

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE MOULOU D MAMMERI DE TIZI OUZOU
FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE



Mémoire de Fin d'Etudes

En vue d'obtention du Diplôme MASTER II en Informatique
Option: Réseaux, Mobilité et Systèmes Embarqués (RMSE)

Thème

**Proposition et implémentation d'un
protocole d'agrégation de données basé sur
l'approche de clustering dans les RCSFs**

Proposé et dirigé par:

Mme AOUDJIT. R

Réalisé par:

Mlle CHEMOUN Karima

Promotion: 2012-2013

Remerciements

Je tiens à remercier Dieu tout puissant qui m'a armé de courage, de volonté, de patience et de bonne santé afin de mener à bon terme ce travail.

J'exprime ma sincère gratitude à toutes les personnes qui ont rendu ce mémoire possible par leurs aides et leurs contributions.

Mes premiers remerciements sont adressés à ma promotrice Mme AOUDJIT RACHIDA pour m'avoir proposé ce sujet et de l'avoir dirigé. Ses conseils et ses encouragements m'ont été précieux.

Je tiens également à remercier les membres du jury qui ont eu l'amabilité d'examiner et de juger mon mémoire.

Je garde une place toute particulière à mes parents, je leurs exprime toute ma profonde reconnaissance, ils m'ont constamment aidé par leur soutien moral et leurs encouragements pour achever ce travail.

Je tiens à remercier tous mes amis plus particulièrement Louiza, Rachida et Jean-Bosco pour toute l'aide qu'ils m'ont apportée et leur soutien.

Enfin, ce projet s'est déroulé comme je l'ai souhaité. Je suis ravie d'avoir pu effleurer un niveau aussi poussé de perfection dans le domaine de l'informatique.

J'espère en profiter encore...

RESUME

Les réseaux de capteurs sans fil (RCSFs) sont composés d'un ensemble de dispositifs électroniques appelés "capteurs" à coût raisonnable, de taille réduite et ressources limitées. Ils sont, généralement, déployés de manière dense dans des environnements hostiles afin de surveiller différents phénomènes physiques. En raison des caractéristiques contraignantes de ces réseaux, plusieurs travaux de recherche ont été menés pour minimiser la consommation d'énergie au niveau des capteurs afin de maximiser leur durée de fonctionnement et, par conséquent, allonger la durée de vie du réseau en entier.

Le but de ce projet est de proposer puis implémenter un protocole d'agrégation de données dans les réseaux de capteurs sans fil basé sur l'approche de clustering. Selon cette approche, les nœuds du réseau sont groupés en clusters. Pour chaque cluster, un nœud dit cluster-head est élu pour gérer les nœuds membres de son propre cluster.

Le protocole proposé repose sur un principe visant à gérer l'énergie consommée dans le réseau, en appliquant la technique d'agrégation sur les données collectées au niveau des nœuds cluster-heads. Ce mécanisme permet d'éliminer les redondances qui peuvent exister en raison du déploiement dense des nœuds capteurs.

Mots clés: réseau de capteurs sans fil, gestion d'énergie, clustering, agrégation de données.

TABLE DES MATIERES

Introduction générale.....	1
CHAPITRE I: Etat de l'art sur les RCSFs	
Introduction	3
I.1 Modèle d'un capteur.....	3
I.1.1 Définition.....	3
I.1.2 Caractéristiques d'un capteur	4
I.1.3 Architecture physique d'un capteur	4
I.1.4 Système d'exploitation d'un capteur	6
I.2 Réseaux de capteurs sans fil	6
I.2.1 Définition.....	6
I.2.2 Architecture des RCSFs.....	7
I.2.3 Caractéristiques des RCSFs.....	9
I.3 Comparaison entre les RCSFs et les réseaux ad-hoc	9
I.4 Pile protocolaire	10
I.4.1 Rôle des couches	11
I.5 Application des RCSFs	13
I.6 Facteurs et contraintes des RCSFs	15
Conclusion.....	17
CHAPITRE II: Clustering dans les RCSFs	
Introduction	18
II.1 Notions de base et motivations	18
II.1.1 Définitions	18
II.1.2 Construction d'une topologie en cluster	20
II.2 Avantages de l'approche de clustering	24
II.3 Quelques Contraints et facteurs de conception lors du processus de clustering.....	24
II.4 Classification de paramètres liés au processus de clustering	26
II.4.1 Propriétés des clusters	28
II.4.2 Capacité d'un cluster-head	29
II.4.3 Processus de clustering.....	29
II.5 Quelques approches de clusterisation.....	30
II.5.1 Les protocoles hiérarchiques.....	31
II.5.2 Les protocoles pondérés.....	33
II.5.3 Les protocoles heuristiques	34

TABLE DES MATIERES

Conclusion.....	36
CHAPITRE III: Agrégation de données dans les RCSFs	
Introduction	37
III.1 Motivation	37
III.2 Fusion de données et Agrégation de données	39
III.3 Agrégation de données	39
III.3.1 Définition.....	39
III.3.2 Principe.....	39
III.4 Approches d'agrégation	40
III.4.1 Agrégation avec réduction de la taille	41
III.4.2 Agrégation sans réduction de la taille.....	41
III.5 Types d'agrégation de données dans les RCSFs.....	41
III.5.1 Agrégation basée sur le flux dans le réseau.....	42
III.5.2 Agrégation basée sur la qualité de service (QoS).....	42
III.5.3 Agrégation basée sur la topologie du réseau	42
III.6 Quelques approches d'agrégation de données basées sur les réseaux hiérarchiques.....	48
III.6.1 LEACH: Low Energy Adaptive Clustering Hierarchical	48
III.6.2 HEED: Hybrid Energy Efficient Distributed clustering approach	49
III.6.3 TAG: Tiny Aggregation	50
III.6.4 PEGASIS: Power-Efficient Data Gathering protocol for Sensor Information System	50
III.7 Impact de l'agrégation de données sur les RCSFs.....	51
Conclusion.....	52
CHAPITRE IV: Description du protocole proposé	
Introduction	54
IV.1 Principe du protocole	54
IV.2 Processus de conception du protocole	55
IV.2.1 Découverte du voisinage	55
IV.2.2 Phase de clustering	55
IV.2.3 Phase de collecte de données.....	62
IV.2.4 Phase d'agrégation.....	63
IV.3 Organisation	64
IV.3.1 Données.....	64
IV.3.2 Les structures de données utilisées.....	64
IV.3.3 Interactions: Description des messages.....	65

TABLE DES MATIERES

Conclusion.....	67
CHAPITRE V: Implémentation et simulation du protocole proposé	
Introduction	68
V.1 Différentes approches de test.....	68
V.1.1 Test en environnement réel.....	68
V.1.2 Simulation	69
V.1.3 Emulation.....	69
V.2 Simulateur J-Sim	70
V.2.1 Présentation de J-Sim.....	70
V.2.2 Architecture détaillée de J-Sim.....	74
V.2.3 Installation de J-Sim	75
V.2.4 Intégration du code source du protocole dans J-Sim	75
V.3 Définition de quelques méthodes du protocole	76
V.4 Exécution et simulation	76
V.5 Evaluation du protocole proposé	77
V.5.1 Environnement de simulation	77
V.5.2 Modèle énergétique	77
V.5.3 Paramètres de simulation	77
V.5.4 Métriques d'évaluation de performance.....	78
V.5.5 Interprétation des résultats	80
Conclusion.....	81
Conclusion générale	82

LISTE DES FIGURES

Figure I.1: Exemples de nœuds capteurs

Figure I.2: Architecture physique d'un capteur

Figure I.3 : Exemple de réseaux de capteurs sans fil

Figure I.4: Topologie plate

Figure I.5: Topologie hiérarchique

Figure I.6: Pile protocolaire

Figure I.7: Exemples d'applications des RCSFs

Figure II.1 Hiérarchie d'un RCSFs

Figure II.2: Exemple de structure en clusters

Figure II.3 Exemple d'une communication CH-BS à 1-saut

Figure II.4 Exemple d'une communication multi-sauts

Figure II.5: Classification des paramètres liés au processus de clustering

Figure III.1: Exemple d'agrégation de données

Figure III.2: Agrégation de données basée sur la topologie du réseau

Figure III.3: Agrégation de données basée-cluster

Figure III.4: Agrégation de données basée-arbre

Figure III.5: Agrégation de données basée-chaîne

Figure III.6: Agrégation de données basée-grille

Figure III.7: La phase de communication dans LEACH

Figure IV.1: Élection des cluster-Heads

Figure IV.2: Formation des clusters

Figure IV.3: Etapes de construction d'une topologie en clusters

Figure IV.4: Illustration de la collecte des données au niveau CHs

Figure V.1 Connexion entre les composants dans J-Sim

LISTE DES FIGURES

Figure V.2 Vue générale des trois types de nœuds dans J-Sim

Figure V.3 Architecture interne du nœud sensor dans J-Sim

Figure V.4: Moyenne d'énergie consommée

Figure V.5. Délai moyen de transmission des paquets de données

LISTE DES TABLEAUX

Tableau I.1 : Comparaison entre RCSFs et réseaux ad-hoc

Tableau IV.1: Structure de la table de voisinage des nœuds

Tableau IV.2: Structure de la liste des cluster-heads

Tableau IV.3: Les différentes données d'un nœud

Tableau IV.4: Entête du paquet Hello

Tableau IV.5: Entête du paquet Weight

Tableau IV.6: Entête du message d'annonce

Tableau IV.7: Entête du message de jointure

Tableau IV.8: Entête du message de données

Tableau V.1: Paramètres de simulation

ACRONYMES

ACA:	Autonomous Component Architecture
ASIC:	Application Specific Integrated Circuit
CAN:	Convertisseur Analogique Numérique
BS:	Base Station (Station de Base)
CH:	Cluster-Head
DCA:	Distributed Clustering Algorithm
DRCL:	Distributed Realtime Computing Laboratory
DSP:	Digital Signal Processors
FEC:	Forward Error Correction
FPGA:	Field Programmable Gate Array
GPS:	Global Positioning System
HCC:	Highest Connectivity Cluster algorithm
HEED:	Hybrid Energy Efficient Distributed clustering approach
JVM:	Java Virtual Machine
LEACH:	Low Energy Adaptive Clustering Hierarchical
MAC:	Medium Access Control
OS:	Operating System
PEGASIS:	Power Efficient Gathering protocol for Sensor Information System
PDA:	Personal Digital Assistant
QoS:	Quality of Service
RAM:	Random Access Memory

ACRONYMES

RCSF:	Réseau de Capteurs Sans Fil
ROM:	Read Only Memory
RUV:	Runtime Virtual
SAR:	Sequential Assignment Routing
SMP:	Sensor Management Protocol
SPIN:	Sensor Protocol for Information Negotiation
SQDDP:	Sensor Query and Data Dissemination Protocol
TADAP:	Task Assignment and data Advertisement Protocol
TAG:	Tiny Aggregation
TCL:	Tool Command Line
TDMA:	Time Division Multiple Access
WCA:	Weighted Clustering Algorithm
WSN:	Wireless Sensor Network

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Les réseaux de capteurs sans fil (RCSFs) sont considérés comme une des plus importantes technologies récentes, en raison des progrès remarquables réalisés dans les domaines des communications sans fil et de systèmes électroniques.

L'utilisation des capteurs n'est pas une nouveauté. En effet, de nombreux domaines en font usages, principalement dans l'aéronautique, l'industrie pétrolière, l'automobile, etc [58]. Ce qui est nouveau, c'est leur miniaturisation grandissante, l'abandon du coûteux ainsi que le câblage encombrant traditionnellement utilisé. Cette nouvelle génération de micro-capteurs est dotée de moyens permettant de collecter des données, les stocker, traiter et les communiquer à un centre de collecte via un canal radio. Les différentes caractéristiques de ces micro-capteurs ainsi que leur diversité (optique, thermique, etc.) ont ouvert de nouvelles perspectives applicatives très larges et très variées pour les réseaux de capteurs dans de nombreux domaines (militaires, domotiques, environnementales, etc).

La préoccupation principale dans pratiquement la majorité des travaux sur les réseaux de capteurs est la réduction de la consommation énergétique. En effet, les réseaux de capteurs sans fil sont, le plus souvent, densément déployés dans des zones hostiles sans aucune intervention humaine. Ainsi, les capteurs proches peuvent intercepter les mêmes données ou des données fortement corrélées. En transmettant ces données redondantes, une énergie supplémentaire est dissipée. Il est donc nécessaire de trouver un moyen permettant la conservation de cette ressource critique.

Afin de répondre à cette problématique, nous proposons un nouveau protocole utilisant une technique d'agrégation de données sur un réseau hiérarchisé en clusters. Cela consiste à combiner les données redondantes ou fortement corrélées afin d'alléger le trafic du réseau.

Pour mener à bien notre travail, nous l'avons organisé en cinq chapitres selon le plan méthodologique suivant:

- **Premier chapitre:** Etat de l'art sur les RCSFs

Dans ce chapitre, nous allons donner une présentation générale sur les réseaux de capteurs sans fil, expliciter les différents concepts de base, discuter les différentes contraintes et facteurs à prendre en considération lors de la conception, puis exposer quelques domaines d'application de ce type de réseaux.

INTRODUCTION GENERALE

– **Deuxième chapitre:** Clustering dans les RCSFs

Dans ce chapitre, nous nous intéresserons à un mécanisme de structuration virtuelle du réseau en un ensemble de clusters. Pour cela, nous définissons ce processus et nous explicitons les différentes étapes à suivre. Pour finir avec des exemples d'approches, étudiées dans la littérature, utilisant ce processus.

– **Troisième chapitre:** Agrégation de données dans les RCSFs

Au cours de ce chapitre, nous allons décrire la technique d'agrégation, énumérer ses différents types, présenter les différents concepts qui lui sont relatifs et finir par quelques exemples de protocoles utilisant cette technique.

– **Quatrième chapitre:** Description du protocole proposé

Ce quatrième chapitre consiste en une étude détaillée de l'approche que nous avons proposée pour remédier au problème de conservation d'énergie dans les RCSFs, afin de percer et comprendre son fonctionnement.

– **Cinquième chapitre:** Implémentation et simulation du protocole proposé

Dans cette dernière partie, nous allons décrire l'outil de simulation choisi pour la mise en œuvre de notre protocole, puis nous présentons les résultats générés après tests et évaluer ses performances.

ETAT DE L'ART SUR LES RCSEs

Introduction

Depuis quelques années, le besoin de superviser des environnements hostiles et d'observer des phénomènes physiques (température, pression,...) est devenu crucial pour différentes applications: militaires, scientifiques, industrielles, etc.

Grâce aux avancées technologiques, des mini-composants appelés "capteurs" ont été mis en place. Ceux-ci sont dotés de moyens leurs permettant de collecter des informations du milieu où ils ont été déployés, les traiter puis les transmettre au monde réel via une communication sans fil. Donc, il est devenu tout à fait possible de déployer un grand nombre de nœuds capteurs collaboratifs constituant un réseau de capteurs sans fils (RCSF) pour surveiller l'environnement.

I.1 Modèle d'un capteur

I.1.1 Définition

Un capteur [1] est un dispositif électronique qui transforme une grandeur physique observée (température, pression, lumière, etc) en une grandeur utilisable (tension, résistance, vitesse, etc). Pour cela, il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre. La figure suivante illustre des exemples de nœuds capteurs.

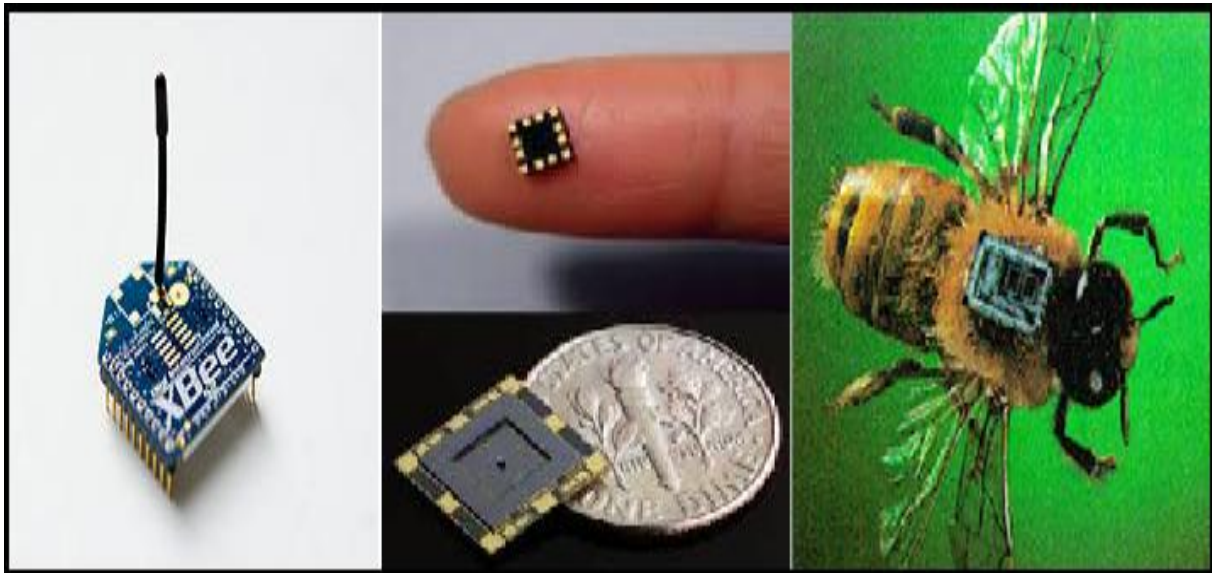


Figure I.1: Exemples de nœuds capteurs

I.1.2 Caractéristiques d'un capteur

Un nœud capteur a les caractéristiques suivantes [3]:

- Petite taille;
- Coût réduit;
- Ressources énergétiques basses;
- Mémoire limitée;
- CPU lente;
- Communication radio (Bande-passante faible, Portée radio courte).

I.1.3 Architecture physique d'un capteur

Suivant le type d'application, il existe une multitude de capteurs sur le marché: les capteurs de température, d'humidité, de pression, etc. Cependant, malgré cette diversité apparente, ils restent dotés d'une architecture matérielle similaire. Un nœud capteur se compose de quatre unités de base: unité d'acquisition, unité de traitement, unité de communication et unité d'énergie. D'autres unités dites optionnelles peuvent être intégrées selon le domaine et le besoin d'application. Parmi ces unités, un capteur peut être doté d'une unité de localisation (GPS: Global Positioning System, par exemple), d'un générateur d'énergie (cellules solaires, par exemple) et d'une unité assurant la mobilité [2].

La figure ci-dessous illustre les différents composants d'un capteur.

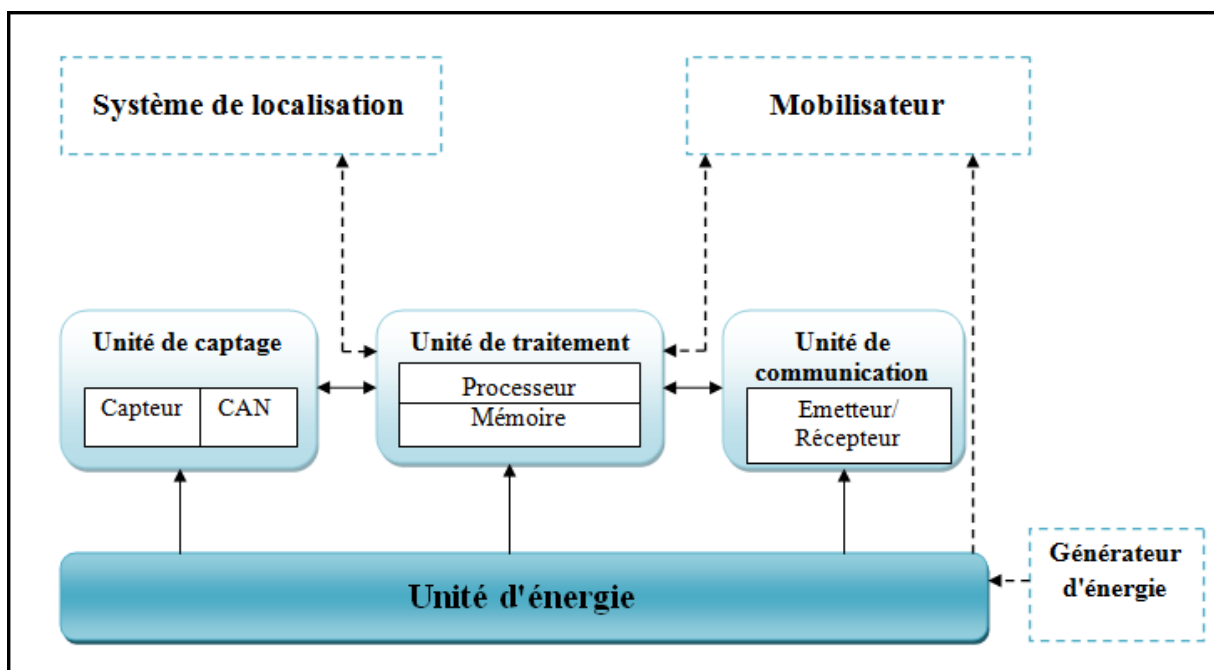


Figure I.2: Architecture physique d'un capteur [2]

– **Unité de captage:**

Cette unité a pour rôle principal de capturer ou mesurer les données physiques à partir de l'objet cible. Elle est composée de deux sous-unités : le capteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique (CAN). Ce dernier transforme ces signaux en données numériques et les transmet à l'unité de traitement. Un capteur peut disposer d'une ou plusieurs unités de captage [15].

