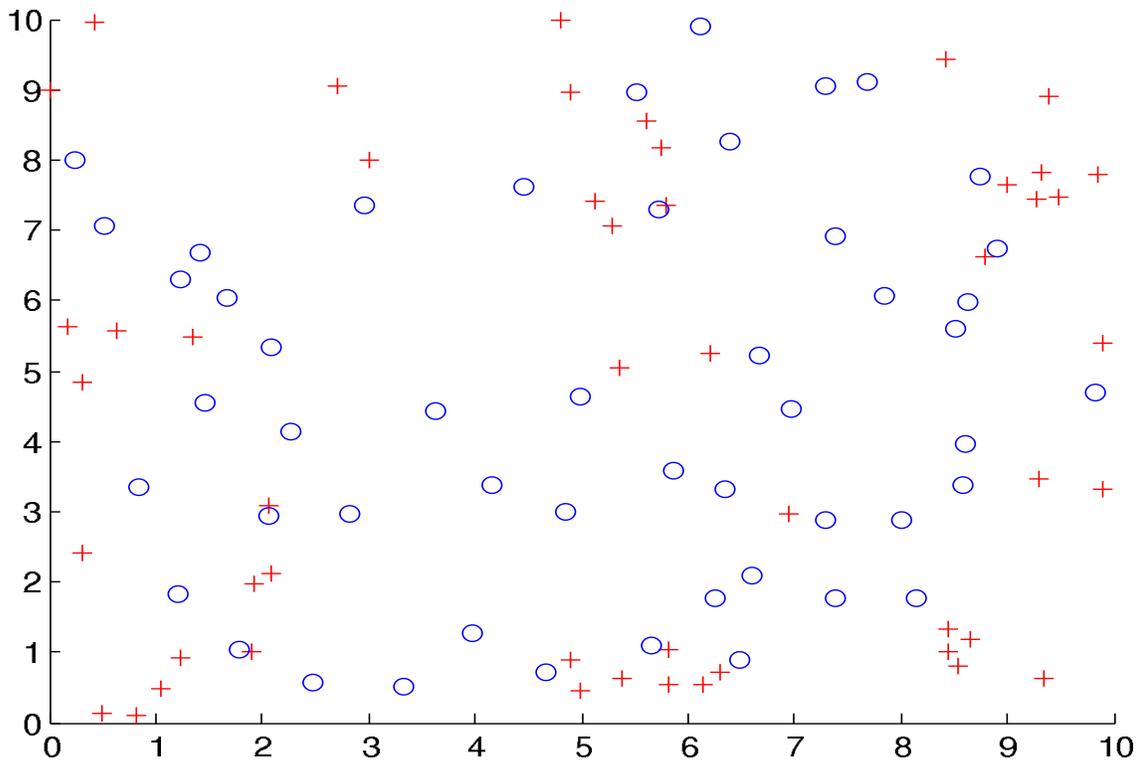


FIGURE 4.16 – Les capteurs en veilles

- détermination du nœud central : La figure 4.17 montre la position du nœud central par la méthode d'estimation de paramètre N.