– **Unité de traitement:**

Elle est composée d'une unité de calcul (processeur) qui effectue des traitements sur les données reçues (si nécessaire), et d'une unité de stockage (mémoire) limitée en termes de taille.

L'unité de calcul recueille les données de l'unité de captage ou d'autres capteurs, effectue un traitement sur ces données (si nécessaire) et décide quand et où les envoyer. Les types de processeurs qui peuvent être utilisés dans un capteur incluent le Microcontrôleur, les DSP (Digital Signal Processors), les FPGA (Field Programmable Gate Array) et les ASIC (Application Specific Integrated Circuit) [15]. Cependant, le Microcontrôleur a été le processeur le plus utilisé pour les capteurs à cause de sa flexibilité à être reliés à d'autres composants (comme, par exemple, l'unité de communication), à son bon prix et sa faible consommation énergétique [4] [15] [16].

L'unité de stockage inclut la mémoire de programme (dont les instructions sont exécutées par le processeur) et la mémoire de données (pour conserver des données fournies par l'unité de captage et d'autres données locales). La taille de cette mémoire est souvent limitée essentiellement par les considérations économiques et s'améliorera aussi probablement au fil des années [5].

– **Unité de communication**

Elle est responsable de toutes les tâches d'émissions et réceptions des données à travers un support de communication sans fil (optique, infrarouge ou fréquences radio) qui représente l'état actif du nœud. Par ailleurs, les nœuds peuvent se mettre en veille ou écouter seulement le trafic. L'unité de communication est l'unité qui consomme le plus d'énergie par rapport aux unités précédentes [5].

– Unité d'énergie

Un capteur est muni d'une source d'énergie, généralement une batterie [17], pour alimenter tous ses composants. Les batteries utilisées sont soit rechargeables ou non. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs et donc celle du réseau global.

I.1.4 Système d'exploitation d'un capteur:

Un système d'exploitation (OS: Operating System) est un ensemble de programmes responsables d'assurer l'exploitation des ressources matérielles d'un dispositif par les applications utilisateur. Les OS classiques ne sont pas appropriés aux nœuds capteurs vu leur caractéristiques. Donc, le nombre de systèmes d'exploitation destinés aux réseaux de capteurs est minime, le plus répandu de ces systèmes est TinyOS.

TinyOS [4] [5] [10] est le système d'exploitation open source développé à l'Université de Berkeley pour des applications fonctionnant sur des dispositifs appelés "motes". Ce système est mis en application avec le langage de programmation NesC et fournit une conception modulaire fortement efficace bien convenue à ces dispositifs. Sa bibliothèque de composants est particulièrement complète puisqu'elle inclut des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. Ces composants peuvent être utilisés tel qu'ils sont, comme ils peuvent être aussi adaptés à des applications précises. En s'appuyant sur un fonctionnement événementiel, TinyOS propose une gestion de la consommation d'énergie du capteur et permet une meilleure adaptation à la communication sans fil.

I.2 Réseaux de capteurs sans fil

I.2.1 Définition

Un réseau de capteurs sans fils (RCSFs) ou wireless sensor networks (WSNs) est un ensemble de nœuds capteurs variant de quelques dizaines d'éléments à plusieurs milliers, déployés de manière dense dans un environnement appelé "champ de captage ou zone d'intérêt" en vue de collection et de transmission des données vers un ou plusieurs points de collecte riches en énergie appelés "station de base" ou "puits". Cette station récupère les données, les stocke puis les met à la disposition des utilisateurs (ordinateur central) en les envoyant par Internet ou par satellite pour les analyser et prendre décisions [6].

Un exemple d'un réseau de capteurs sans fil est illustré dans la figure suivante:

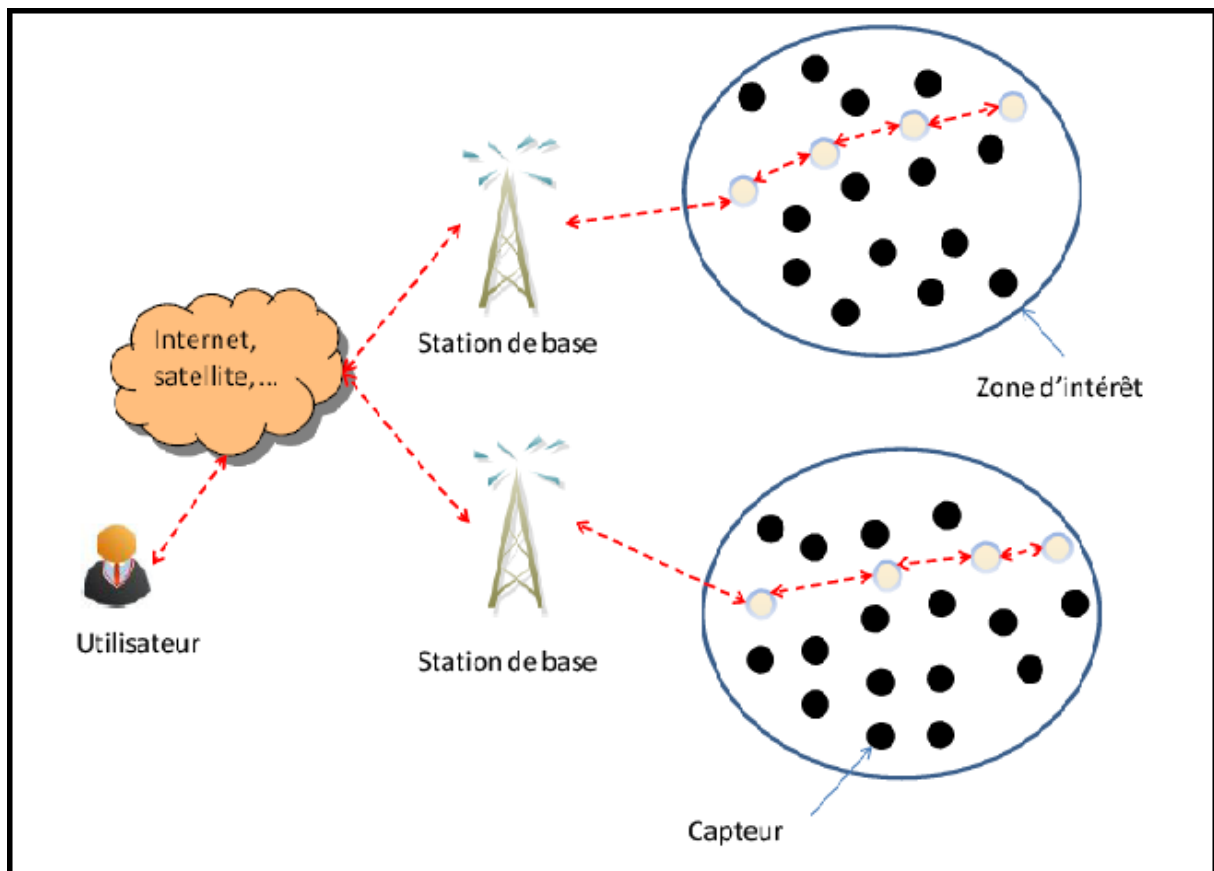


Figure I.3 : Exemple de réseaux de capteurs sans fil [7]

I.2.2 Architecture des RCSFs

Typiquement, un réseau de capteurs sans fil peut être organisé en topologie plate ou hiérarchique [6].

– Topologie plate

Dans une topologie plate, tous les nœuds assurent le même rôle et collaborent entre eux pour accomplir la tâche de routage. Cette topologie est appliquée aux petits réseaux qui sont facile à concevoir, à implémenter et à maintenir. Une fois que le nombre de nœuds soit important, cette topologie devient dure à contrôler; donc, ces réseaux présentent une faible scalabilité. La figure suivante illustre l'organisation des capteurs dans une topologie plate.

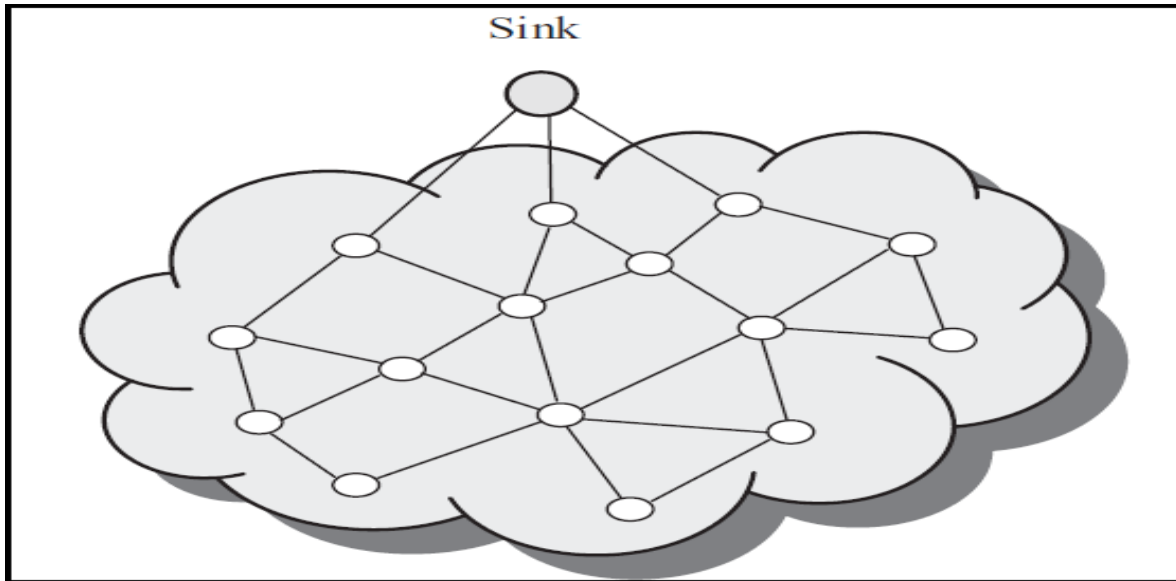


Figure I.4: Topologie plate [6]

– **Topologie hiérarchique**

Dans une topologie hiérarchique, les nœuds capteurs sont organisés en groupes (clusters), où les membres d'un cluster envoient leurs données à leurs propres nœuds chefs tandis que ces derniers servent de relais à transmettre les données à la station de base. Un nœud ayant une petite puissance énergétique peut être employé pour accomplir la tâche de captage et pour envoyer les données senties au nœud chef de son propre cluster à distance courte, alors qu'un nœud doté d'une énergie plus élevée puisse être choisi comme nœud chef pour traiter les données de ses membres et pour transmettre ces données traitées à la station de base.

Ce processus peut non seulement réduire la consommation d'énergie pour la communication, mais également équilibre la charge du trafic et améliore la scalabilité quand la taille de réseau se développe. La figure qui suit illustre un exemple d'une topologie hiérarchique.

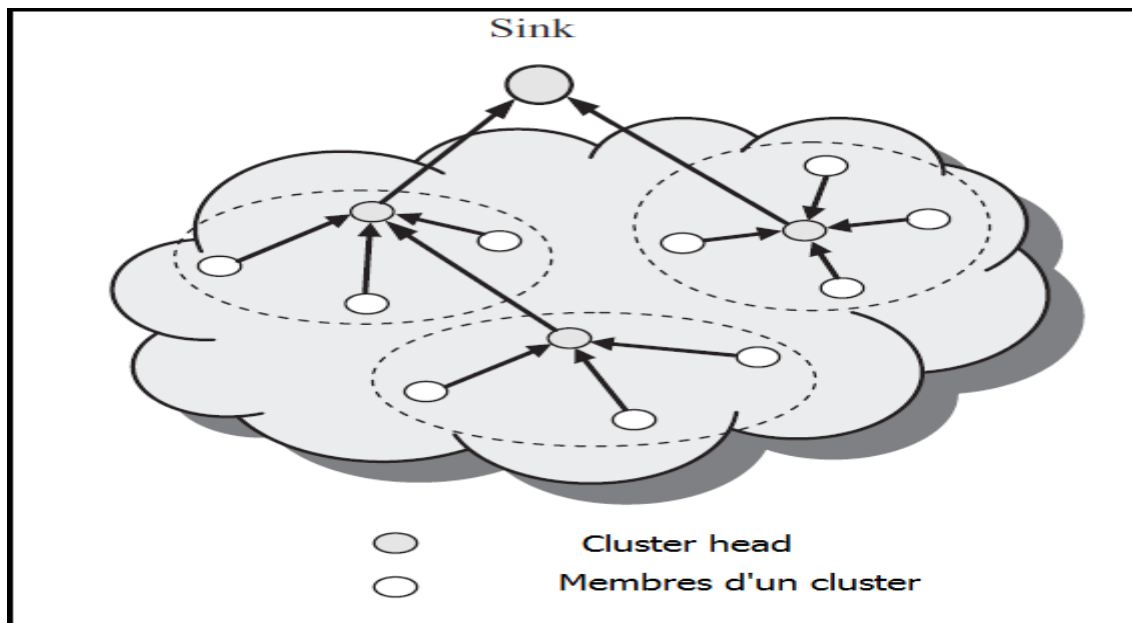


Figure I.5: Topologie hiérarchique [6]

I.2.3 Caractéristiques des RCSFs

Ces réseaux sont caractérisés par [19]:

- Forte densité de nœuds: nombre de nœuds est bien plus élevé que dans un réseau traditionnel;
- Les nœuds capteurs sont limités en termes de puissance de calcul, capacité et taille mémoire;
- Faible énergie et durée de vie limitée;
- Auto-organisation;
- Topologie dynamique en raison des changements fréquents (ajout, déplacement, défaillance des capteurs, etc.);
- Manque de sécurité.

I.3 Comparaison entre les RCSFs et les réseaux ad-hoc

Comme les réseaux de capteurs sans fil sont apparentés aux réseaux ad hoc, ces deux types de réseaux ont de nombreux points communs, citons l'absence d'infrastructure, l'autonomie et l'utilisation des ondes radio pour la communication [11]. Mais les contraintes et les besoins de ces réseaux diffèrent.

Le tableau suivant résume les différences entre ces deux types de réseau.

Caractéristiques	RCSF	Réseau ad hoc
Composition	Petits micro-capteurs	Portables, PDA
Nombre de nœuds	Très grand (de l'ordre de mille)	Moyen (de l'ordre de cent)
Relation entre les nœuds	Collaboratifs pour un même objectif	Chacun a son propre objectif
Contrainte clé	Ressource énergétique	Débit et qualité de service (QoS)
Capacité de traitement et de stockage	Moindre	Plus importante
Les pannes	Plus susceptible aux pannes	Moins susceptible aux pannes
Communication	Diffusion (broadcast)	Point à Point
Flot de communication	Plusieurs à un (many to one)	Plusieurs à plusieurs (many to many)
Mobilité	Faible	En constante évolution et mobilité forte

Tableau I.1: Comparaison entre RCSFs et réseaux ad-hoc

I.4 Pile protocolaire

Les nœuds capteurs sont en général déployés dans une zone de surveillance d'une manière aléatoire, chacun d'eux possède une capacité d'acquérir des données, les traiter puis les acheminer vers le nœud puits et par la suite vers l'utilisateur via Internet ou Satellite.

Dans le but d'un établissement efficace d'un réseau de capteurs sans fil, une architecture en couches est adoptée afin d'améliorer la robustesse du réseau. Cette pile prend en charge le problème de la consommation énergétique, intègre le traitement des données transmises dans les protocoles de routage, et facilite le travail coopératif entre les capteurs [12].

Elle est donc composée de cinq couches [2] [6]: application, transport, réseau, liaison de données et physique. Ainsi que trois plans: gestion d'énergie, gestion de mobilité et gestion des tâches.

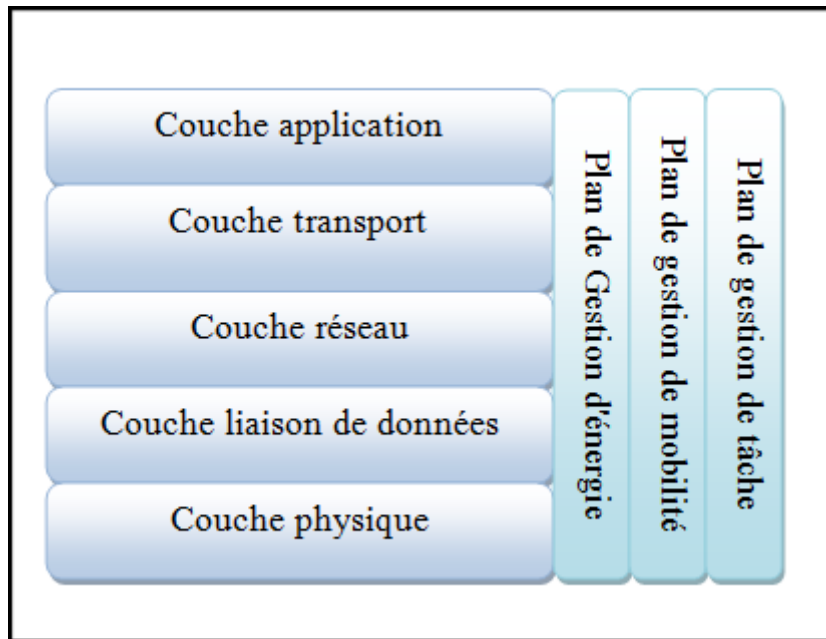


Figure I.6: Pile protocolaire

I.4.1 Rôle des couches

– Couche application

Cette couche constitue l'ensemble des applications implémentées dans un RCSF. Elle doit fournir des mécanismes pour permettre à l'utilisateur d'interagir avec le réseau à travers des interfaces, et éventuellement, par l'intermédiaire d'un réseau étendu (Internet à titre d'exemple), tout en restant transparente par rapport aux matériels et aux mécanismes de communication dans les couches inférieures [13]. Parmi les protocoles d'application, nous citons: SMP (Sensor Management Protocol), TADAP (Task Assignment and Data Advertisement Protocol) et SQDDP (Sensor Query and Data Dessiminitation Protocol) [1].

– Couche transport

La couche transport est l'interface entre la couche réseau et la couche application. Elle se charge du cheminement des données, leur découpage en paquets, contrôle de flux, conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

– Couche réseau

Cette couche a pour but d'établir des routes entre les différents nœuds du réseau et sélectionner le meilleur chemin en termes d'énergie, délai de transmission, débit, etc. Parmi les protocoles conçus au niveau de cette couche [2] : LEACH (Low Energy Adaptive Clustering Hierarchy), SAR (Sequential Assignment Routing), SPIN (Sensor Protocol for Information via Negotiation), etc.

– **Couche liaison de données**

Cette couche est responsable du multiplexage des flux de données, la détection des trames, le contrôle des accès au média de transmission et le contrôle d'erreurs. Elle établit également des connexions point à point ou point à multipoints fiables durant la communication entre les nœuds capteurs.

Citons parmi les protocoles dédiés à cette couche [2]: MAC (Medium Access Control), FEC (Forward Error Correction), etc.

– **Couche physique**

Cette couche est responsable du canal de transmission, de la sélection de fréquences, de la génération des ondes porteuses, de la détection du signal, de la modulation et cryptage des données. Deux médias peuvent être utilisés pour communiquer dans un réseau de capteurs sans fil: l'infrarouge et la radiofréquence [14].

Trois autres couches supplémentaires sont intégrées dans la pile protocolaire décrite ci-dessus:

– **Plan de gestion d'énergie**

Cette entité gère la manière dont les nœuds exploitent leurs énergies.

Comme la durée de vie du nœud a une dépendance forte à l'égard de la vie de la batterie, il doit donc contrôler et minimiser sa consommation d'énergie.

Par exemple, un nœud capteur met son Emetteur/Récepteur en mode off après avoir reçu un message de son voisin afin d'éviter la réception des messages dupliqués. De plus, quand un nœud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de captage [12].

– **Plan de gestion de la mobilité**

Un système de gestion de mobilité est indispensable dans un réseau de capteurs sans fil. Un tel système doit être capable de détecter et enregistrer tous les mouvements des nœuds capteurs afin de maintenir des informations sur leurs localisations et d'entretenir continuellement une route vers l'utilisateur final.

– **Plan de gestion de tâches**

Un système de gestion des tâches permet d'assurer une répartition équilibrée des tâches entre les différents nœuds capteurs afin de garantir un travail coopératif et efficace en termes de la consommation énergétique, et delà, prolonger la durée de vie du réseau.

I.5 Application des RCSFs

L'évolution remarquable que les réseaux de capteurs sans fil aient connue (capteurs de taille très réduite, faible coût, variés, etc) leur a permis d'envahir notre vie quotidienne et marquer leur utilité dans plusieurs domaines d'application. Nous présentons dans ce qui suit quelques exemples d'application.

– **Applications militaires**

Comme pour beaucoup d'autres technologies d'information les réseaux de capteurs sans fil sont considérés comme produit des recherches militaires. Les nœuds capteurs peuvent être déployés dans un champ de bataille, dans une zone de surveillance de l'ennemi ou plutôt pour détecter la présence d'armes nucléaire etc.

– **Applications environnementales**

Des micro-capteurs peuvent être dispersés dans la nature, zone à accès difficile afin de surveiller des catastrophes naturelles (inondations, séismes, etc.), détecter des incendies dans les forêts, surveiller la qualité d'air ou de l'eau [6], etc.

– **Applications médicales**

Le champ de contrôle de santé (Health monitoring) représente un grand marché pour les réseaux de capteurs sans fil qui a tendance à croître très rapidement [18]. Les capteurs peuvent être implantés dans le corps humain pour contrôler les problèmes médicaux (le cancer, fréquences cardiaques,...etc.), surveiller les patients et l'avancement de leurs états dans un hôpital [6].

– **Applications domestiques**

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance.

En plaçant, sur le plafond ou dans le mur, des capteurs, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes et seulement si c'est nécessaire [18].

– **Applications industrielles**

Dans l'industrie, des réseaux de capteurs peuvent être employés pour surveiller des processus de fabrication ou l'état des équipements industriels. L'intégration des micro-capteurs dans un processus de stockage et de livraison de marchandises peut être utilisée pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré [6] [18].

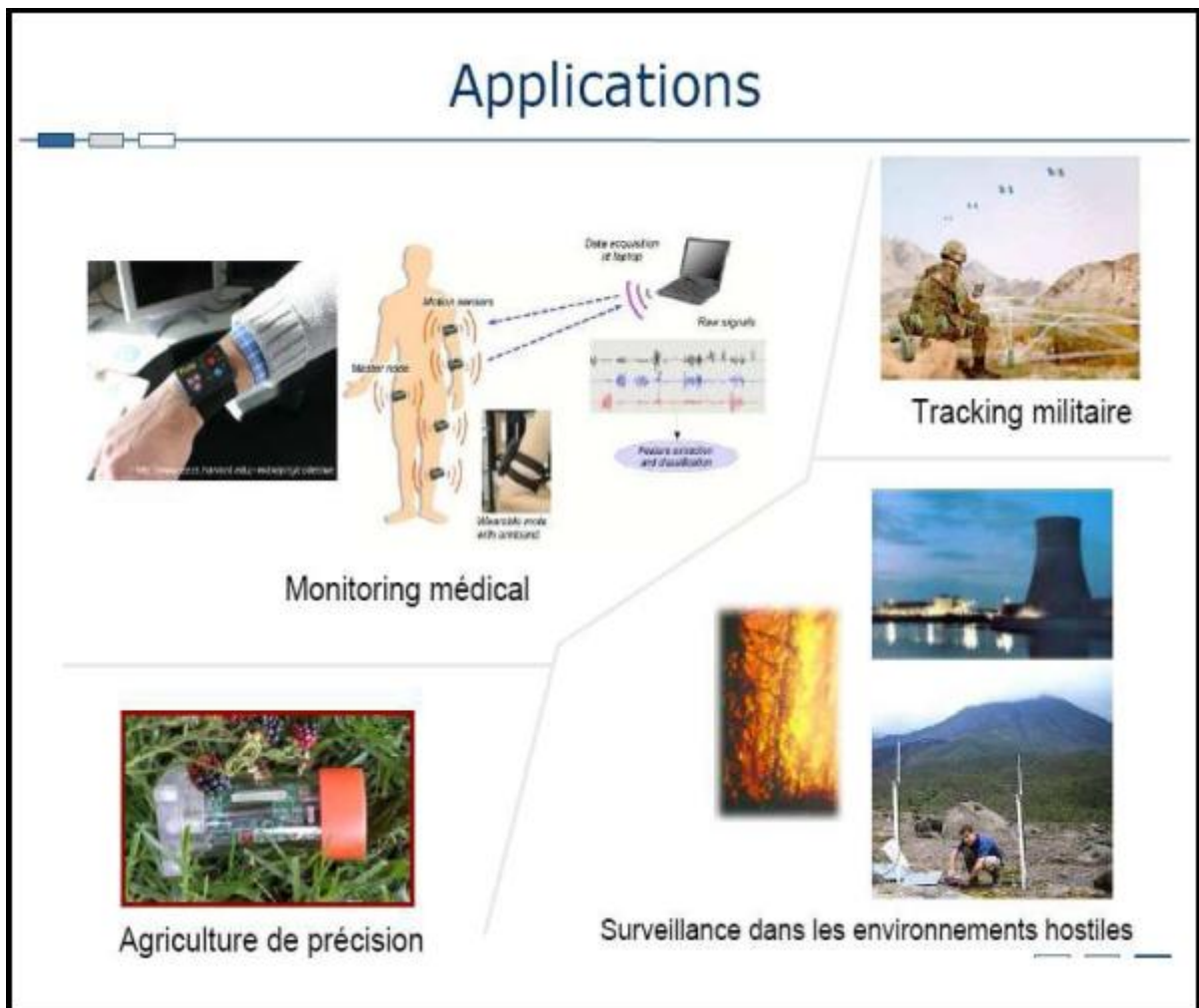


Figure I.7: Exemples d'applications des RCSFs

I.6 Facteurs et contraintes des RCSFs [2] [9]

La conception et la réalisation des réseaux de capteurs sans fil sont influencées par plusieurs paramètres. Ces facteurs servent de directives pour le développement des algorithmes et protocoles utilisés dans les RCSFs, et Ils sont aussi considérés comme des mesures de comparaison de performances entre les différents travaux dans le domaine.

– Topologie réseau

Plusieurs centaines de nœuds capteurs sont déployés avec une densité pouvant être supérieure à 20 nœuds par m³, ceci exige une bonne gestion de la maintenance de la topologie du réseau. Nous examinons, dans ce qui suit, les différents problèmes liés aux topologies des réseaux de capteurs et leurs changements [2].

• Phase de pré-déploiement et de déploiement

Les nœuds capteurs peuvent être lancés par catapulte (jet d'un avion, roquette ou missile) ou placés nœud par nœud d'une manière individuelle (manuellement ou à l'aide des robots) sur le champ de captage. Le nombre important de nœuds utilisés dans un réseau de capteurs empêche leur déploiement suivant un plan soigneusement établi, cependant un schéma général pour le déploiement initial doit être conçu pour permettre de :

- Réduire les coûts d'installation;
- Augmenter la flexibilité d'arrangement des nœuds;
- Faciliter l'auto-organisation des nœuds et leur tolérance aux pannes.

• Phase de post-déploiement

Après la phase de déploiement, la topologie du réseau est exposée fréquemment à des changements dues aux :

- Changement de positions des nœuds;
- Epuisement d'énergie;
- Dysfonctionnement des nœuds ou aux besoins pour leur application.

• Phase de redéploiement de nouveaux nœuds

Des nœuds capteurs additionnels peuvent être installés à tout moment pour remplacer ceux qui sont en panne ou bien répondre aux besoins des tâches assignées au réseau. Cette addition entraîne la réorganisation du réseau et le changement de sa topologie.

– Echelle (scalabilité)

Le nombre de nœuds capteurs déployés dans un champ de captage pour l'étude d'un phénomène est très important de l'ordre de centaines ou milliers, selon l'application il peut atteindre une valeur extrême des millions, ce qui peut garantir une couverture totale et assure une transmission plus fiable des données sur l'état du champ.

– Environnement

Les nœuds capteurs sont déployés de manière dense soit très proche d'un phénomène à observer, soit à l'intérieur du phénomène. Généralement, ils opèrent dans des endroits hostiles: au fond d'un océan, dans des champs de bataille au-delà des lignes ennemies, des champs biologiquement ou chimiquement souillés, etc. Donc, ces nœuds doivent être conçus d'une manière à pouvoir résister aux différentes et sévères conditions de l'environnement.

– Média de transmission

Dans un réseau de capteurs sans fil, les nœuds sont liés via un moyen de communication sans fils et ceci, en utilisant un support optique, ou des fréquences radio. Cependant, il faut s'assurer de la disponibilité du support de transmission choisi par l'environnement de captage afin de s'assurer de l'accomplissement des tâches effectuées par le réseau.

– Tolérance aux fautes

La défaillance ou le blocage des nœuds dans un réseau de capteurs peut être engendré par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique, ou les interférences liées à l'environnement. La propriété de tolérance aux pannes est définie par l'habilité du réseau à maintenir ses fonctionnalités sans interruptions provoquées en cas de défaillance d'un ou plusieurs de ses capteurs. Elle vise donc à minimiser l'influence de ces pannes sur le fonctionnement global du réseau.

– Consommation d'énergie

Comme les nœuds capteurs sont des composants micro-électroniques, ils ne peuvent être équipés que par des sources limitées d'énergie. De plus, dans certaines applications, ces nœuds ne peuvent pas être dotés de mécanismes de rechargement d'énergie, par conséquent, la durée de vie d'un nœud capteur dépend fortement de la durée de vie de la batterie associée. Le facteur de consommation d'énergie est d'une importance capitale dans les réseaux de capteurs.

La majorité des travaux de recherche menés actuellement se concentrent sur ce problème afin de concevoir des algorithmes et protocoles, spécifiques à ce genre de réseau, qui consomment le minimum d'énergie.

– **Agrégation de données**

Les nœuds dans un réseau de capteurs peuvent collecter les mêmes données. En effet, les paquets similaires peuvent être agrégés afin de réduire le nombre de transmissions. L'agrégation de données est la combinaison des données en provenance des différentes sources selon une certaine fonction d'agrégation (par exemple, minimum, maximum, ou moyenne). Cette technique a été employée pour réaliser l'optimisation de transfert et l'efficacité énergétique.

– **Coût de production**

Le coût de production d'un seul micro-capteur est très important pour l'évaluation du coût global du réseau, si ce dernier est supérieur à celui nécessaire pour le déploiement des capteurs classiques, l'utilisation de cette nouvelle technologie ne serait pas financièrement justifiée. Par conséquent, réduire le coût de production jusqu'à moins de 1\$ par nœud est un objectif important pour la faisabilité de la solution des réseaux de capteurs sans fil.

Conclusion

Les réseaux de capteurs sans fil suscitent un intérêt croissant vu la diversité de leurs applications: militaire, santé, environnement, etc. Dans ce premier chapitre, nous avons présenté l'état de l'art de ces réseaux en exposant les concepts généraux qui leur sont liés. Pour cela, nous avons commencé par une description matérielle et logicielle d'un capteur. Puis, nous avons procédé à l'étude des réseaux de capteurs, leurs architectures de communication, la pile protocolaire et leurs diverses applications. Plusieurs facteurs et contraintes embarrassent la gestion de ce type de réseau. En effet, ces réseaux se caractérisent par une puissance énergétique limitée ce qui rend difficile la tâche d'optimisation de la durée de vie du réseau.

CLUSTERING DANS LES RCSEs

Introduction

Un réseau de capteurs sans fil typique se compose d'un ensemble de nœuds déployés dans un champ, dotés de capacités de détection, traitement, stockage et communication, fonctionnant de manière coopérative.

En raison des contraintes liées aux réseaux de capteurs sans fil, il est nécessaire de concevoir des techniques permettant de les structurer de façon à minimiser efficacement l'utilisation des différentes ressources et par conséquent prolonger leur durée de vie.

Le clustering a été prouvé pour être une technique très économe telle qu'elle permet d'économiser la bande passante de transmission, d'exploiter de façon meilleure les ressources (radio, cpu) et diminuer la perte d'énergie totale du réseau. Donc, il permet de mieux gérer le trafic du réseau et d'alléger la quantité d'informations qui circule en effectuant des traitements au sein du cluster avant de propager les données vers le reste du réseau.

II.1 Notions de base et motivations:

L'intervention humaine est généralement impraticable dans les zones de déploiement des nœuds capteurs, cependant l'organisation manuelle du réseau en groupes est irréalisable. Pour cette raison, l'organisation des réseaux de capteurs en clusters a été largement explorée au cours de ces dernières années, conduisant à l'apparition d'un grand nombre de protocoles de clustering [20].

Le clustering [21] est un processus qui permet d'organiser virtuellement les nœuds d'un réseau en plusieurs groupes dits clusters. Chaque cluster est composé d'un nœud chef dit Cluster-Head qui le représente, ainsi que plusieurs nœuds appelés : nœuds membres.

Les nœuds membres d'un cluster transmettent leurs données au cluster-head approprié, qui est principalement, responsable de leur envoi à la station de base directement ou par l'intermédiaire d'autres cluster-heads ou d'autres nœuds.

II.1.1 Définitions:

Ci-dessous nous présenterons les principaux composants d'une architecture hiérarchique pour un réseau de capteur [22].

- **Nœud capteur** : est le composant principal d'un RCSF, tel qu'il puisse assurer une multiplicité de rôles : acquisition, stockage, routage ainsi que traitement des données dans une zone de surveillance.

- **Cluster** : un cluster est un groupe de nœuds capteurs qui forme l'unité organisationnelle du réseau. La nature dense de ces réseaux exige de les répartir en groupes afin de simplifier les tâches de communication et faire face aux différentes contraintes.
- **Cluster-Heads** : (leaders, chefs de groupes) sont les représentants des clusters ; ils sont souvent requis pour l'organisation des activités dans les clusters. La tâche d'un cluster-head ne se limite pas au cheminement des données, mais elle peut être accompagnée par un traitement, agrégation, fusion des données venant des autres capteurs afin de simplifier la gestion du réseau et en particulier optimiser l'exploitation des différentes ressources. Les nœuds peuvent être ordinaires ou dotés de plus d'énergie.
- **la station de base** : est au niveau supérieur de la hiérarchie d'un réseau de capteurs sans fil. Elle fournit la liaison de communication entre le réseau et l'utilisateur final.
- **Utilisateur final**: les données dans un réseau de capteurs peuvent être utilisées par une variété d'applications. Par conséquent, une application particulière peut utiliser les données du réseau via internet, un PDA ou même un ordinateur bureau.

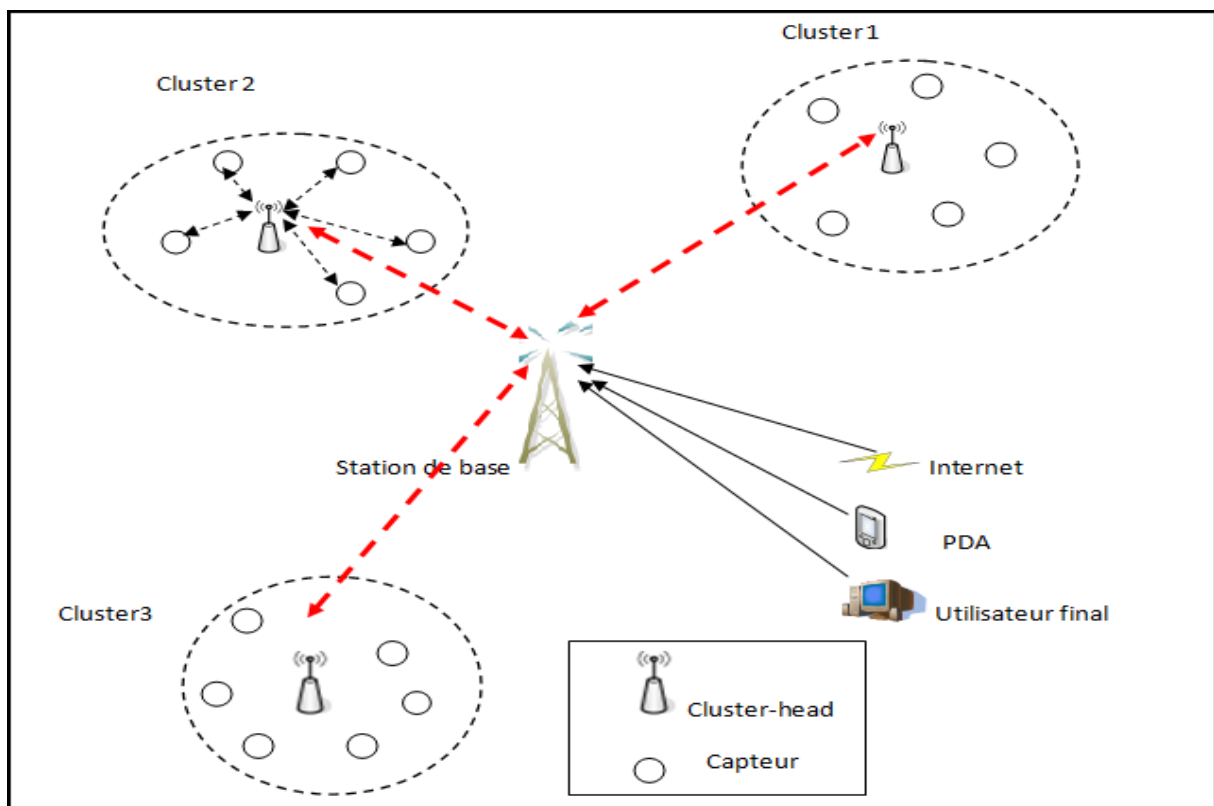


Figure II.1 Hiérarchie d'un RCSFs

II.1.2 Construction d'une topologie en clusters:

Une multiplicité de techniques de clustering a été présentée dans la littérature. Selon le mode de déploiement des nœuds capteurs (déterministe ou aléatoire), le processus d'élection des cluster-heads, le nombre et la taille des clusters formés ainsi que leur capacité de s'adapter aux changements de la topologie du réseau, on peut distinguer une variété d'algorithmes de clustering [23].

La figure ci-dessous illustre un exemple de réseau structuré en plusieurs clusters.