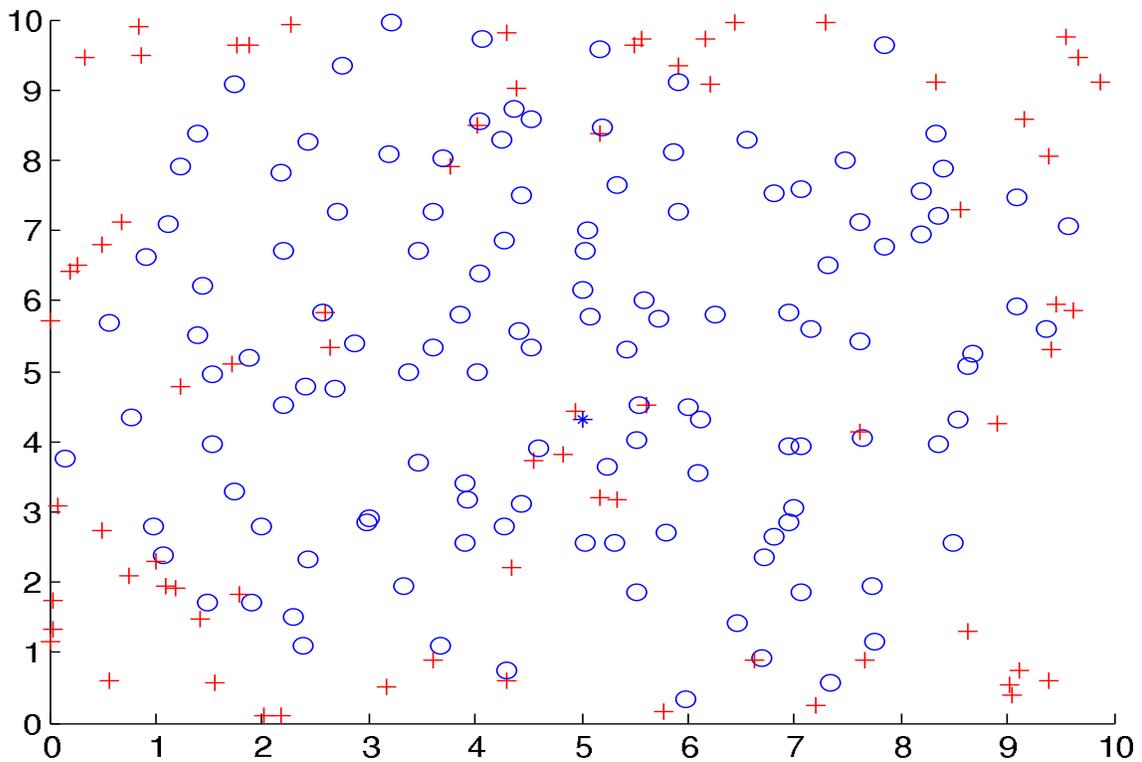


FIGURE 4.17 – Lencœud central

Nous pouvons dire que la taille des messages échangés (Identité, nombre de voisins, ensemble des voisins (limité)...) dans les différentes phases n'est pas grande et que le nombre maximum de messages ne dépasse pas 37. Le temps d'exécution dans ce déploiement (100 nœuds) est : $t = 16.224$ second

Nous avons augmenté le nombre de nœuds capteur déployés (à 300) pour évaluer le passage à l'échelle de notre proposition.

L'histogramme résultant de ce déploiement (300 nœuds) est présenté dans la figure 4.18. Où nous constatons qu'une augmentation par 50 nœuds résulte d'un ajout en moyenne de 10 messages engendrés par notre proposition.

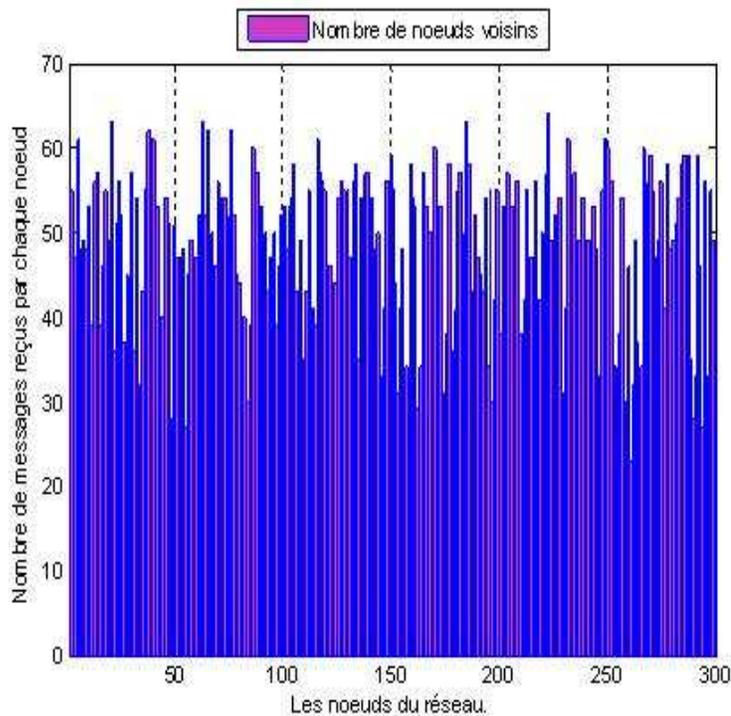


FIGURE 4.18 – Nombre de messages pour chaque nœud d'un réseau de 300 nœuds.

Après avoir réalisées différentes expériences sur des RCSFs de tailles variant de 50 à 300 nœuds, nous avons calculé le nombre maximum de nœuds pouvant être mis en veille et le nombre maximum de messages pouvant être reçus par un nœud en fonction du nombre des nœuds capteurs dans le réseau. Les deux figures 4.19 et 4.20 résument ces résultats :

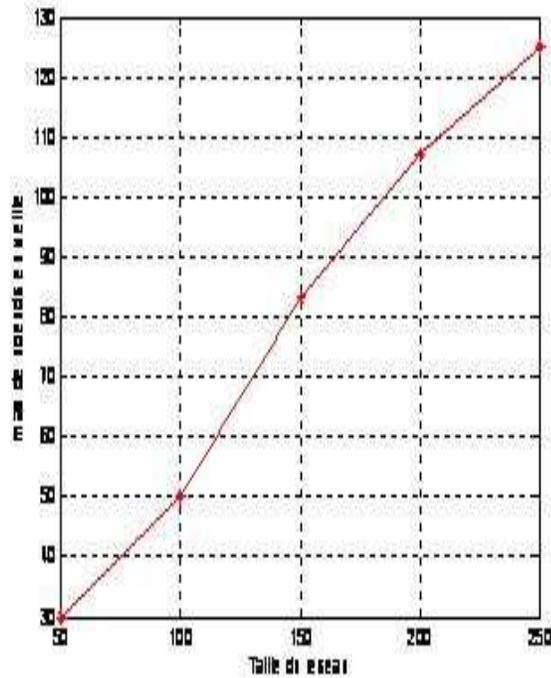


FIGURE 4.19 – Nombre maximum de nœuds mis en veille en fonction de la taille du réseau.