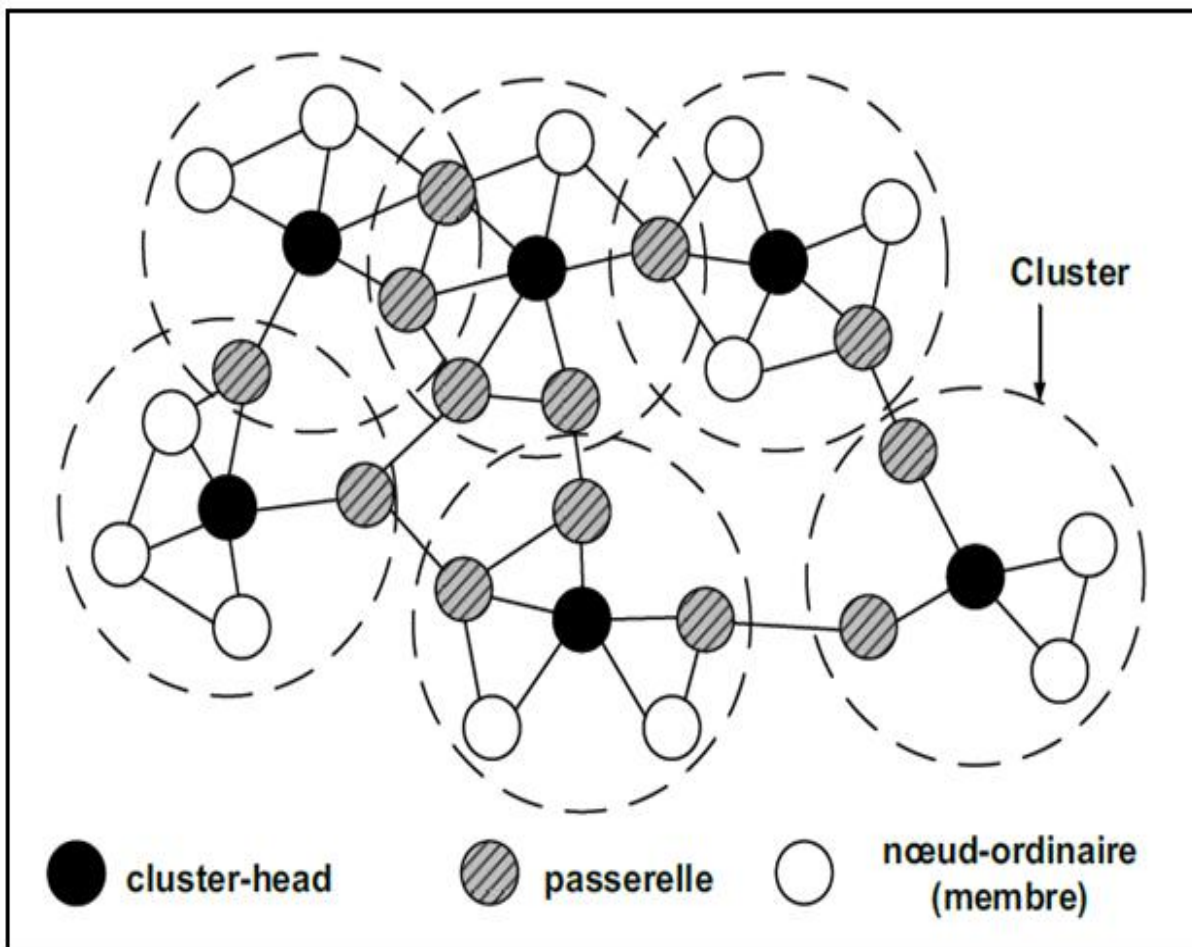


Figure II.2: Exemple de structure en clusters [25]

Plusieurs méthodes de formation de clusters existent ; cependant, un algorithme simple de clustering peut se décrire en 03 phases :

– **Election de cluster-head:**

Un nœud du réseau décide de devenir cluster-head à partir d'une métrique spécifique ou d'une combinaison de métriques tels que : le plus petit ou plus grand ID dans son voisinage, le degré de connectivité, l'énergie résiduelle ou selon un poids (weight) représentant une alliance de quelques métriques. Un nœud élu cluster-head, diffuse son statut à son voisinage afin d'inviter les nœuds qui lui sont adjacents à le rejoindre dans son cluster.

– **Communication intra-cluster et inter-cluster:**

Une fois les clusters sont formés, une phase de communication peut être lancée.

La communication intra-cluster c'est-à-dire entre le nœud cluster-head et les autres nœuds membres du même cluster peut se faire, soit directement en un seul saut, soit en multi-sauts [24].

Lors d'une communication directe (en un seul saut), chaque nœud du cluster transmet ses données directement à son propre cluster-head qui les acheminera vers la station de base. Cela admet que les nœuds membres sont capables d'atteindre le cluster-head et ce dernier, la station de base ; mais cela peut engendrer une surconsommation d'énergie si la distance entre le cluster-head et les nœuds est grande et de même avec le nœud puits.

La figure suivante illustre un exemple de communication à 1-saut CHs-station de base.

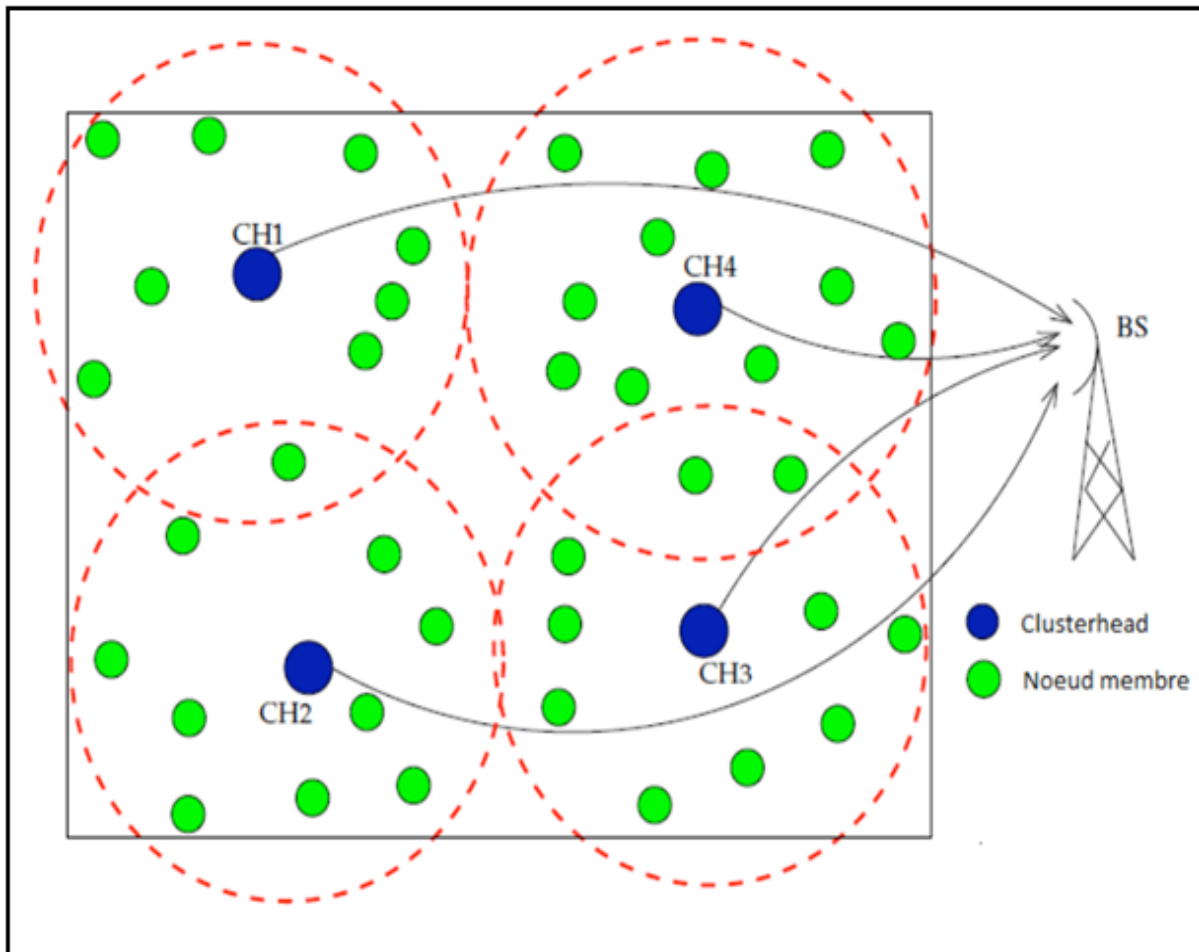


Figure II.3 Exemple d'une communication CH-BS à 1-saut [26]

Pour se faire à ce problème, une communication en multi-sauts de petites distances est mise en place. Dans ce cas, chaque membre du cluster envoie ses données au membre le plus proche de son cluster jusqu'à l'aboutissement du cluster-head.

Par la suite, les données se trouvant au niveau du chef de cluster doivent être communiquées à la station de base c'est-à-dire communication inter-cluster. Dans ce cas, les cluster-heads communiquent avec la station de base en multi-sauts à travers d'autres nœuds relais (nœuds Gateway) qui peuvent être des cluster-heads ou bien des nœuds membres d'un cluster (illustration: figure II.4). Mais cela n'empêche pas quelques limites, telle qu'une défaillance d'un nœud relai puisse engendrer une perte considérable de paquets d'où la nécessité de reconstruction de nouvelles routes et puis retransmission des données (si nécessaire).

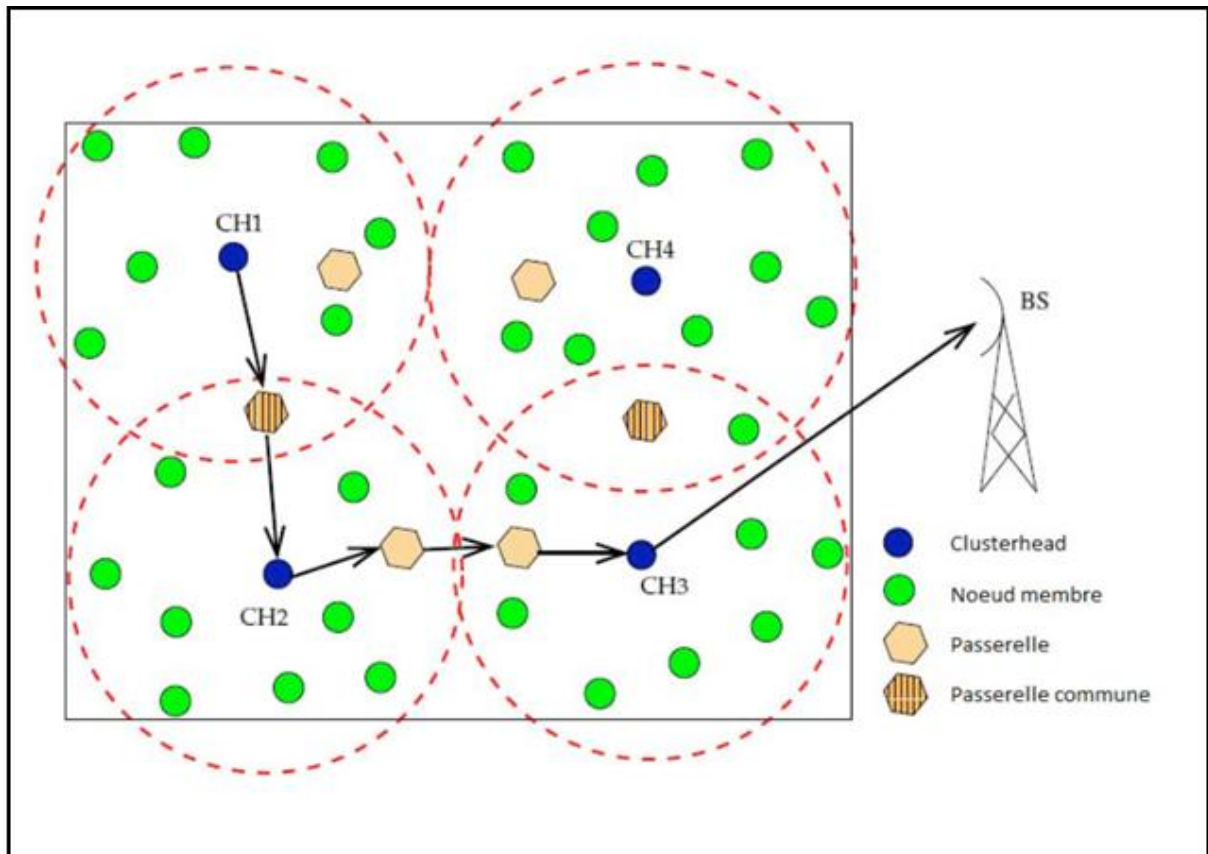


Figure II.4 Communication multi-sauts [26]

– **Maintenance des clusters :**

Un algorithme de clustering doit s'adapter aux changements de la topologie du réseau, donc une mise à jour périodique des clusters s'avère particulièrement nécessaire.

Cela prend en considération l'ajout ou la défaillance inattendue d'un nœud due à l'épuisement de sa batterie ou à des conditions environnementales.

Les clusters formés durant cette phase peuvent être recouvrant ou pas [25] c'est-à-dire interconnectés ou disjoints. Dans le cas recouvrant, un nœud peut appartenir à plus d'un cluster et, en général un rôle de passerelle est attribué à ces nœuds dans les communications entre clusters. Dans le cas contraire, chaque nœud du réseau n'est associé qu'à un seul cluster.

II.2 Avantages de l'approche de clustering:

Le mécanisme de clustering a de nombreux avantages [23] [27], il permet de :

- Réduire la taille des tables de routage stockées au niveau des nœuds, telle que la mise en place des routes soit faite au niveau des nœuds cluster-heads au lieu qu'elles soient au niveau de chaque nœud du réseau;
- Limiter la portée des interactions inter-clusters aux cluster-heads, éviter les échanges redondants des messages entre les nœuds, réduire les conflits de canaux ainsi que les collisions des paquets, et donc optimiser d'avantage l'utilisation de la bande passante;
- Stabiliser la topologie du réseau au niveau des nœuds et donc minimiser les frais de sa maintenance. Un nœud n'a qu'à effectuer son choix sur le cluster-head à joindre, ainsi il n'est pas affecté par les changements externes à son cluster;
- Mettre en place un processus d'agrégation sur les données transmises à l'utilisateur. L'ensemble des données interceptées par les nœuds d'un cluster peuvent être agrégées au niveau du cluster-head qui les communique par la suite à la station de base d'où réduction des paquets échangés;
- Implémenter des stratégies de gestions optimisées au niveau des cluster-heads afin de prolonger la durée de vie de la batterie de chaque nœud et donc la durée de vie du réseau. Par exemple, le cluster-head programme les activités dans le cluster de sorte que les nœuds puissent se mettre en veille le plus de temps possible et donc diminuer le taux de communication ainsi que le taux de collisions pour y accéder au canal de communication.

II.3 Quelques Contraints et facteurs de conception lors du processus de clustering :

L'approche de clustering ne se limite pas à l'organisation du réseau, mais peut considérablement affecter les performances de ce dernier. Il existe une multiplicité de contraintes et de facteurs à considérer lors du processus de clustering dans les réseaux capteurs sans fil [22] [27] :

– **Energie limitée et durée de vie du réseau:**

L'énergie est une ressource critique dans les réseaux capteurs sans fil et toute dépense de celle-ci se traduit par une durée de vie limitée du réseau.

Cette contrainte doit être tenue en compte par le processus de clustering de telle sorte à ce que la structuration du réseau en clusters réduise au maximum l'énergie consommée par les nœuds et donc prolonger sa durée de vie globale.

– **Capacités limitées:**

La petite taille physique ainsi que la petite quantité d'énergie stockée dans un nœud capteur limite beaucoup ses capacités en termes de traitements et communications.

Un bon algorithme de clustering devrait se servir des ressources du réseau, tout en tenant compte de la limitation de leurs capacités.

– **Domaines d'application et formation des clusters:**

Le type d'application s'appuie fortement sur l'organisation du réseau en clusters, étant donné un bon algorithme de clustering doit être capable de s'adapter aux divers besoins de l'application. Selon l'application, les concepteurs doivent examiner soigneusement la formation des clusters dans le réseau, telles que certaines exigences pour le nombre de nœuds d'un cluster ou le choix du cluster-head dans ce réseau peuvent avoir un impact sur son fonctionnement.

– **Synchronisation:**

Une des principales contraintes des réseaux de capteurs sans fil est la ressource énergétique des nœuds. Des systèmes de transmission basés slot (comme TDMA) permettent de fixer régulièrement des intervalles de repos (mise en veille) aux nœuds afin de minimiser l'énergie dépensée ; Mais cela nécessite la mise en place d'un concept important: la synchronisation. Lorsque l'on considère un système de clustering, la synchronisation a un effet remarquable sur la performance globale du réseau.

– **Agrégation de données:**

La densité est l'une des caractéristiques majeure des réseaux de capteurs sans fil. Cependant, il existe souvent des informations similaires interceptées par plusieurs nœuds capteurs dans le réseau. L'agrégation de données permet de combiner les mêmes données en une information significative, d'où l'élimination des redondances.

Entre autre l'intégration d'un mécanisme d'agrégation aux algorithmes de clustering permettra une réduction considérable de la quantité des données échangées dans le réseau, minimisera encore d'avantage les communications entre les nœuds et allégera en conséquence la congestion du réseau.

II.4 Classification des paramètres liés au processus de clustering:

Cette section fera l'objet d'énumération de l'ensemble des paramètres qui peuvent être utilisés afin de différencier et classifier les différents protocoles de clustering dédiés aux réseaux de capteurs sans fil.

La figure ci-après représente une classification des différents paramètres liés au processus de clustering dans les RCSFs [23] [29].

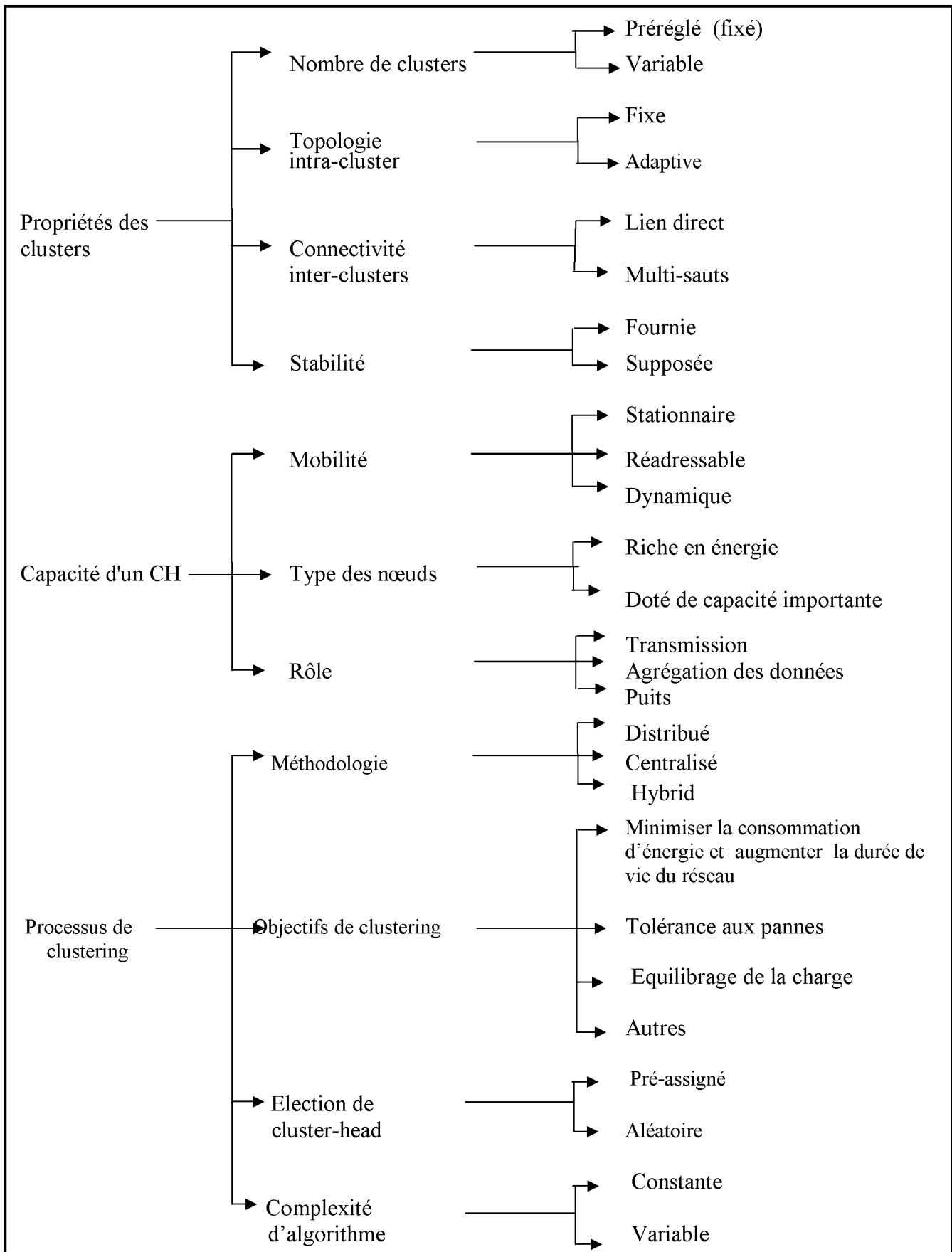


Figure II.5: Classification des paramètres liés au processus de clustering

II.4.1 Propriétés des clusters:

- **Nombre et taille des clusters:**

Le nombre et la taille des clusters sont des mesures importantes dans l'identification de la performance d'un algorithme donné.

Dans certaines approches, le nombre de cluster-heads est prédéterminé et donc le nombre de clusters dans le réseau est prédéfini. Toutefois, la sélection des cluster-heads se fait de manière aléatoire, ce qui en résulte un nombre variable de clusters.

- **Topologie intra-cluster:**

Dans certaines approches de clustering, une communication directe capteur-CH est adaptée. Cependant, des communications multi-sauts sont parfois nécessaires, surtout lorsque la portée de consommation d'un capteur est limitée. Ainsi, la topologie d'un cluster varie par l'ajout ou l'élimination des liaisons vers des nœuds voisins.

- **Connectivité inter cluster-heads:**

Selon les distances séparant les cluster-heads de la station de base ainsi que leurs capacités de communication, un chemin CH-station de base est établi. Dans certains cas, on suppose qu'un cluster-head est capable d'atteindre directement la station de base; cependant, d'autres travaux considèrent que la connexion doit être assurée par l'établissement d'une route inter-cluster-heads jusqu'à l'atteinte de la station de base.

- **Stabilité:**

Lorsque la structure de clustering subit des changements, variation du nombre de clusters et/ou évolution de l'adhésion des nœuds aux clusters, un compromis entre ces changements et la stabilité du cluster doit être établi afin de conserver la structure aussi stable que possible. Autrement, elle est supposée fixe telle que chaque nœud soit propre à son cluster ainsi que le nombre de clusters dans le réseau soit fixe tout au long de sa durée de vie.

II.4.2 Capacité d'un cluster-head:

Les attributs d'un nœud cluster-head ci-dessous représentent des facteurs permettant de différencier les différentes approches de clustering:

- **Mobilité:**

La mobilité d'un nœud cluster-head influe sur les nœuds membres, donc une phase de maintenance régulière est nécessaire. Tandis qu'un cluster-head stationnaire ait pour objectif de former un cluster stable et donc faciliter la gestion du réseau.

Mais, parfois un cluster-head se repositionne sur des distances limitées afin de garantir une meilleure performance du réseau.

- **Type de nœuds:**

Dans un réseau de capteurs sans fil, les nœuds sont généralement déployés de manière aléatoire. Dans certains cas, parmi ces nœuds, un sous ensemble ayant une plus grande capacité énergétique est désigné comme cluster-heads. Tandis que dans d'autres cas, les cluster-heads sont des capteurs dotés de plus de moyens de calcul et de communication.

- **Rôle:**

Un nœud cluster-head peut servir de relais pour le trafic généré dans le réseau ou exécuter une approche d'agrégation ou fusion des différentes données collectées par les autres nœuds. Quelque fois, un cluster-head peut fonctionner en tant qu'une station de base et agir sur les phénomènes détectés.

II.4.3 Processus de clustering:

Les processus de clustering varient selon la méthodologie adaptée, l'objectif à atteindre ainsi que la manière dont les cluster-heads sont sélectionnés, ceux qui permettront par la suite de classifier quelques approches de clustering proposées dans la littérature.

- **Méthodologie:**

L'algorithme de clustering peut s'exécuter au niveau de chaque nœud du réseau, et chacun d'eux décide indépendamment des autres de son rôle puis le fait connaître par l'envoi des messages. Autrement, c'est la station de base qui se charge de l'affectation des rôles et lance le processus de formation des clusters et garde une vue globale du réseau.

Parfois, des techniques hybrides sont utilisées dans le cas des cluster-heads riches en ressources de telle manière à ce que chaque cluster-head lance la formation de son propre cluster, et la coopération inter cluster-heads soit assurée de manière distribuée.

– **Objectifs du clustering:**

Les protocoles de clustering varient selon les objectifs à atteindre afin de répondre aux exigences des applications, parmi eux :

- Minimisation de la consommation énergétique qui est d'une importance capitale dans les réseaux de capteurs et donc étalement de leurs durées de fonctionnement.
- Tolérance aux pannes des nœuds cluster-heads, dues aux dommages physiques, est nécessaire afin d'éviter la perte des paquets interceptés par les membres.
- Equilibrage de la charge (load balancing) est l'une des solutions permettant d'améliorer l'efficacité des applications et la durée de vie du réseau. Il vise essentiellement à répartir équitablement les nœuds sur les clusters telle que la mise en clusters à taille égale empêchera l'épuisement de l'énergie d'un sous ensemble de cluster-heads à taux élevé de nœuds membres, ou de les rendre prématurément dysfonctionnels.

– **Election de cluster-head:**

Un nœud cluster-head peut être pré-assigné ou sélectionné de façon aléatoire parmi l'ensemble des nœuds de son réseau. En général, un nœud est élu cluster-head sur la base de plusieurs métriques : son énergie résiduelle, le nombre de voisins, sa mobilité tel qu'un nœud de plus faible mobilité. Par ailleurs, un poids constituant une combinaison de métriques attribué à chacun des nœuds du réseau peut être utilisé pour l'élection du cluster-head.

– **Complexité de l'algorithme:**

Selon l'objectif et la méthodologie, une multiplicité d'algorithmes de clustering a été proposée. Le taux de complexité peut être constant ou variable en fonction du nombre de clusters ou cluster-heads.

II.5 Quelques approches de clusterisation:

Dans cette section du chapitre, nous présentons quelques algorithmes de clusterisation, proposés et étudiés dans la littérature, classifiés selon des critères et métriques de décision utilisés dans le processus de clustering [23] [28].

II.5.1 Les protocoles hiérarchiques:

Dans ce type d'approches, la construction des clusters et l'élection des cluster-heads dans le réseau se basent sur une formule de probabilité tenant compte du nombre de nœuds le composant, la taille des clusters, le nombre de fois qu'un nœud ait été élu cluster-head, ainsi que l'énergie résiduelle des capteurs.

– LEACH: Low Energy Adaptive Clustering Hierarchical

LEACH est le premier algorithme de clustering proposé pour les RCSFs, il s'applique sur les réseaux homogènes [28]. Sa technique consiste en la division du réseau en deux classes: les cluster-heads (CHs) choisis d'une manière aléatoire et les nœuds membres.

L'algorithme se déroule en plusieurs rounds et pour chaque round, une rotation du rôle de CH est initiée afin d'assurer une répartition efficace de la consommation énergétique ainsi que l'utilisation de la bande passante entre les nœuds du réseau.

Afin de réduire la quantité d'informations communiquées à la station de base, les cluster-heads exécutent le processus d'agrégation sur les données interceptées par les membres appartenant à leurs propres clusters, et puis transmettent le paquet résultant à la station de base.

LEACH s'exécute en deux phases principales: la phase "setup" et la phase "steady state" [24] [28]. La phase "setup" est la phase de construction durant laquelle le réseau s'organise en clusters et les cluster-heads sont sélectionnés.

Durant cette phase, le nœud puits ou l'administrateur du réseau définit une fraction f des nœuds désirant devenir cluster-heads telle que f est choisie selon la densité du réseau [31].

Chaque nœud n du réseau choisit aléatoirement un nombre "nb" compris entre 0 et 1, puis compare ce nombre à un certain seuil $T(n)$. Si nb est inférieur à $T(n)$, alors le nœud n est élu cluster-head.

$T(n)$ est défini comme suit [28]:

$$T(n) = \begin{cases} \frac{f}{1-f(r \bmod (\frac{1}{f}))} & \text{si } n \in G \\ 0 & \text{sinon} \end{cases}$$

Avec:

f: fraction des nœuds désirant devenir CHs;

r: numéro du round courant;

G: ensemble des nœuds n'ayant pas été élus CHs durant les 1/f dernières périodes précédentes.

Par la suite, chaque nœud élu cluster-head émet un message de notification aux autres nœuds du réseau. Après réception des messages de notification, les nœuds non cluster-heads décident de leurs appartenances à un cluster selon l'amplitude du signal reçu tel que le CH ayant un signal le plus fort est choisi (i.e. le plus proche de son voisinage). Ainsi, chaque nœud informe son chef de son adhésion.

Après l'élection des CHs, une phase de communication appelée phase "steady state" est lancée (à voir dans III.6.1).

– **HEED: Hybrid Energy Efficient Distributed clustering approach**

HEED est un protocole de clustering itératif [30] qui vise la formation efficace des clusters afin de maximiser la durée de vie du réseau. Il adopte un critère hybrid pour la construction des clusters et la sélection des cluster-heads en se basant sur l'énergie résiduelle d'un nœud ainsi qu'un second paramètre qui dépend de la proximité du nœud à ses voisins ou de son degré.

Dans le processus de clusterisation utilisé par HEED, chaque nœud prend sa décision selon des informations locales. Par ailleurs, les deux paramètres mentionnés précédemment sont utilisés pour l'élection des cluster-heads.

Au départ, seuls les nœuds ayant une énergie résiduelle plus importante peuvent avoir une grande probabilité (CH_{prob}) de s'élire cluster-head, tel que:

$$CH_{prob} = C \times \frac{E_{residual}}{E_{max}} \quad [24]$$

avec: C détermine le nombre optimal de CHs, spécifié par l'utilisateur;

$E_{residual}$ est l'énergie résiduelle estimée du nœud

E_{max} est l'énergie initiale correspondante à une batterie entièrement chargée.

Par la suite, chaque nœud élu CH diffuse dans son voisinage son nouveau statut de cluster-head. Ainsi, les nœuds qui sont à l'écoute choisissent le CH à joindre selon le coût.

Pour cela, une métrique de coût est définie dans le processus d'élection des CHs.

$$AMRP(u) = \frac{\sum_{i=1}^M \text{Min Pwr}(i)}{M} \quad [30]$$

AMRP d'un nœud u est définie par la puissance moyenne nécessaire pour qu'un nœud u soit atteint par M nœuds dans son voisinage.

A chaque itération, un nœud qui n'est couvert par aucun CH peut s'auto-élire CH suivant sa probabilité CH_{prob} .

II.5.2 Les protocoles pondérés:

Dans cette catégorie d'algorithmes, une combinaison de plusieurs métriques est utilisée pour élire les cluster-heads. La fonction de combinaison représente le poids d'un nœud.

Ces algorithmes visent à élire un ensemble de cluster-heads qui sera le plus adapté à une topologie dans le but de satisfaire un ou plusieurs besoins.

Par exemple, dans un réseau de capteurs où l'énergie est un facteur important, le paramètre d'énergie résiduelle peut obtenir un poids plus élevé dans la somme pondérée de la métrique résultante [25].

– WCA: Weighted Clustering Algorithm [31]

Cet algorithme considère quatre métriques dans le calcul du poids d'un nœud: le degré D , la somme des distances avec ses voisins P , la mobilité relative moyenne M et le temps de service au tant que cluster-head T , tel que:

$$\text{Poids} = \alpha D + \beta P + \gamma M + \delta T \quad \text{avec: } \alpha + \beta + \gamma + \delta = 1$$

- Le degré $D(u)$ est la différence entre le degré du nœud u et une valeur m qui représente le nombre de nœuds qu'un cluster-head peut servir. Les auteurs n'explicitent pas le moyen de déterminer la valeur de m .
- Les distances $P(u)$ entre les nœuds sont calculées à l'aide d'un GPS.
- La mobilité relative $M(u)$ d'un nœud représente le rapport des niveaux de puissance des transmissions successives reçues par un nœud de ses voisins.

Le nœud possédant le plus petit poids dans son voisinage devient CH. Cette élection est garantie par un processus itératif où chacun des nœuds du réseau diffuse son poids et maintient une liste des poids émanant de ses voisins.

Les informations sauvegardées seront à nouveau échangées avec les voisins directs, ainsi ce processus se poursuit jusqu'à ce que tous les nœuds aient des informations sur les poids des autres nœuds du réseau, ce qui permet d'identifier les nœuds ayant le plus petit poids. Une fois le CH est défini, ses voisins s'y rattachent. Ce processus se répète jusqu'à ce que tous les nœuds soient affectés aux clusters.

– **DCA: Distributed Clustering Algorithm [28] [32]**

Ce protocole est destiné aux réseaux dans lesquels les nœuds sont immobiles ou se déplacent avec une faible vitesse. Il considère que chaque nœud du réseau possède un poids unique et les CHs sont choisis à la base des poids assignés aux nœuds. Ainsi, un nœud est sélectionné pour être CH s'il possède le plus grand poids dans son voisinage. Autrement, il joint le CH qui se trouve dans son voisinage. Cet algorithme opère selon deux phases: formation des clusters et maintenance de ceux générés.

Durant la phase de formation des clusters, le protocole suppose que les informations de voisinage à un seul saut sont disponibles. Les nœuds ayant le poids le plus élevé dans leurs voisinages sont choisis pour être CHs et diffusent par la suite cette information à leurs voisins. Une fois qu'un nœud ait reçu les messages d'annonces des autres nœuds de poids plus élevés, s'y rattache au CH ayant le plus grand poids parmi ceux reçus et confirme son adhésion par l'envoi d'un message de rattachement. Par ailleurs, un nœud n'ayant reçu aucun message d'annonce de ses voisins prendra la décision de jouer le rôle de CH.

Sachant que le réseau dépend de plusieurs contraintes de son environnement (mobilité ou dysfonctionnement des nœuds), une phase de maintenance, durant laquelle la structure des clusters est reconfigurée et le processus de clustering est relancé, s'avère nécessaire.

II.5.3 Les protocoles heuristiques:

Une heuristique est toute approche pouvant être mise en œuvre afin d'obtenir une solution réalisable proche de l'optimum en un temps de calcul raisonnable pour un problème donné [33].

Plusieurs protocoles de clustering proposés dans la littérature sont basés sur les heuristiques. La plupart d'entre eux ont une complexité temporelle $O(n)$ où n représente le nombre total de nœuds. Parmi ces algorithmes, nous citons quelques exemples dans ce qui suit.

– **HCC: Highest Connectivity Cluster Algorithm [22]**

Dans ce type d'algorithme, la connectivité (degré) d'un nœud est considérée. L'idée est que les nœuds de fort degré sont favorisés pour être cluster-heads, car ils sont connectés à un grand nombre de nœuds, ce qui en résulte un nombre réduit de clusters générés. En cas de conflit, c'est-à-dire, deux voisins ont le même degré, c'est celui de plus faible ID qui est élu.

Par ailleurs, avec ce protocole, un nœud déjà élu cluster-head a tendance de garder son état de CH plus longtemps, car il a une forte chance de se réélire plus souvent cluster-head par rapport à d'autres nœuds.

– **Max-Min-D cluster [34]:**

Cette approche est considérée comme une nouvelle procédure pour l'élection de cluster-heads, développée pour étendre la notion de clusters à 1-saut et donc construire des D-clusters non recouvrants (cluster à D-sauts), avec: D un paramètre de l'heuristique dont le nombre de clusters formés dépend. Les règles heuristiques utilisées par ce protocole permettent une convergence rapide vers la solution.

Cette approche utilise la métrique identifiant ID du nœud pour l'élection du cluster-head, et se déroule en trois rounds.

Lors du premier round, chaque nœud diffuse son ID à ses voisins à D-sauts et collecte leurs IDs. Il en garde le plus grand des IDs qu'il diffusera à nouveau dans le second round. Au cours du deuxième round, chaque nœud garde le plus petit des IDs parmi ceux qu'il a reçu durant cette phase i.e. le plus petit des IDs parmi les plus grands. Le troisième round consiste à choisir les cluster-heads sur la base des IDs enregistrés lors des deux premières phases.

Un nœud se déclare CH s'il a reçu son ID durant le second round de diffusion. Sinon, si ce nœud a reçu un ID dans chacun des deux rounds 1 et 2, il choisira le nœud ayant cet ID comme cluster-head. Sinon, il choisira le nœud possédant le plus grand ID dans son voisinage à D-sauts comme CH.

Cet algorithme permet de produire une structure de clusters robuste. Cependant la durée de déroulement de l'algorithme n'est pas négligeable. Mais l'utilisation d'un nombre important de messages pour l'élection de cluster-heads augmente considérablement l'overhead du réseau. De même, ne tient pas compte de la mobilité des nœuds ainsi que les changements de la topologie.

Conclusion

Dans ce chapitre, nous avons établi un état de l'art sur une technique de structuration hiérarchique du réseau, le clustering.

Après une brève présentation des différents éléments constituant une structure hiérarchique, nous avons explicité le processus de construction des différents clusters ainsi que les contraintes et les facteurs à considérer lors de ce processus. Puis, nous avons énuméré quelques points positifs garantis par cette approche. Et pour finir nous avons donné un aperçu sur quelques algorithmes de clustering, les plus étudiés dans la littérature.

Après que le réseau soit structuré en clusters, d'autres approches de traitements sur les données collectées, peuvent être appliquées; pour cela le chapitre suivant fera l'objet d'étude d'une de ces techniques, l'agrégation de données.

**AGREGATION DE
DONNEES DANS LES
RCSEs**

Introduction

Un réseau de capteurs sans fil est une collection de nœuds capables d'intercepter des données de leur environnement, les traiter, les communiquer et les acheminer à la station de base. Ces nœuds capteurs sont déployés de manière dense et souvent dans des environnements hostiles avec un accès limité. Donc, il est impossible de les ajuster dans leur champ.

Par conséquent, plusieurs nœuds peuvent intercepter des données fortement corrélées ou les mêmes données qui doivent être traitées et acheminées à la station de base tout en gardant le réseau fonctionnel le plus longtemps possible. Cependant, garantir une telle caractéristique n'est une tâche facile surtout que ces nœuds sont à base de dispositifs électroniques limités en capacités matérielles (énergie, mémoire, bande passante).

Basé sur le principe que la station de base n'a pas obligatoirement besoin de toutes les données collectées par chacun des nœuds capteurs du réseau en raison de leur redondance, mais uniquement d'un résumé ou d'un agrégat de données effectué au niveau d'un nœud dit agrégateur.

L'agrégation de données est une approche intéressante qui peut considérablement aider à la conservation de l'énergie tout en éliminant les données redondantes ainsi, prolonger la durée de vie du réseau et maintenir son fonctionnement.

III.1 Motivation:

Un réseau de capteurs sans fil se compose d'un grand nombre de nœuds contraints en ressources, en termes de puissance de traitements, capacité de stockage, et bande passante de communication, dues à leur taille minuscule ainsi que l'énergie limitée [46].

En effet, les redondances qui surviennent dans le réseau pourraient avoir un impact négatif sur ces ressources: gaspillage d'énergie et de la bande passante, par exemple. Donc, il est impératif de développer une technique permettant la réduction de ces redondances: l'agrégation de données. Pour surmonter ce problème, il est essentiel de connaître ces contraintes et comment il introduit cette technique d'agrégation dans les RCSFs.

Les principales contraintes sont énumérées ci-dessous:

– **Energie:**

L'énergie est la contrainte majeure pour les réseaux de capteurs sans fil, et en raison de la nature de l'environnement où ces capteurs sont déployés, recharger ou remplacer leurs batteries n'est pas toujours possible. Dans ce type de réseaux, la consommation d'énergie est considérée par trois niveaux [47]: niveau de captage, de calcul et de communication.

Le niveau calcul représente le niveau microprocesseur où un nœud traite une diversité de données et exécute des processus opérationnels sur ces dernières, nécessitant beaucoup d'énergie. Sur le niveau communication, le nœud capteur dépense une quantité importante de son énergie à écouter, transmettre et recevoir des données.