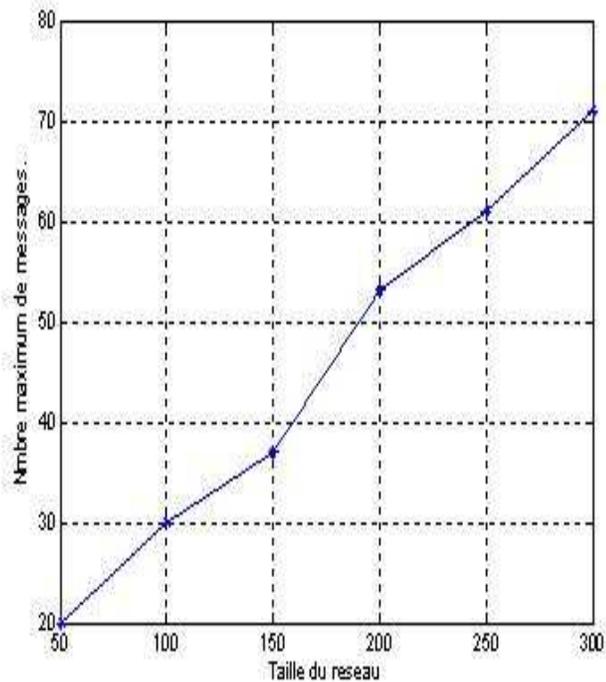


FIGURE 4.20 – Nombre maximum de message reçus par un nœud en fonction de la taille

– construction de l'arbre : La figure 4.21 présente l'arborescence, tel que $R = 2$, et les rouges sont

des nœuds feuilles.

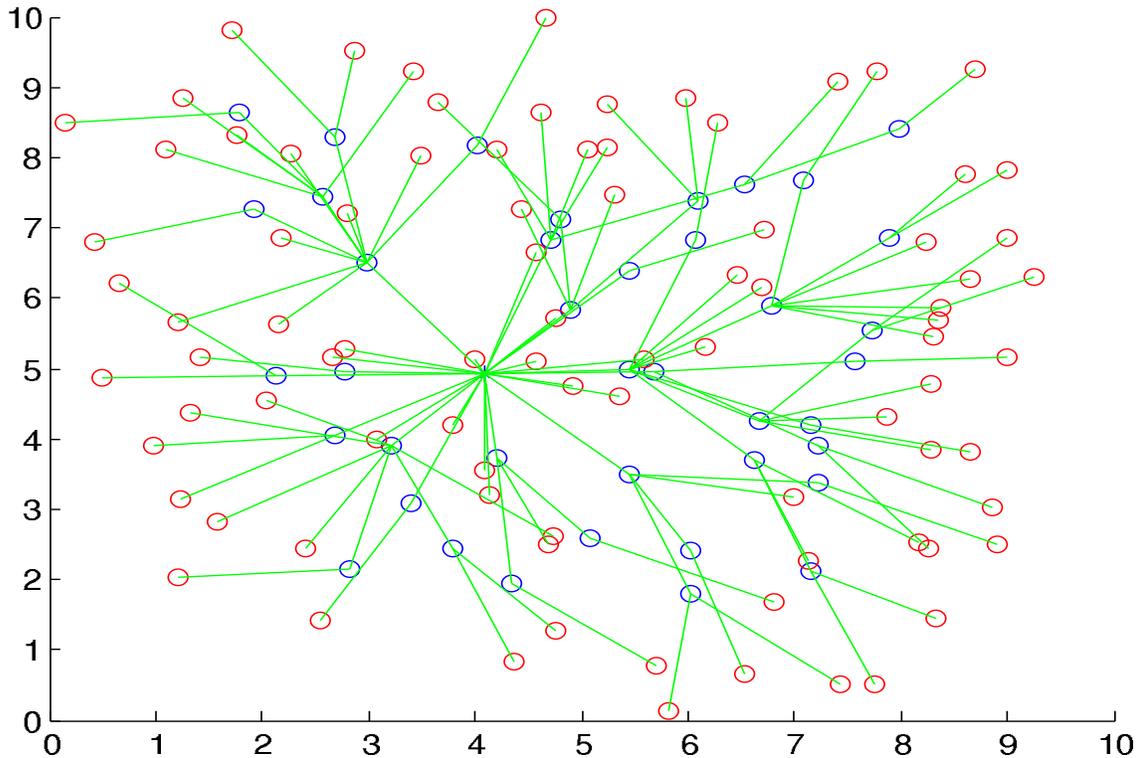


FIGURE 4.21 – L'arborescence du réseau

- La segmentation du réseau : Il s'agit d'une phase très importante, la figure ?? présente la partition du réseau en segments.

4.7 Cordonnées virtuelles pour les réseaux de capteurs

L'objectif est d'émerger une structure globale sans l'aide de l'extérieur. Il s'agit de prendre des cordonnées virtuelles cohérente avec la position physique et en plus économique en énergie, alors cette proposition répond à ces contraintes d'une manière complètement décentralisée. Chaque nœud est configuré initialement avec un paramètre d : la dimension du système de cordonnées (nombre de cordonnées), les cordonnées d'un nœud sont représentées par un tuple $(x_1, \dots, x_j, \dots, x_d)$, où x_j est la projection du nœud i sur le $j^{\text{ème}}$ axe de l'espace virtuel à d dimensions.

Dans cette proposition les cordonnées sont définis comme suit. d la dimension du système de cordonnées, chaque nœud prend sa dimension à travers le numéro du segment auquel il appartient, le numéro de sa branche et la distance entre lui et le centre (voir la figure 4.22, avec d est égal à 3), par exemple les cordonnées $(1, 1, 2)$ signifient que le nœud est appartient au segment 1, le numéro de sa branche est égale à 1 et il est distant de deux sauts du centre.

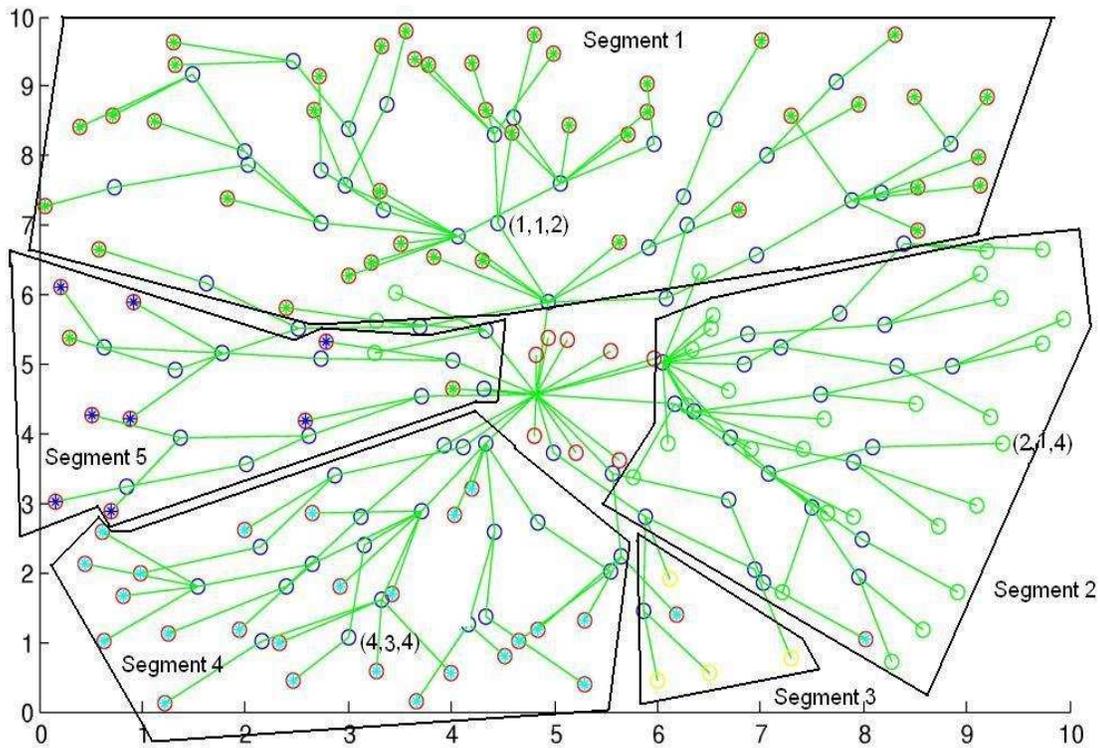


FIGURE 4.22 – La segmentation du réseau

La segmentation du réseau a été faite d'une manière spontanée, tels que les segments ils ont pas la même taille. Nous pouvons alors estimer la consommation d'énergie pour chaque capteur par l'équation suivante :

$$E(i) = Unit * \sum_{j=1}^n M_j \quad (4.11)$$

tel que :

Unit : est l'énergie consommé pour envoyer ou recevoir d'un message ;

j : le nombre de messages envoyés et reçus.

Nous avons segmenté le réseau spontanément qui intéressant en facteur énergétique. Mais la segmentation du réseau est mal structurée, nous allons dans l'étape suivante émerger une structure bien formée tels que tous les segments ont les mêmes tailles (voir la figure 4.23). Le principe de cette technique se déroule comme suit :

- chaque nœud feuille envoie un message de propagation qui contient son identité,
- le nœud feuille $F_i d$ ayant la plus petite identité lance un sondage pour calculer le nombre de feuilles,
- $F_i d$ estime le nombre de segments et la taille de chaque segment,
- $F_i d$ lance la segmentation.