– **Mémoire:**

Mémoire et espace de stockage est une autre contrainte dans les RCSFs, due à la petite taille du capteur. La mémoire du nœud est constituée [47] à la fois d'une mémoire flash (ROM) comportant le code du programme et le système d'exploitation, et d'une mémoire RAM permettant de stocker les données recueillies avant de les envoyer ailleurs; ainsi que les paquets émanant des autres nœuds. Donc, il n'est pas toujours possible d'emmagasiner un tas de données et de paquets. D'où la nécessité de la réduction du trafic échangé dans le réseau.

– **Bande passante de communication:**

Les réseaux de capteur sans fil disposent d'une bande passante de communication limitée. En effet, l'échange de données similaires ou fortement corrélées pourrait entraîner une réduction remarquable de sa largeur.

La limite de la bande passante peut conduire à des problèmes de congestion dans le réseau, augmentation du retard, pertes de paquets, donc nécessité de leurs retransmissions, et finalement perte considérable de la quantité d'énergie [48].

Afin de remédier à ces contraintes, des points d'agrégation doivent être introduits dans le réseau. Donc, réduction de la totalité des messages échangés dans le réseau, conservation d'énergie et puis augmentation de sa durée de fonctionnement.

III.2 Fusion de données et Agrégation de données [45]:

De nombreuses définitions ont été attribuées à la fusion de données, dite aussi fusion d'information, au fil du temps. Selon Hall et Llinas, la fusion de données est la combinaison des données provenant de capteurs multiples et d'autres informations pertinentes fournies par des bases de données associées afin d'aboutir à une meilleure exactitude et des inférences plus spécifiques que celles qui pourraient être atteintes par l'utilisation d'un seul capteur. Dans ce cas, la fusion est réalisée avec l'objectif d'améliorer la précision.

En particulier, pour les réseaux de capteurs sans fil, les données peuvent être fusionnées avec au moins deux objectifs: amélioration de la précision et économie d'énergie.

Par la suite, le terme agrégation de données est devenu populaire dans la communauté de réseaux de capteurs sans fil comme synonyme de fusion de donnée, mais cela doit être évité car il ne se réfère qu'à une instance de la fusion: le résumé.

Selon Van Renesse, l'agrégation est la capacité de résumer. Ce qui signifie que le volume de données manipulées est réduit à l'aide des fonctions de synthèse, tels que le maximum et la moyenne.

III.3 Agrégation de données:

III.3.1 Définition:

L'agrégation de données est définie comme un processus qui combine les informations interceptées par les différents nœuds capteurs du réseau tout en éliminant les redondances, réduisant au minimum le nombre de transmissions; puis fournit le résultat à la station de base directement ou à l'aide des nœuds intermédiaires, et donc conservation de l'énergie et augmentation de la durée de fonctionnement du réseau. [35][37]

III.3.2 Principe:

L'agrégation de données dans les réseaux de capteurs sans fil consiste à remplacer les lectures individuelles de chaque capteur par une vue globale, collaborative sur une zone donnée. [30]

Une multiplicité de fonctions d'agrégation peuvent être utilisées telles que: SOMME, MOYENNE, MIN, MAX afin d'obtenir un seul paquet à transmettre à la station de base, résumant une série de n messages et donc alléger le trafic dans le réseau et conserver d'avantage de l'énergie.

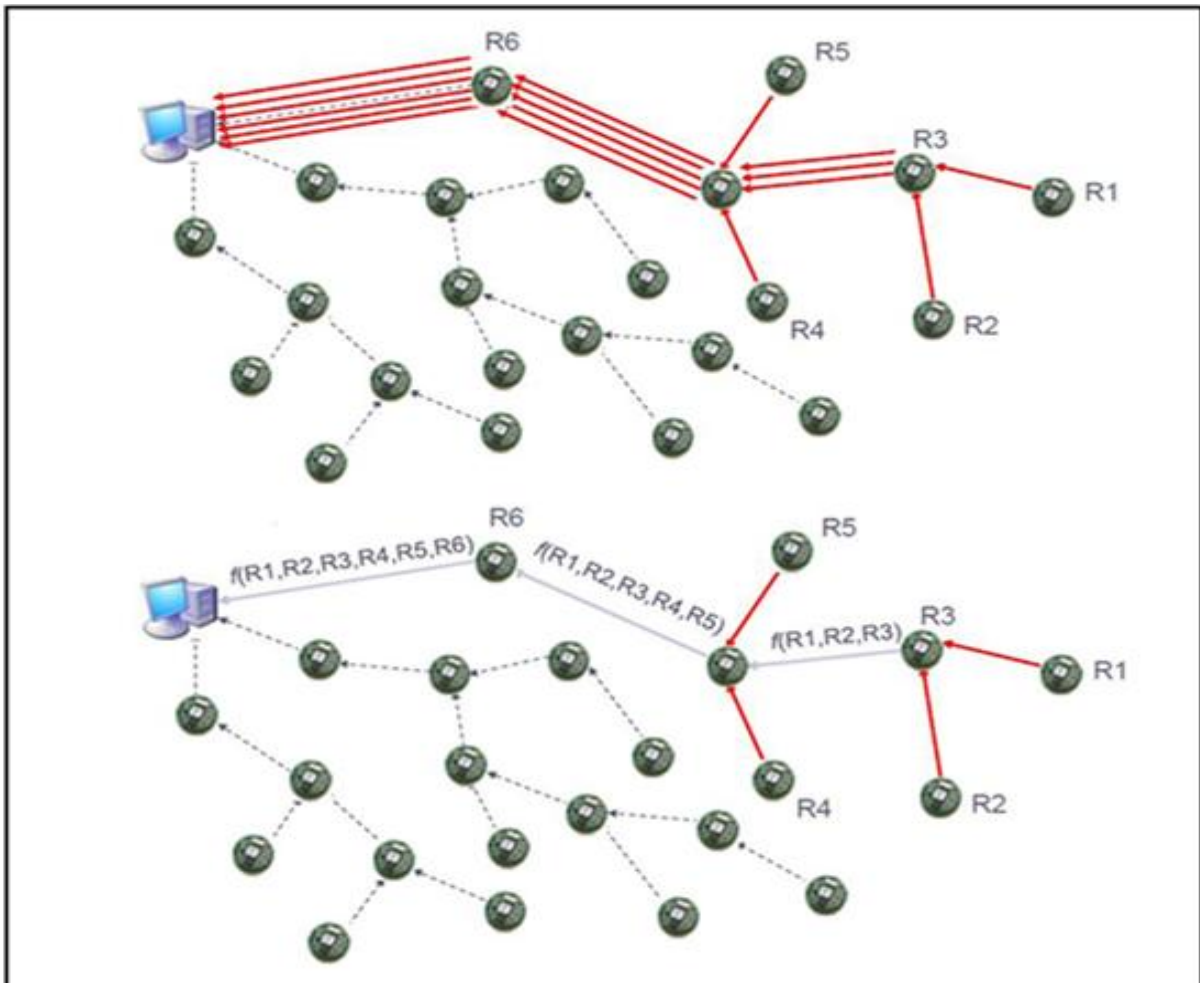


Figure III.1: Exemple d'agrégation de données [30]

La figure précédente illustre un exemple d'agrégation de données dans un réseau de capteurs sans fil. Au total, 18 messages sont envoyés sur le réseau.

Après l'utilisation du mécanisme d'agrégation de données, on obtient un total de 7 messages uniquement.

III.4 Approches d'agrégation:

On distingue deux approches d'agrégation: [37]

- avec réduction de la taille (with size reduction)
- sans réduction de la taille (without size reduction)

III.4.1 Agrégation avec réduction de la taille:

Cette approche fait allusion au processus de combinaison et de compression des données émanant du voisinage afin de réduire la longueur d'un paquet de données à transmettre à la station de base.

Considérons comme exemple, un nœud recevant deux paquets contenant les températures localement mesurées. Dans ce cas, le nœud effectue la moyenne des deux informations et n'envoie qu'un seul paquet à la station de base.

Cette approche a l'avantage de réduire de manière considérable la quantité des données communiquées dans le réseau et donc réduction de la consommation énergétique. En contre partie, après l'opération d'agrégation, la valeur de la donnée peut perdre sa précision et donc il n'est pas toujours possible de reconstituer le paquet original.

III.4.2 Agrégation sans réduction de la taille:

Cette approche correspond au processus de fusion des paquets de données venant des différentes sources dans un seul paquet sans aucun traitement sur les données.

Considérons le cas où deux paquets peuvent contenir des données décrivant deux phénomènes physiques différents: la température et l'humidité. Ces deux grandeurs ne peuvent pas être traitées ensemble mais elles peuvent être fusionnées et transmises dans un même paquet.

Bien que cette approche préserve les valeurs des données mais transmet beaucoup plus de bits sur le réseau à cause de la longueur des paquets à communiquer. Néanmoins, elle n'empêche pas la réduction du coût de transmission et donc préserve la ressource énergétique.

Le choix d'utilisation de ces deux approches dépend de plusieurs facteurs, parmi eux, le type et les besoins de l'application, ainsi que les caractéristiques du réseau.

III.5 Types d'agrégation de données dans les RCSFs:

Les techniques d'agrégation de données peuvent être classifiées selon trois critères: [35] [38]

- Agrégation basée sur le flux dans réseau
- Agrégation basée sur la qualité de service (QoS)
- Agrégation basée sur la topologie du réseau

III.5.1 Agrégation basée sur le flux dans le réseau

Dans le but de minimiser la consommation d'énergie dans un réseau de capteurs sans fil, on cherche à optimiser le flux de données qui circule entre les nœuds de ce réseau. Dans un problème de flux, on part de la vision du réseau sous forme de graphe où chaque nœud du graphe représente un nœud du réseau et les arcs représentent les liens directs ayant une capacité et une unité de coût.

Dans ce cas, on considère que l'agrégation appliquée dans les protocoles est modelée comme problème de flux dans le réseau [38]. Le but principal de ces protocoles est l'optimisation de la durée de vie du réseau soumise à des contraintes d'énergie des nœuds ainsi que le flux d'information circulant dans le réseau, en appliquant des algorithmes d'agrégation efficaces et aussi des algorithmes de routage minimisant la consommation énergétique de chaque paquet communiqué dans le réseau en choisissant le plus court chemin en terme du coût total d'énergie.

III.5.2 Agrégation basée sur la qualité de service (QoS)

La plupart des protocoles d'agrégation conçus ont pour objectif principal de garantir une efficacité énergétique et donc aboutir à des réseaux de capteurs avec une longue durée de vie.

Cependant, certaines applications exigent principalement une qualité de service en termes de mesures telles que: la bande passante, le délai de bout-en-bout et le débit de l'information. L'agrégation de données basée sur la qualité de service peut être classée en deux catégories [38]:

- Le premier cas, QoS par rapport au taux de données, considère les protocoles d'agrégation de données maximisant la quantité d'informations communiquées au nœud puits soumises à des contraintes d'énergie, la latence et le flux de données.
- Le second cas, QoS par rapport aux erreurs, admet les protocoles d'agrégation de données mettant l'accent sur le contrôle de congestion ayant pour objectif principal d'optimiser l'utilisation du canal de transmission.

III.5.3 Agrégation basée sur la topologie du réseau

Cette approche joue un rôle vital dans la réalisation des différents protocoles d'agrégation. [38]

Selon la topologie du réseau, l'agrégation de données peut être divisée en deux types [40]:

- Agrégation dans les réseaux plats
- Agrégation dans les réseaux hiérarchiques

La figure suivante illustre un organigramme qui représente les techniques d'agrégation de données dans le cas où on s'intéresse à l'architecture du réseau.

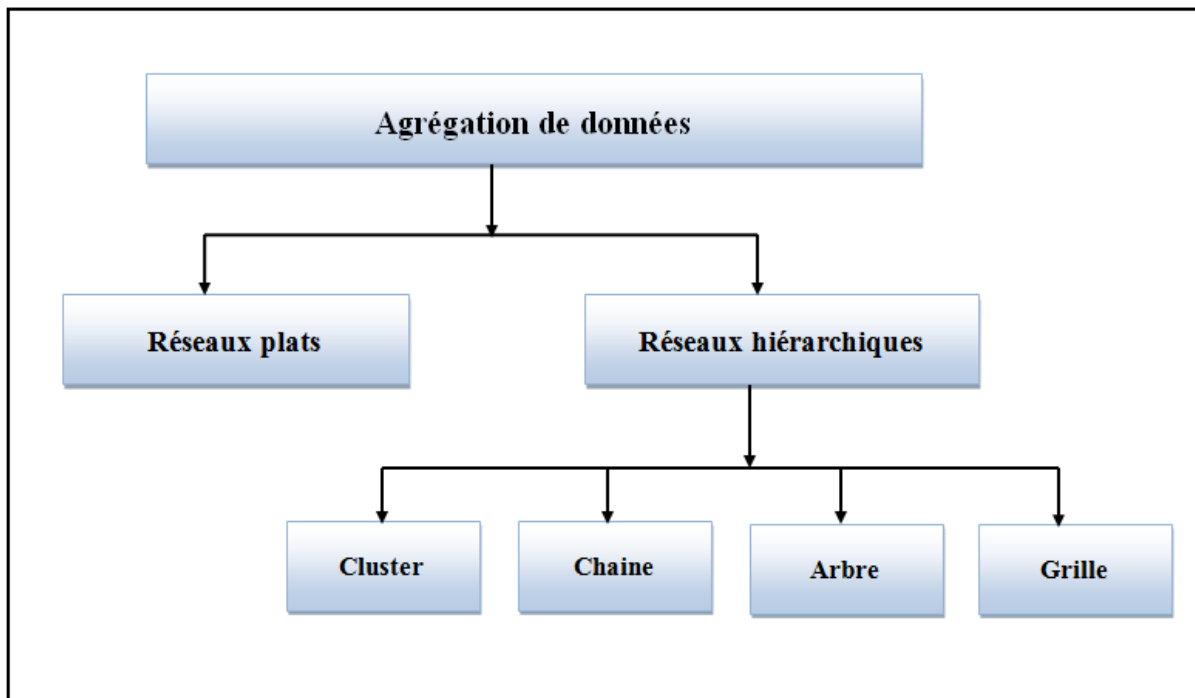


Figure III.2: Agrégation de données basée sur la topologie du réseau [40]

III.5.3.1 Agrégation dans les réseaux plats

Dans le cas des réseaux plats, tous les nœuds sont considérés homogènes c'est-à-dire fonctionnent tous de la même façon et sont dotés approximativement de la même puissance énergétique.

La tâche d'agrégation est accomplie par le cheminement central des données tel que seul le nœud puits soit à la charge de la collecte des données issues des différents nœuds sources.

Les communications et les calculs excessifs sont exécutés au niveau du nœud puits [40] ce qui induit un épuisement rapide de sa puissance énergétique. Et donc, tout épuisement ou échec au niveau du nœud puits reflétera un dysfonctionnement global du réseau.

Pour faire face à la contrainte d'énergie, plusieurs approches hiérarchiques ont été proposées.

III.5.3.2 Agrégation dans les réseaux hiérarchiques

Les réseaux hiérarchiques attribuent des rôles différents aux nœuds du réseau, tels que, les nœuds les plus puissants soient à la charge des tâches les plus coûteuses en énergie, tandis que les nœuds les plus contraints en ressources énergétiques se limitent à la phase du captage.

De ce fait, le processus d'agrégation des données issues des différents nœuds capteurs est exécuté par ces nœuds spéciaux réduisant le nombre de messages communiqués à la station de base et donc, réduction de l'énergie consommée au niveau du réseau et puis augmentation de sa durée de vie.

➤ Agrégation de données basée-cluster

L'ensemble des nœuds capteurs du réseau sont organisés en clusters (groupes) suivant une métrique ou une combinaison de métriques (énergie, voisinage, etc.). Chaque cluster est représenté par un nœud cluster-head (ou chef de groupe) élu parmi les membres du cluster. Le cluster-head se charge de l'exécution du processus d'agrégation des données en provenance des différents membres du cluster, et communique le résultat de l'agrégat au nœud puits de façon directe ou en multi-sauts à travers d'autres cluster-heads ou des nœuds dits relais [37] [40].

La technique d'agrégation appliquée sur un réseau structuré en clusters permet de réduire la complexité des algorithmes de routage (les nœuds effectuent des communications sur de petites distances avec leurs CHs), d'optimiser le support de transmission, de simplifier la gestion du réseau et d'optimiser les dépenses d'énergie. La figure suivante représente un réseau organisé en un ensemble de clusters réalisant l'agrégation de données.

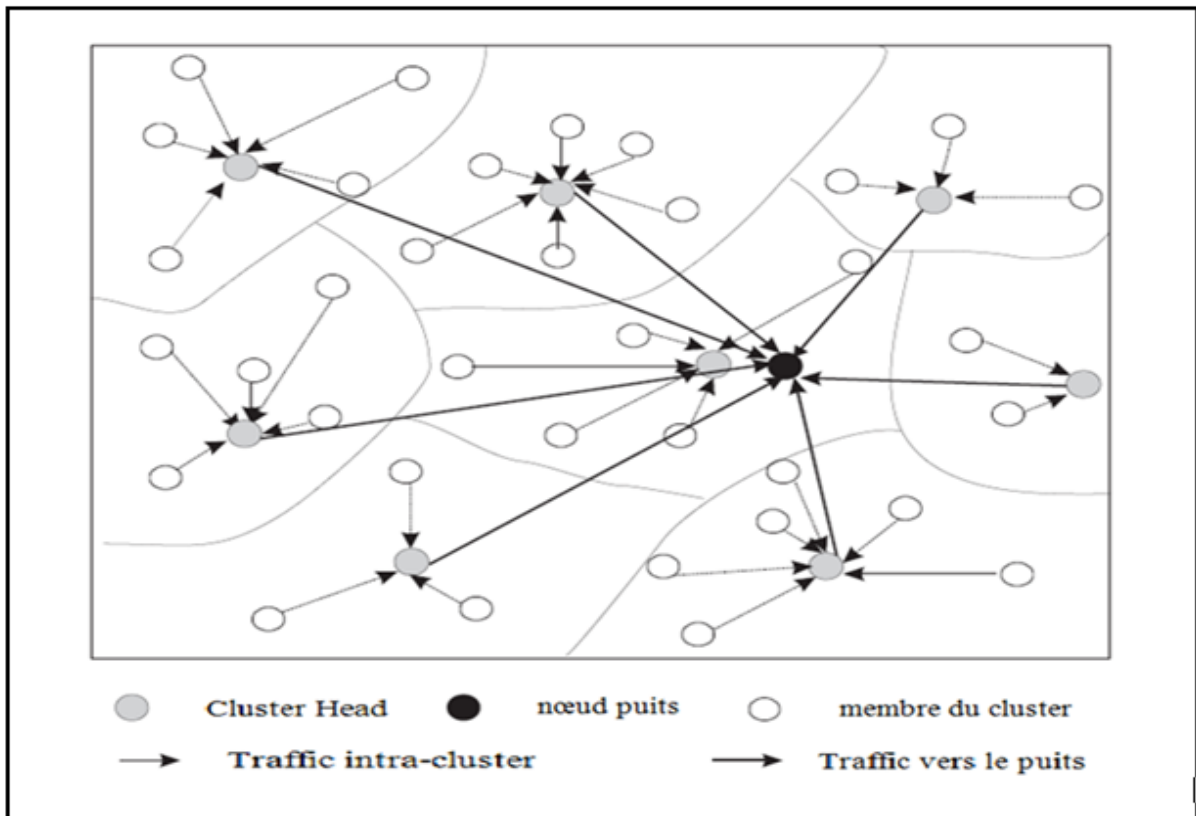


Figure III.3: Agrégation de données basée-cluster [41]

➤ Agrégation de données basée-arbre

Les techniques basées sur les arbres effectuent l'agrégation par construction d'un arbre d'agrégation (figure III.4), dont le nœud racine représente la station de base et les feuilles symbolisent les nœuds sources. Le flux de données commence à partir des nœuds feuilles vers la racine [37].

Chaque nœud intermédiaire dans l'arbre accomplit la tâche d'agrégation sur les données reçues de ses différents nœuds fils, ce processus se répète jusqu'à l'obtention du résultat de l'agrégat qui sera acheminé vers la station de base.