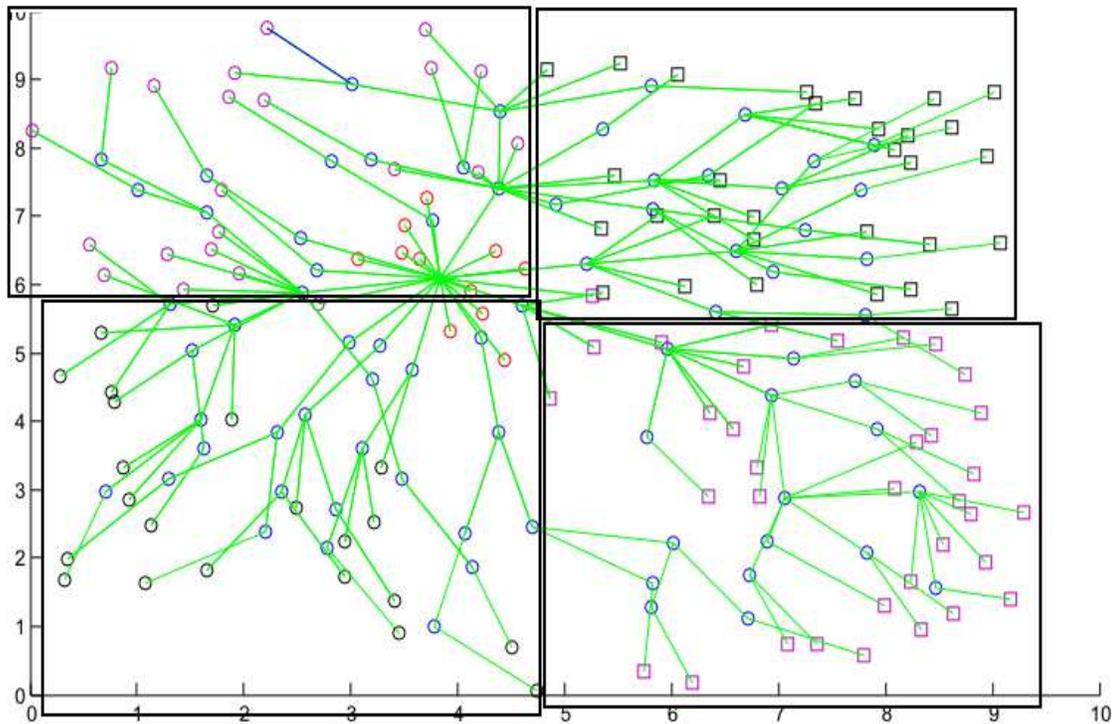


FIGURE 4.23 – Structuration avec des segments équivalents

4.7.1 Évaluation des résultats

Après l'étude de protocole de structuration VINCOS. et ressorti ses avantages et ses inconvénients surtout pour RCSFs, nous avons proposé ce protocole d'auto-structuration des RCSFs. Notre protocole a été conçu après une étude approfondie de VINCOS, nous avons remédié les inconvénients de ce dernier. Pour une bonne évaluation de ce protocole, le tableau ci-dessous montre une comparaison entre les deux protocoles. Notre proposition offre aussi d'autres avantages, tels que :

- un chemin unique existe lors de l'apparition d'un incident, ceci assure un routage efficace de donnée dans la couche réseau.
- la maintenance du réseau se fait d'une manière simple et efficace, qu'une propagation de message de reconstruction envoyée par le centre pour maintenir notre réseau.

Dans la section précédente nous avons présenté quelques histogrammes pour estimer la durée de vie de notre réseau. Nous pouvons conclure :

- Dans l'histogramme 4.15, le nombre maximum de messages reçus par un capteur est 37 messages et le nombre minimum est 8 messages. Donc le nombre moyen est 23 messages avec une distribution de 100 nœuds capteurs dans un espace de 100 mètres carré. Donc on peut calculer l'énergie moyenne consommée dans le réseau pour que chaque capteur détecte ses voisins $E_{\text{energie consommée}} = \text{nombre moyen de messages reçus} * \text{l'unité écoulé pour envoyer un message}$.
- L'histogramme 4.18 présente pour chaque nœud le nombre de messages reçus et le nombre de ses voisins. On constate que l'ajout de 50 nœuds capteurs résulte une augmentation de 10 message pour chaque capteur.
- Pour évaluer la valeur $\frac{\text{nombre total de capteurs}}{\text{nombre de capteurs en veille}}$, les histogrammes 4.19 et 4.20 montrent le nombre maximum des nœuds mis en veille et le nombre de messages reçus en fonction de la taille du réseau. On conclut que le nombre de capteurs mis en veille est = 40 % de nombre total.

Donc, on peut calculer l'intérêt de cette technique de la façon suivante :
tel que :
 T est la durée de vie de notre réseau.
 T' est la nouvelle durée de notre réseau.
 $T_{écoulé}$ est le temps écoulé pour détecter les capteurs mis en veilles.

$$T' = T + \frac{\text{nombre total de capteurs}}{\text{nombre de capteurs en veilles}} - T_{écoulé} \quad (4.12)$$

<i>VINCOS</i>	<i>Notre proposition</i>
<ul style="list-style-type: none"> - La détection de bordure aux travers des initiateurs de bordure et un nombre indéterminé d'itérations pour une découverte complète de la bordure. - VINCOS suffit un sondage pour segmenter le réseau. - La disparition d'un nœud de bordure peut empêcher le fonctionnement de réseau dans un temps important pour reconstruire la bordure. - Les trous dans le réseau posent un défi important surtout dans la détection de la bordure. Les nœuds qu'il sont au tour du trou sont considérés comme des nœuds de bordure. 	<ul style="list-style-type: none"> - La détermination du nœud central à travers de deux techniques (paramètre N et 1-densité) et la détection de la bordure au travers au centre. Une seule itération suffit pour la découverte toute la bordure. - On peut segmenter le réseau automatiquement, mais avec des segments variants. - la disparition d'un nœud feuille n'a aucun impact à la structure de réseau. - Nous avons aussi minimisé le nombre des capteurs participants, pour garder le réseau vivre longtemps possible.

4.8 Conclusion

Dans ce chapitre, nous avons présenté notre proposition. On a d'abord modélisé le réseau par un graphe non orienté, ceci montre les liens entre les nœuds et aussi pour calculer la densité de notre réseau. Par la suite on a donné un algorithme d'auto-structuration qui prend l'énergie comme une ressource critique. L'algorithme a été décomposé en cinq phases pour simplifier son principe et afin de simuler le déroulement des capteurs, on a utilisé MATLAB pour montrer la complexité en communication et le passage à l'échelle.

Il s'agit de structurer le réseau et mettre des coordonnées virtuelles à chaque nœud capteur d'une manière autonome (qui est un grand défi pour les RCSFs). Deux types de déploiements ont été étudiés dans ce chapitre ; l'un uniforme pour montrer que l'algorithme donne des résultats corrects, et aussi pour bien comprendre le principe de notre proposition, et l'autre a été implémenté dans une distribution aléatoire tel que notre objectif est de mettre des coordonnées virtuelles à chaque capteur. Après la structuration du réseau, deux techniques ont été présentées afin de le segmenter ; la première permet de segmenter le réseau d'une manière spontanée avec moins de consommation d'énergie, et la deuxième consiste à segmenter le réseau au travers d'un sondage entre les nœuds feuilles pour obtenir des segments de tailles identiques. Enfin, nous avons mis les coordonnées virtuelles de telle manière que chaque nœud capteur i prend sa position avec les paramètres suivants : le numéro de son segment d'appartenance, le numéro de la branche à laquelle il appartient et la distance qui le sépare du centre.

Conclusion générale et perspectives

L'objectif, au travers de ce mémoire, a été de développer un protocole de structuration des réseaux sans fil à larges échelles afin de répondre aux besoins liés aux RCSFs. Pour cela, on a proposé une solution distribuée qui se décompose en plusieurs phases. Chacune de ces phases répond à une application de tels réseaux tout en considérant leurs contraintes. Chaque algorithme est né d'une étude des contraintes ou des structures et se compose d'algorithmes distribués, auto-structurants et robustes, nécessitant peu de ressources en énergie, bande passante ou taille mémoire.

On a d'abord étudié le réseau par une méthode analytique qui consiste à modéliser le réseau par un graphe non orienté. Après la modélisation, toutes ces phases ont été étudiées par simulation. En effet, la faible capacité énergétique habituellement assignée à chaque nœud, et l'impossibilité d'une intervention humaine, font que ces réseaux doivent être capables de s'auto-structurer et de manière autonome et sans aucune entité externe. Le travail consigné dans ce mémoire a été le fruit d'une recherche menée dans le contexte des RCSFs et ce relativement au problème d'auto-structuration.

Enfin, les performances d'un réseau sans fil sont très contraintes par l'utilisation de communications radio. Or les simulations ne permettent que de donner un aperçu des propriétés inhérentes à un tel environnement. Seules des expérimentations dans un environnement réel permettent réellement d'apprécier les performances d'un protocole donné. Nous avons donc souhaité compléter l'évaluation de performances analytique et par simulations menée précédemment. Le protocole de localisation tirant parti de cette auto-structuration a également été implémenté. Nous avons pu ainsi confirmer que ces protocoles présentent des performances intéressantes, corroborant le fait qu'une auto-structuration permet de simplifier les protocoles tout en améliorant leurs performances.

Perspectives

Les différents algorithmes proposés dans chaque étape de notre travail présentent tous des caractéristiques qui peuvent être étudiées plus en profondeur ou des points qui mériteraient certaines optimisations. Par exemple :

- Le comportement de chaque algorithme pourrait être analysé dans un environnement plus mobile.
- Un autre point important est que toutes les analyses réalisées sur les différents algorithmes de cette proposition ont été conduites en supposant une couche MAC idéale et un modèle «Unit Disk Graph» pour définir le voisinage des nœuds. Un autre défi est d’imaginer une couche MAC non idéale qui permet de n’étudier que les caractéristiques propres de l’algorithme considéré.
- De même, l’aire de transmission d’un nœud est rarement un cercle, puisque la propagation des ondes dépend du milieu et des obstacles que peut rencontrer le signal. Il serait donc intéressant d’évaluer les différents algorithmes proposés dans ce manuscrit, en considérant différents protocoles pour les couches inférieures et différents modes de propagation des ondes suivant les environnements, aussi bien en ce qui concerne les analyses par simulation que les analyses théoriques.

En effet, les réseaux de capteurs offrent un certain nombre de défis et de verrous scientifiques. Ils sont le reflet de l’évolution à la fois des systèmes, des réseaux, de leurs composants mais aussi de leur organisation et des interactions et communications entre systèmes. Malgré des tailles souvent petites, ils intègrent une forte complexité, notamment de par l’infrastructure logicielle qui se retrouve distribuée à une grande échelle et qui se doit d’offrir des services d’auto-adaptabilité.

D’autre part, une auto-structuration paraît présenter un avantage certain dans l’exploitation d’un RCSF. Cependant, qu’en est-il des autres domaines ? Un réseau mobile hétérogène possède des contraintes au regard des RCSFs : l’hétérogénéité de la topologie est plus élevée et le contrôle est fait par une station de base, comment une auto-structuration conçue pour les réseaux de capteurs peut-elle être adaptée aux réseaux ad-hoc ? Allons même plus loin : un réseau ad-hoc nécessite-t-il une auto-structuration ?