Comme les réseaux de capteurs sans fil ne sont pas exempts aux pannes, cette technique présente certains inconvénients. Dans le cas d'une perte de paquets à n'importe quel niveau de l'arbre, les données seront perdues non seulement pour un seul niveau mais aussi bien pour tout le sous-niveau correspondant. Ainsi, la maintenance de la structure arborescente du réseau est très coûteuse et donc, le système n'est pas robuste.

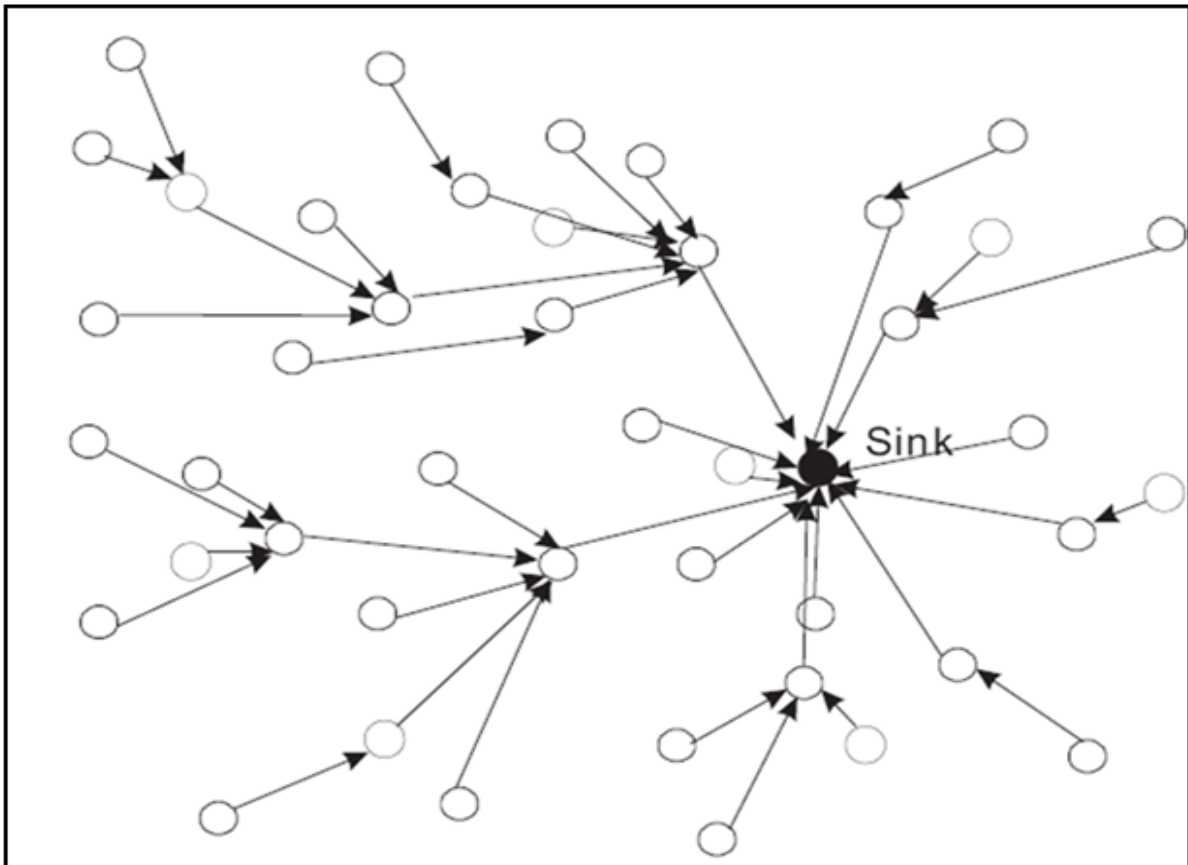


Figure III.4: Agrégation de données basée-arbre [41]

➤ **Agrégation basée-chaîne**

L'idée principale de l'approche basée-chaîne est que chaque nœud du réseau ne puisse communiquer ses données qu'à son voisin immédiat (direct). Et donc, le réseau s'organise de façon à constituer des chaînes de nœuds (illustration: figure III.5).

Lorsque des paquets de données circulent sur cette chaîne, chaque nœud les transmettant peut y agréger ses propres données afin de réduire la quantité transférée. A la fin, un nœud central dit "leader" terminera l'agrégation avant d'envoyer le message résultant au demandeur.

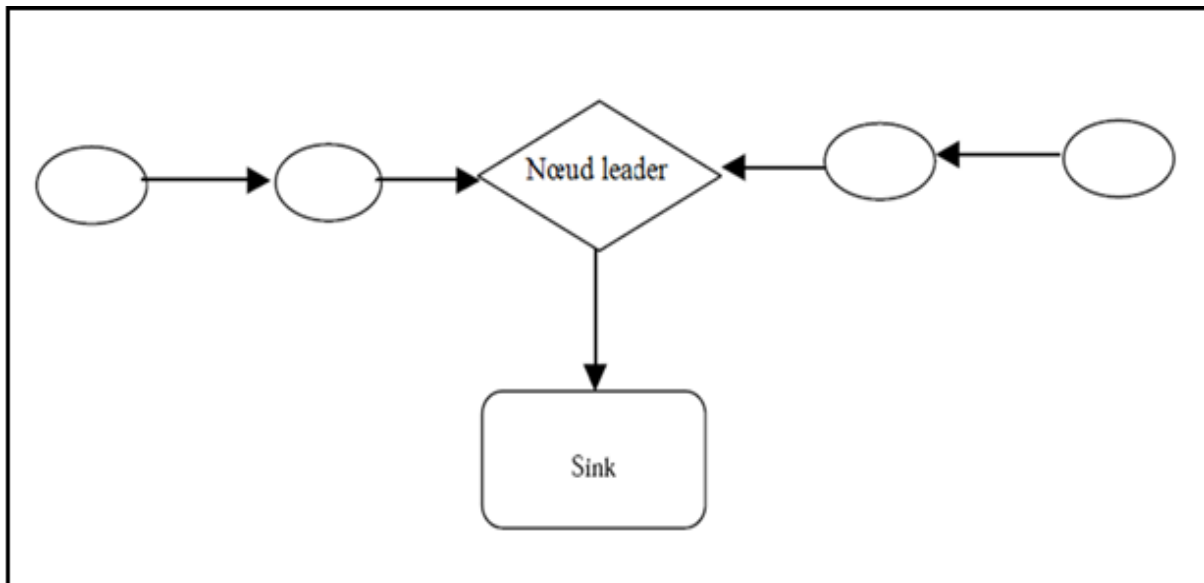


Figure III.5: Agrégation de données basée-chaîne [38]

➤ **Agrégation basée-grille**

Dans cette approche, le réseau est divisé en un ensemble de grilles. Dans chacune des grilles un nœud dit agrégateur est sélectionné pour exécuter le processus d'agrégation sur les données venant directement des différents nœuds de la même grille et puis communique le message résultant à la station de base. Cependant les nœuds dans une même grille ne peuvent pas se communiquer entre eux [35] [41].

La figure suivante montre un réseau de capteurs organisé sous forme de grilles afin de réaliser la tâche d'agrégation.

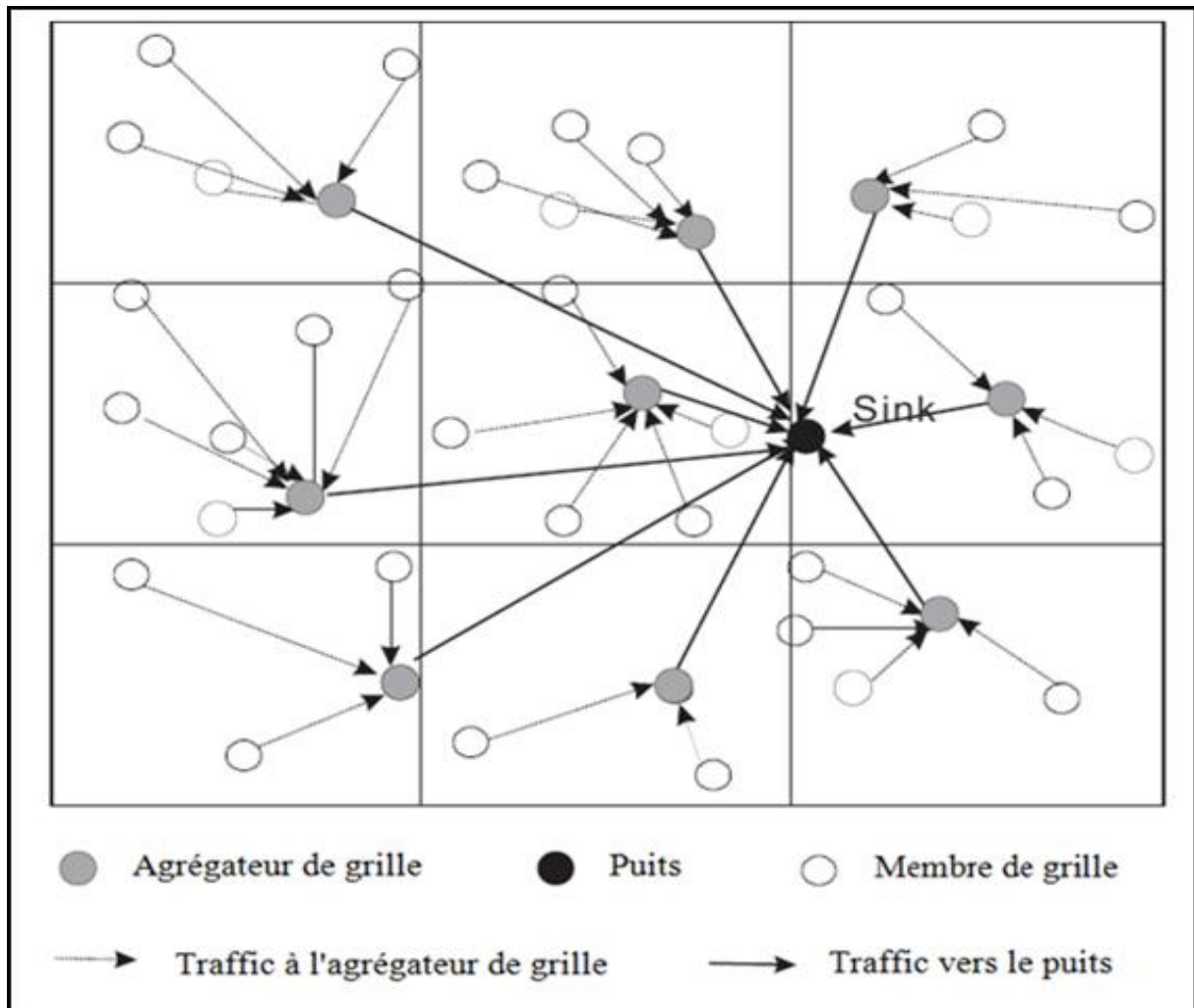


Figure III.6: Agrégation de données basée-grille [41]

III.6 Quelques approches d'agrégation de données basées sur les réseaux hiérarchiques:

Dans cette partie du chapitre, nous allons établir un état de l'art de quelques protocoles appliquant la technique d'agrégation sur des réseaux hiérarchiques.

III.6.1 LEACH: Low Energy Adaptive Clustering Hierarchical

Après la phase de structuration du réseau en clusters (section II.5.1), une phase de communication est lancée. La phase "steady-state" est la phase de communication durant laquelle tous les nœuds peuvent entamer l'envoi de leurs données à leurs cluster-heads pendant les slot d'émission qui lui ont été attribués. Ce mécanisme permet aux nœuds capteurs d'éteindre leur interface de communication en dehors de leurs slots d'émission, ce qui a pour effet d'économiser leur énergie [30].

Ensuite, le CH agrège les données reçues des membres de son cluster avant de les communiquer au collecteur (la station de base). LEACH suppose que la communication entre un CH et le collecteur se fait d'une manière directe i.e. Les CHs peuvent atteindre la station de base en un seul saut.

La figure ci-après, explicite la phase de communication du protocole LEACH dans un réseau organisé en clusters.

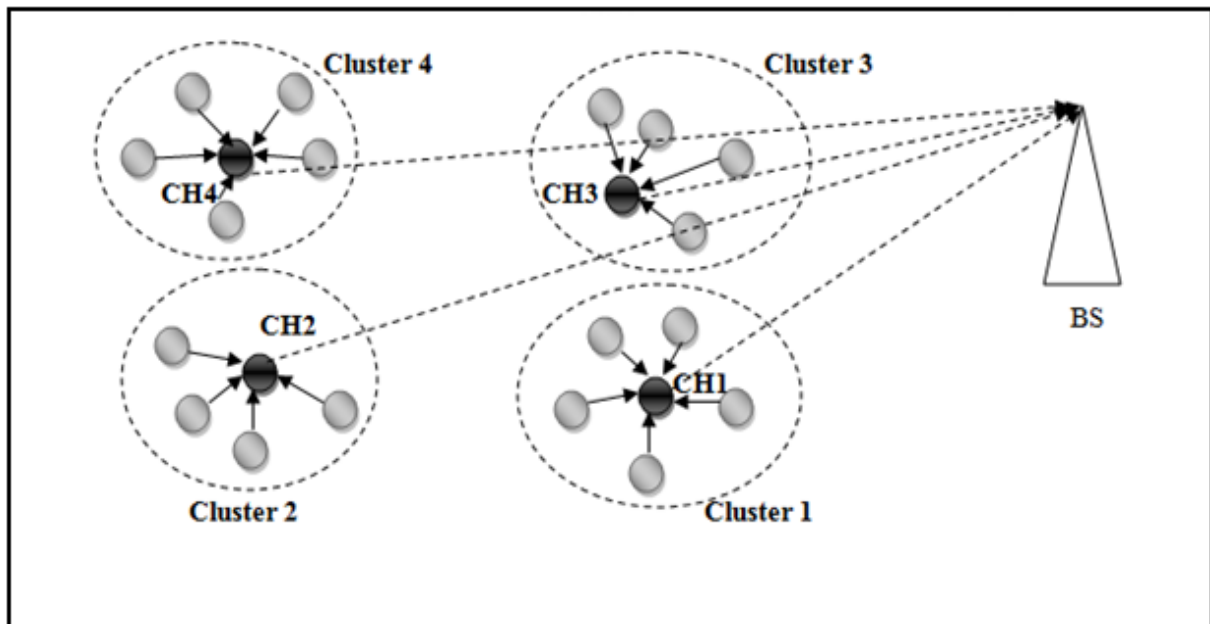


Figure III.7: La phase de communication dans LEACH [32]

Après que la phase de communication soit achevée, la phase de construction recommence à nouveau.

III.6.2 HEED: Hybrid Energy Efficient Distributed clustering approach

Après que les clusters soient formés suivant la démarche présentée dans la section II.5.1, une phase de communication de données commence. HEED suit la même démarche que celle du protocole LEACH tels que les nœuds d'un même cluster transmettent leurs données à leur CH qui exécutera la tâche d'agrégation sur ces dernières afin d'obtenir un seul paquet qui sera acheminé pas directement mais en multi-sauts à travers d'autres CHs vers la station de base.

III.6.3 TAG: Tiny Aggregation [44]

Dans cette approche, un arbre d'agrégation est mis en place par le nœud puits étant la racine. Ce protocole fonctionne en deux phases: distribution et collection:

Durant la phase de distribution, une requête d'agrégation est diffusée par le nœud puits dans le réseau tandis que les autres nœuds s'organisent sous forme d'un arbre d'agrégation tel que le nœud puits représente la racine de l'arbre.

La requête diffusée possède un champ "niveau" qui s'incrémente à chaque retransmission. Ainsi, chaque nœud choisit un de ses voisins possédant un niveau plus inférieur comme parent dans l'arbre d'agrégation.

La phase de collection est initiée à partir des nœuds feuilles vers la racine. Les données agrégées seront continuellement acheminées vers le haut i.e. des fils aux parents jusqu'à atteindre la racine tel que, chaque nœud parent (agrégateur) se mette en attente de l'arrivée des données de ses fils afin de les agréger avec ses propres données avant de les communiquer.

TAG peut être inefficace dans le cas des topologies dynamiques ou cas d'échec d'un nœud de l'arbre, ainsi que la réorganisation et la maintenance de la structure de l'arbre sont très coûteuses en énergie.

III.6.4 PEGASIS: Power-Efficient Data Gathering protocol for Sensor Information System [35] [38]

PEGASIS est un protocole d'agrégation de données basée chaîne. Son idée de base est que les nœuds s'organisent de telle manière à ce qu'ils forment une chaîne linéaire, en utilisant un algorithme avide (greedy algorithm) ou déterminée de façon centralisée par le puits.

L'algorithme avide suppose que chaque nœud ait une connaissance globale du réseau. Le nœud le plus lointain du puits lance la formation de la chaîne et, à chaque étape, le voisin le plus proche (voisin direct) d'un nœud est choisi au tant que son successeur dans la chaîne. Après que la chaîne soit formée, chaque nœud possèdera deux voisins: un prédécesseur et un successeur.

La première transmission des données est lancée par les nœuds des extrémités de telle manière à ce que chaque nœud recevant un paquet de données de son voisin (prédécesseur) l'agrège avec ses propres données puis transmet le paquet résultant à son successeur dans la chaîne. Par la suite, les paquets arriveront à un nœud chef choisi, dit leader, qui se charge de la communication directe du résultat final au nœud puits.

L'inconvénient principal de PEGASIS est la nécessité de la connaissance globale des positions de tous les nœuds afin de sélectionner les voisins appropriés et minimiser la distance maximale entre nœuds. Ainsi, la longueur de la chaîne formée peut influencer négativement sur les délais de communication tandis que les retards puissent être intolérables.

III.7 Impact de l'agrégation de données sur les RCSFs:

L'exécution du processus d'agrégation sur les données interceptées par les différents nœuds capteurs permet [30] [37] de:

- réduire l'overhead (superflus, surcharge) des paquets redondants communiqués dans le réseau et donc, éviter les traitements et les stockages inutiles;
- réduire le trafic circulant dans le réseau et donc diminuer sa charge ainsi que le taux de communications;
- filtrer de manière intelligente les données communiquées à la station de base.

Ce processus permet d'alléger d'avantage le réseau, en effet, il admet une conservation efficace de la bande passante allouée, de l'énergie des nœuds et puis garantie une longue durée de vie du réseau.

L'agrégation de données a un effet très positif sur les réseaux de capteurs sans fils mais cela n'exclut pas quelques inconvénients, citons:

- augmentation du délai de livraison des paquets au puits, dû au fait que chaque nœud agrégateur doit attendre une certaine durée prédéfinie pour recueillir des données auprès de ses voisins; et donc, le besoin d'une synchronisation stricte entre un nœud agrégateur et son voisinage peut être critique dans le cas des paquets sensibles.
- perte de la précision et l'intégrité des données en cas d'agrégation inopportune;
- nécessité d'un moyen de conservation actif des données livrées à la destination en raison de grandes possibilités d'altération;

Cependant, il existe un compromis entre l'économie d'énergie, l'exactitude et la cohérence des données ainsi que la fraîcheur; i.e. plus la durée d'attente d'un nœud est grande plus la possibilité de recevoir de nouvelles données augmente et donc, des informations plus précises sont communiquées au puits. D'autre part, un grand délai d'attente pourra entraîner des données non valides (périmées) qui pourront interférer avec les prochaines données.

Conclusion

L'agrégation de données est une technique prometteuse dans les réseaux de capteurs sans fil, permettant un gain important de ressources épuisées en raison des redondances et la surcharge du réseau.

Dans ce chapitre, nous avons défini le processus d'agrégation tout en précisant son principe de fonctionnement. Puis, nous avons établi une étude détaillée sur cette technique, qui a permis de distinguer ses différentes approches et types classifiés selon des critères et suivant les besoins d'application. Par la suite, nous avons établi un état de l'art de quelques protocoles utilisant cette technique.

Et d'après cette étude, nous avons pu conclure l'impact et l'effet de cette technique sur les réseaux de capteurs sans fil en explicitant quelques avantages et limitations.

Dans la suite de ce travail, nous allons proposer un protocole d'agrégation de données basée sur la méthode de clustering afin de gérer la consommation de la ressource énergétique dans les RCSFs.