Bibliographie

- [1] A. Ward, A. Jones, and A. Hopper : *A new location technique for the active office* IEEE Personal Communications, 4(5) :42-47, October 1997.
- [2] Anne-Marie Kermarrec, Achour Mostefaoui, Michel Raynal, Gilles Tredan, and Aline C. Viana *Large-Scale Networked Systems : From Anarchy to Geometric Self-structuring* INRIA Rennes-IRISA, Université de Rennes 1-IRISA, INRIA Saclay, France. 2008
- [3] Andy Hopper, Andy Harter, and Tom Blackie : *The active badge system (abstract)* In Conference on Human factors in computing systems (CHI), pages 533-534, Amsterdam, The Netherlands, April 1993. ACM Press.
- [4] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro : *A Novel Lightweight Algorithm for Time-Space Localization in Wireless Sensor Networks* in Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, October 2007, pp. 336-343.
- [5] A. Magnani and K. Leung : *Self-organized, scalable gps-free localization of wireless sensors* In IEEE Wireless Communications and Networking Conference (WCNC), pages 3798-3803, Hong Kong, China, March 2007.
- [6] B. N. Clark, C. J. Colbourn, and D. S. Johnson. *Unit Disk Graphs. Discrete Mathematics* vol.86, pp.165-177, 1990.
- [7] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System : Theory and Practice.* Springer, 2001.
- [8] Charl jaco leuschner : *The design of a simple energy efficient routing protocol to improve wireless sensor network lifetime* University of pretoria, avril 2005
- [9] C. A. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. *On the scalability of ad hoc routing protocols.* In INFOCOM, New York, USA, June 2002. IEEE.
- [10] Self-organization in communication networks C. Prehofer and C. Bettstetter : *Self-organization in communication networks.* principles and design paradigm. IEEE Communications Magazine, 43(7) :78-85, July 2005.

-
- [11] Cerpa and D. Estrin : *Adaptive Self-Configuring sSensor Networks*. Topologies, Proceedings of INFOCOM 2002, New York, NY, USA (2002)
- [12] D. Niculescu and B. Nath : *DV Based Positioning in Ad hoc Networks*, Kluwer Journal of Telecommunication Systems, vol. 22, no. 1-4, pp. 267-280, January 2003.
- [13] Fabrice VALOIS. *Auto-organisation de réseaux radio multi-sauts* L'INSA de Lyon et l'Université Claude Bernard - Lyon 1. Le 19 novembre 2007.
- [14] F. Heylighen. *Principles of systems and cybernetics* an evolutionary perspective. Cybernetics and Systems, 23 :3-10, 1992.
- [15] G. Yu and F. Yu : *A Localization Algorithm for Mobile Wireless Sensor Networks* in Proceedings of IEEE International Conference on Integration Technology, March 2007, pp. 623-627.
- [16] I. Stojmenovic : *Handbook of Sensor Networks : Algorithms and Architectures* ISBN : 978-0-471-68472-5, Ed. : Wiley Interscience, 2005.
- [17] Ian F. AKYILDIZ, Weilian SU, Yogesh SANKARASUBRAMANIAM et Erdal CAYIRCI : *Wireless sensor networks* a survey. Elsevier Computer Networks Journal, 38(4) :393 – 422, mars 2002.
- [18] Jamal N. Al-Karaki, Ahmed E. Kamal *Routing techniques in wireless sensor network : a survey* dept. of electrical and computer engineering, Iowa State University
- [19] K. Heurtefeux *PROTOCOLES LOCALISÉS POUR RÉSEAUX DE CAPTEURS* 2009
- [20] Karel Heurtefeux and Fabrice Valois : *Protocoles d'auto-organisation : une étude qualitative au cours de la vie des réseaux de capteurs* ARES INRIA CITI,INSA-Lyon,F-69621,France
- [21] L. Hu and D Evans : *Localization for mobile sensor networks* In 10th annual international conference on Mobile computing and networking (MobiCom), pages 45-57, Philadelphia, Pennsylvania, USA, September 2004. ACM Press
- [22] L. Li and T. Kunz : *Cooperative Node Localization Using Non-linear Data Projection* ACM Transactions on Sensor Networks, vol. 5, no. 1, pp. 1-26, February 2009.
- [23] Marcos Augusto M. VIEIRA, Dio genes Cec í lio DA SILVA JUNIOR, Claudionor N. COEHLO JUNIOR et José M. DA MATA, *Survey on wireless sensor network devices*.
- [24] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, *The Cricket Location-Support System* in Proceedings of 6th Annual ACM International Conference on Mobile Computing and Networking (MOBICOM), August 2000, pp. 32-43.
- [25] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller : *Anchor-Free Distributed Localization in Sensor Networks* MIT LCS, Technical Report TR-892 June 2008.

- [26] N. Bulusu, J. Heidemann, and D. Estrin : *Gps-less low-cost outdoor localization for very small devices* IEEE Personal Communications Magazine, 7(5) :28-34, October 2000.
- [27] P. Bahl and V. Padmanabhan : *RADAR : An in-building RF-based user location and tracking system* in Proceedings of IEEE International Conference on Computer Communications (INFOCOM), March 2000, pp. 775-784.
- [28] Q. Cao and T. Abdelzaher : *Scalable logical coordinates framework for routing in wireless sensor networks* ACM Transactions on Sensor Networks, 2(4) :557-593, November 2006.
- [29] Stéphane DURAND : *Amélioration de la précision de la localisation différentielle temps réel par mesure de phase des systèmes GNSS* janvier 2003
- [30] S. Capkun, M. Hamdi, and J-P. Hubaux : *Gps-free positioning in mobile ad-hoc networks* In 34th Annual Hawaii International Conference on System Sciences
- [31] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher : *"Range-free Localization Schemes for Large Scale Sensor Networks,"* in *Proceedings of International Conference on Mobile Computing and Networking* (ACM MOBICOM), September 2003, pp. 81-95.
- [32] W. Chen et al., *A Localization Algorithm Based on Discrete Imprecision Range Measurement in Wireless Sensor Networks* in Proceedings of 2006 IEEE International Conference on Information Acquisition, August 2006, pp. 644-648.
- [33] Yann BUSNEL : *Systèmes d'information pair-à-pair pour les réseaux de capteurs larges échelles* Novembre 2008
- [34] Y. Shang and W. Ruml *Improved MDS-Based Localization* in Proceedings of IEEE International Conference on Computer Communications (INFOCOM), March 2004, pp. 2640-2651.
- [35] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz : *Localization from Connectivity in Sensor Networks* IEEE Transactions on Parallel and Distributed Systems, vol. 15, no. 11, pp. 961-974, November 2004.

L'auto-structuration dans les Réseaux de Capteurs Sans Fil

Résumé

L'apparition des applications de surveillances dans l'industrie ces dernières années, et l'impossibilité de l'être humain à intervenir pour maintenir ou renouveler l'énergie, ont poussé un grand nombre de chercheurs à proposer des solutions à économie d'énergie au niveau de la pile protocolaire dans la perspective de minimiser la dépense énergétique nodale et de maximiser la durée de vie du réseau global d'une manière complètement autonome.

Dans ce mémoire, nous avons proposé un protocole de structuration pour les réseaux des capteurs sans fil. En fait, il n'existe qu'un seul sujet dans la littérature pour la structuration décentralisée des nœuds capteurs, il a été proposé par un groupe de l'IRISA. Ces chercheurs ont pu développer un protocole d'auto-structuration des RCSFs, VINCOS, ce protocole offre beaucoup d'avantages aux réseaux installés dans des zones hostiles où l'intervention humaine n'est pas possible, tel que les coordonnées virtuelles. En revanche, ce protocole consomme beaucoup plus d'énergie surtout dans la phase « initialisation de la bordure », cette phase contient un nombre indéterministe d'itérations.

Pour cette raison, notre contribution comprend un protocole qui prend en compte la consommation d'énergie comme une ressource critique. Une seule itération suffit pour déterminer la bordure, mais nous avons abordé au centre. Ce dernier était déterminé au travers d'une technique simple et efficace, il s'agit de déterminer le nœud central en calculant le paramètre de convergence N . En plus, nous avons minimisé le nombre de capteurs avec la couverture maximale de la zone d'intérêt. L'idée maîtresse est de mettre en veille chaque capteur qui a une table de voisinage incluse dans (ou égale à) une table de l'un de ses voisins, et que tous les autres nœuds vont se mettre actifs jusqu'à la disparition de l'un de ces nœuds, un capteur voisin en veille va le remplacer afin de garder le réseau vivant le plus longtemps possible.

Mots clés : *Réseaux sans fil, Réseaux de capteurs, Découverte de voisinage, Auto-structuration, Systèmes autonomes.*

Self-structuring in Wireless Sensors Networks

Abstract

The appearance of surveillance applications in industry in recent years, and the inability of humans to intervene to maintain or renew energy, this has led many researchers to propose solutions to saving energy on the level of the protocol stack with a prospect to minimize the nodal energy expenditure and to maximize the lifespan of the total network in a completely autonomous way.

In this memory, we proposed a protocol structure for wireless sensor networks WSN. Indeed, there is one subject in the literature for structuring decentralized sensor nodes, was proposed by the group of "IRISA". They have developed a protocol for self-structuring of WSNs, VINCOS, this protocol offers many advantages to networks installed in hostile areas where human intervention is not possible, such as virtual coordinates. However, this protocol consumes much more energy especially in the phase "initialization of the border", this phase includes an indeterminate number of iterations.

From this reason, we proposed a protocol in our contribution taken into account the energy consumption as a critical resource. One iteration to determine the border (the leaves of the tree), but we approached the center. In addition we also minimized the number of sensors with the maximum coverage area of interest. The idea is standby each sensor has a neighbor table included or equal in a neighbor table of its neighbors, and all other nodes will remain active, to the disappearance of one these nodes. At the disappearance of one of the active nodes, a neighbor standby sensors will replace to keep the network live a long time possible.

keywords : *wireless networks, wireless sensor networks, neighbor discovery, Self-structuring, Autonomous systems.*