DESCRIPTION DU PROTOCOLE PROPOSE

Introduction

Un très grand nombre de travaux de recherche dans les RCSFs se concentrent sur la contrainte d'énergie afin de concevoir des algorithmes et des protocoles spécifiques à ces réseaux optimisant au maximum la conservation d'énergie. Dans ce chapitre, nous allons présenter l'approche proposée suivant le plan méthodologique suivant:

Nous allons d'abord expliciter le principe de base du protocole puis nous présenterons le processus permettant sa conception. Nous allons dans un premier temps détailler la phase de structuration du réseau en clusters illustrée par des organigrammes (section IV.2.2). Ensuite, nous allons présenter dans la section IV.2.4 la phase d'agrégation appliquée sur les différentes données collectées dans le réseau. Et, nous allons finir par donner un aperçu général sur les différents messages échangés ainsi que les structures de données utilisées dans l'algorithme.

IV.1 Principe du protocole:

Le principe de base du protocole proposé est de gérer l'énergie consommée par les différents nœuds en adoptant une technique d'agrégation sur les données collectées dans le réseau. Le protocole opère en deux phases de façon cyclique: la première phase consiste en la structuration virtuelle du réseau en un ensemble de clusters après une élection des cluster-heads suivant une méthode de calcul du poids. La seconde phase permet d'appliquer une technique d'agrégation afin de réduire le nombre de paquets communiqués au nœud puits, dans le but d'éviter une surconsommation d'énergie qui peut être causée par la présence de données redondantes ou fortement corrélées dans le réseau. Pour cela quelques hypothèses sont préalablement posées:

- Le réseau de capteurs est stationnaire (Les nœuds sont immobiles);
- Les nœuds capteurs sont initialement homogènes tels qu'ils sont similaires dans leur capacité de traitement, de communication, d'énergie et leur capacité de stockage qui est supposée suffisante pour sauvegarder les différentes structures de données;
- Le déploiement des nœuds est aléatoire dans une zone d'intérêt de forme carrée ou rectangulaire (dimension $L \times l$);
- La station de base n'a pas de contraintes sur l'énergie ni sur les capacités de calcul et de stockage.

IV.2 Processus de conception du protocole:

Ci-dessous, Les différentes étapes suivies pour l'élaboration du protocole sont présentées.

IV.2.1 Découverte du voisinage:

Initialement, chacun des nœuds du réseau diffuse un message hello "msg_hello" contenant son identifiant "Src_Id", afin de découvrir et connaître son voisinage à 1-saut; et maintient une liste pour sauvegarder les identifiants de ses voisins directs, dite liste de voisinage "list_neighbor".

Lors de la réception d'un msg_hello, le nœud met à jour sa table de voisinage en éditant l'identifiant de l'émetteur i.e. son 1-voisin. Donc, après cette phase d'échange de messages hello, chaque nœud détecte tous ses liens directs.

L'exécution cyclique du protocole admet un échange périodique des messages hello, donc permet de détecter la perte de liens entre les voisins ou de créer de nouveaux liens. Après que cette étape soit achevée, le processus de clustering se lance.

IV.2.2 Phase de clustering:

L'algorithme de clustering utilisé est distribué tel qu'il s'exécute au niveau de chaque nœud du réseau. Ainsi, chacun d'eux décide, indépendamment des autres, de son rôle et fait connaître sa décision par l'envoi de messages. Chaque nœud peut être dans l'un des trois états suivants: non décidé pour celui qui est récemment injecté dans le réseau, cluster-head ou membre.

Dans cette entité, nous présentons d'abord la méthode d'élection des cluster-heads, puis nous exposons les deux principales phases de la technique de clustering: la formation des clusters et la maintenance de ces derniers.

– Election de cluster-head:

Un algorithme de clustering doit être capable d'élire les nœuds appropriés pour assurer le rôle de cluster-head. Comme déjà mentionné dans le chapitre II, une multiplicité de métriques peut être utilisée pour le choix des CHs.

L'habilité de chaque nœud à être CH est déterminée en fonction d'une combinaison de métriques: la quantité d'énergie disponible, le degré de connectivité du nœud, ainsi que la distance moyenne le séparant de ses voisins.

Dans notre cas, nous avons introduit la métrique suivante appelée Weight ou Poids:

$$\text{Weight (i)} = \alpha \frac{E_i}{E_{max}} + \beta \frac{d_i}{d_{max}} + \gamma \frac{D_i}{D_{max}}$$

Avec:

E_i représente la quantité d'énergie résiduelle du nœud i

E_{max} représente la quantité d'énergie maximale

d_i est le degré de connectivité du nœud i

d_{max} est le degré de connectivité maximal du nœud représentant la taille du réseau

D_i représente la distance moyenne du nœud i par rapport à ses voisins

D_{max} est la distance maximale pour un nœud

α, β, γ : constantes de pondération tel que $\alpha + \beta + \gamma = 1$ et $\alpha > \beta > \gamma$.

Après que le nœud ait déterminé son poids, le sauvegarde dans une variable "weight_" puis le diffuse à son 1-voisinage (voisins immédiats) par un message de poids appelé My_weight.

A la réception de ce paquet, les nœuds mettent à jour leurs tables de voisinage en éditant le poids. Cette table est maintenue durant la phase de formation des clusters et contient les champs suivants:

Src_ID	Weight_

Tableau IV.1: Structure de la table de voisinage des nœuds

A la fin d'un intervalle de temps préfixé, chaque nœud compare son poids avec ceux de son 1-voisinage et celui possédant le plus grand poids s'élit pour rôle de cluster-head. En cas d'égalité, le nœud ayant plus d'énergie est choisi. Donc, deux nœuds voisins ne peuvent pas être tous deux CHs au même temps. Cette étape de la phase de clustering peut être résumée par l'organigramme suivant:

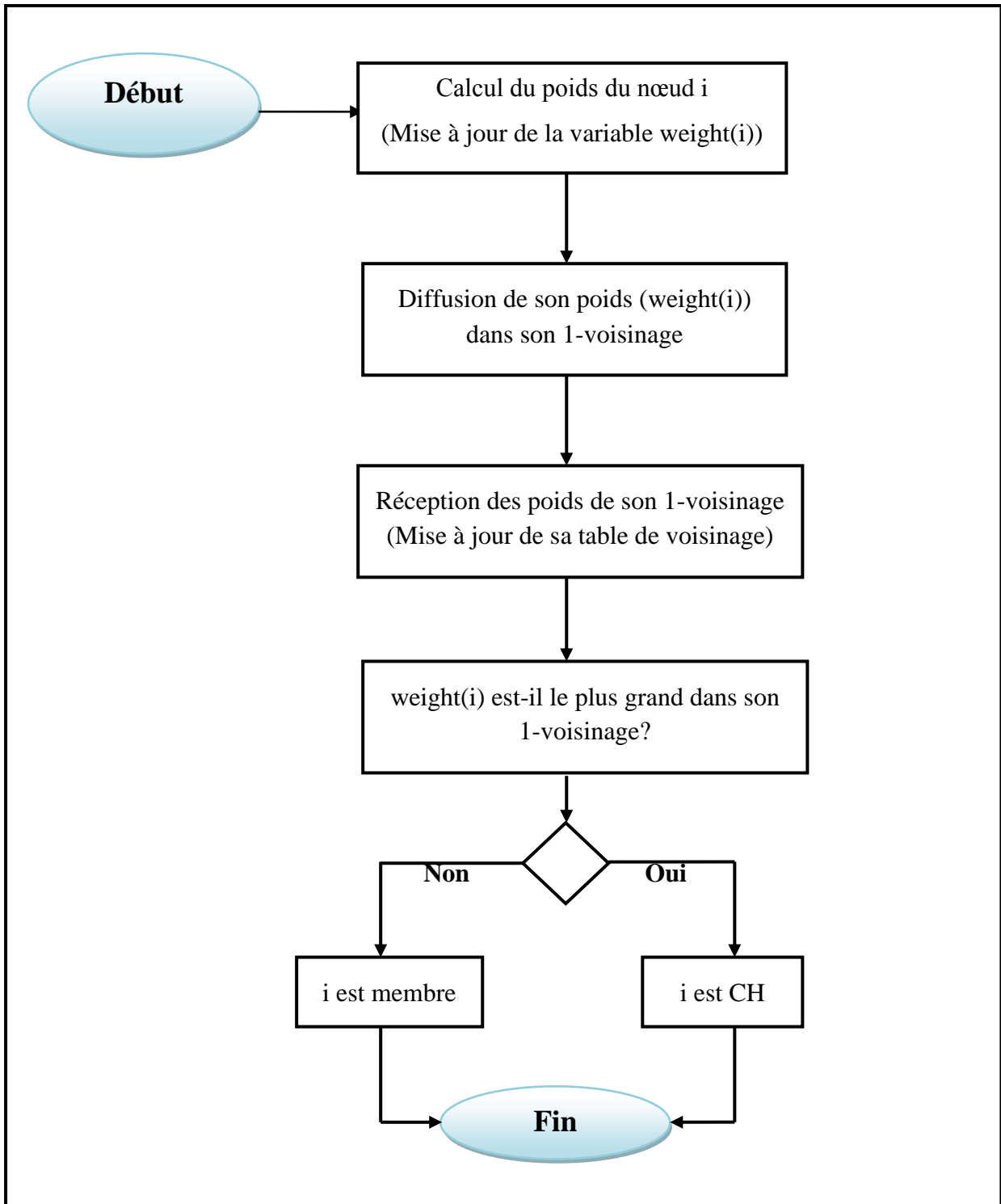


Figure IV.1: Élection des cluster-Heads

– **Formation des clusters:**

Une fois qu'un nœud du réseau s'élit CH, il doit annoncer son nouveau statut dans le cycle courant aux autres nœuds de son voisinage par la diffusion d'un message d'annonce msg_ch_adv. Lors de la réception des messages d'annonce, deux cas de figures peuvent se présenter:

Si (le nœud récepteur est un CH) **alors**

//il vérifie sa liste des membres "list_member" dont il sauvegarde les nœuds joignant son cluster

Si (list_member est vide) **alors**

- Il se désistera de son rôle de CH;
- Il rejoint le CH l'ayant invité, avec le plus grand poids;

Sinon il communique ses données jusqu'à aboutir la BS;

Fin si;

Sinon //le nœud est non cluster-head

- Il maintient une liste des CHs (list_CH) contenant les identifiants des cluster-heads, leurs poids ainsi que leurs positions;
- Il fixe un temps d'attente des annonces Tadv;

– **Si** (Tadv est écoulé) **alors**

Le nœud prend la décision d'appartenance à un CH suite à une comparaison des poids tel que le nœud CH doté du plus grand poids prévaut. En cas de conflit de poids i.e. deux cluster-heads avec un même poids, le moins distant est sélectionné.

Fin si;

Fin si;

La liste list_CH qui maintient l'ensemble des cluster-heads a la structure suivante:

ID_CH	Weight_CH	Position_CH

Tableau IV.2: Structure de la liste des cluster-heads

La décision de joindre un cluster-head est confirmée par l'envoi d'un message de jointure Msg_join_req qui contient son identifiant ainsi que l'identifiant du CH à joindre.

D'autre part, au niveau de chaque cluster-head une liste de ses membres "list_member" est maintenue. Cette dernière est interrogée par la suite dans la phase de collecte de données afin d'appliquer le processus d'agrégation. Cette phase est résumée par l'organigramme qui suit:

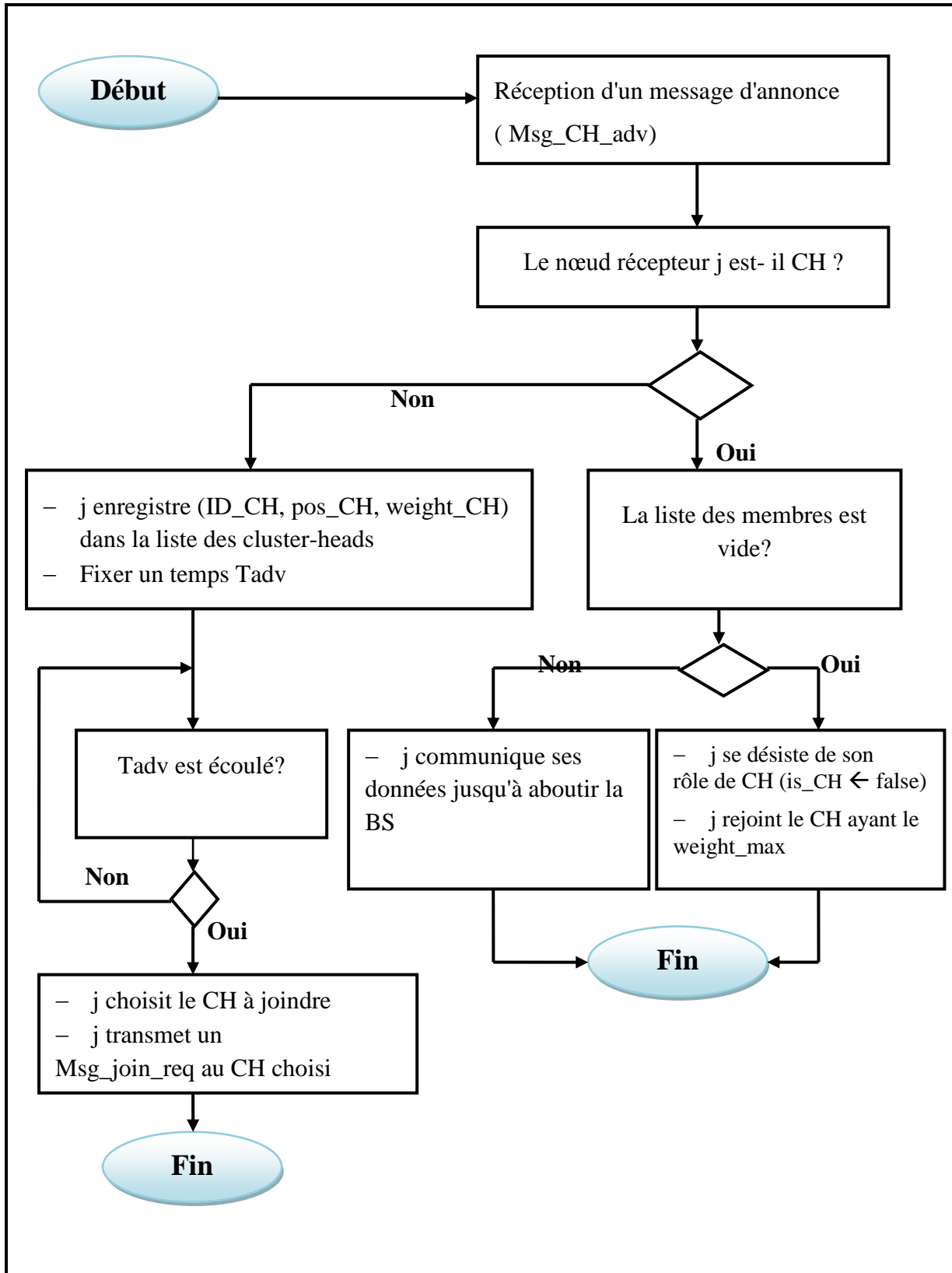


Figure IV.2: Formation des clusters

– **Maintenance des clusters:**

L'algorithme de clustering conçu doit être adaptatif, c'est-à-dire capable de prendre en considération les changements de la topologie du réseau ainsi, reconnaître les nœuds ajoutés ou ceux disparus.

- **Cas de défaillance d'un nœud:**

Lorsqu'il s'agit d'un nœud ordinaire, sa disparition n'affecte pas trop le fonctionnement du protocole, car ce type de nœuds ne s'occupe que de la fonction de captage. Mais quand il s'agit d'un nœud CH qui est d'une importance capitale pour le réseau, sa disparition a un effet considérable sur le fonctionnement du protocole.

Les nœuds cluster-heads reçoivent beaucoup de paquets, exécutent le processus d'agrégation puis communiquent le résultat à la station de base. Ils sont donc ceux dont l'énergie sera épuisée rapidement, surtout s'ils sont élus pour une longue période.

En effet, si un nœud garde son rôle de cluster-head pour longtemps même s'il ne possède pas, par exemple, le plus grand poids dans son propre cluster, il se désistera de son rôle une fois son niveau d'énergie est descendu au dessous d'un seuil prédéfini. Pour cela, notre protocole est conçu de façon à ce qu'il s'exécute en plusieurs cycles en relançant le processus de clustering périodiquement.

De cette manière, une répartition efficace de la consommation énergétique est assurée grâce à une réélection périodique des cluster-heads. Ce mécanisme permet de maintenir les nœuds pour une longue durée de vie et puis le réseau en entier.

- **Cas d'ajout d'un nœud:**

Dans le cas où un nouveau nœud rejoint le réseau, se met en état non décidé. Puisque le protocole fonctionne de façon cyclique, ce nœud sera reconnu et pris en considération lors de la phase de découverte de voisinage du prochain cycle.

La figure suivante montre les différentes étapes suivies par notre protocole pour la structuration virtuelle du réseau en clusters.

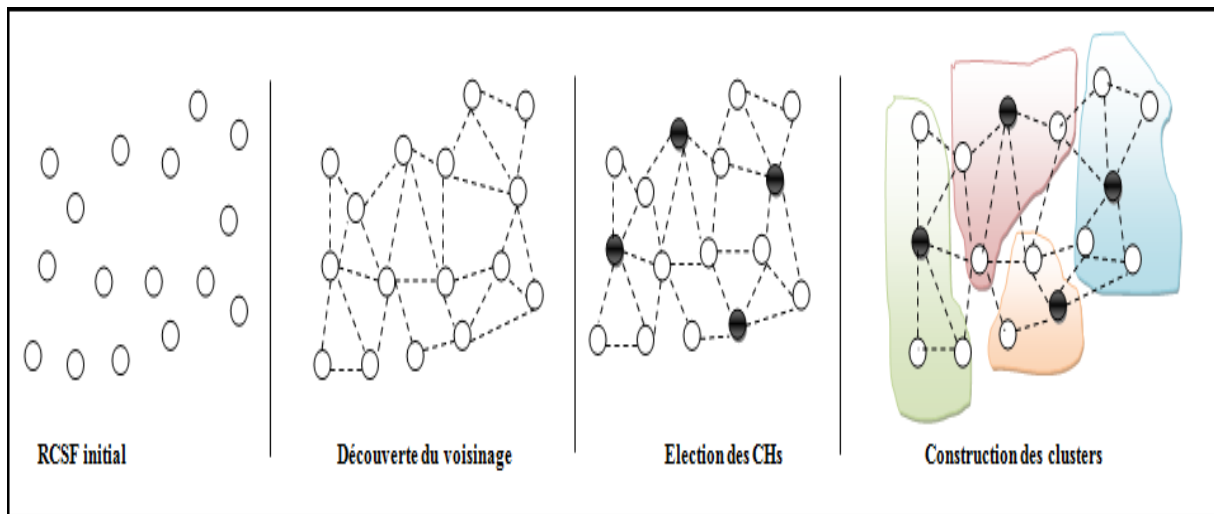


Figure IV.3: Etapes de construction d'une topologie en clusters

IV.2.3 Phase de collecte de données:

Cette phase se déroule au niveau du nœud puits et au niveau des nœuds cluster-heads.

– Au niveau CHs:

Au niveau du nœud cluster-head, deux types de collecte de données sont distinguées: la collecte des données de sa propre unité de captage et celles en provenance des autres nœuds capteurs membres de son propre cluster. Afin d'accomplir cette phase, chaque nœud CH:

- Maintient une liste de données "list_Data" dont il sauvegarde les données qui lui ont été envoyées par les membres de son cluster;
- Initialise un timer, afin d'éviter une attente indéfinie des données.

En effet,

lorsque la phase de collecte de données soit lancée, chaque CH initialise son propre timer. Et, à la réception d'une donnée, le cluster-head la rajoute à sa propre liste de données. A l'expiration du timer, la phase d'agrégation est déclenchée.

– Au niveau puits:

La station de base se charge de la collecte des données agrégées en provenance des nœuds cluster-heads. De même, les données venant des nœuds isolés ne faisant partie d'aucun cluster.

La phase de collecte de données au niveau des cluster-heads est illustrée par la figure suivante:

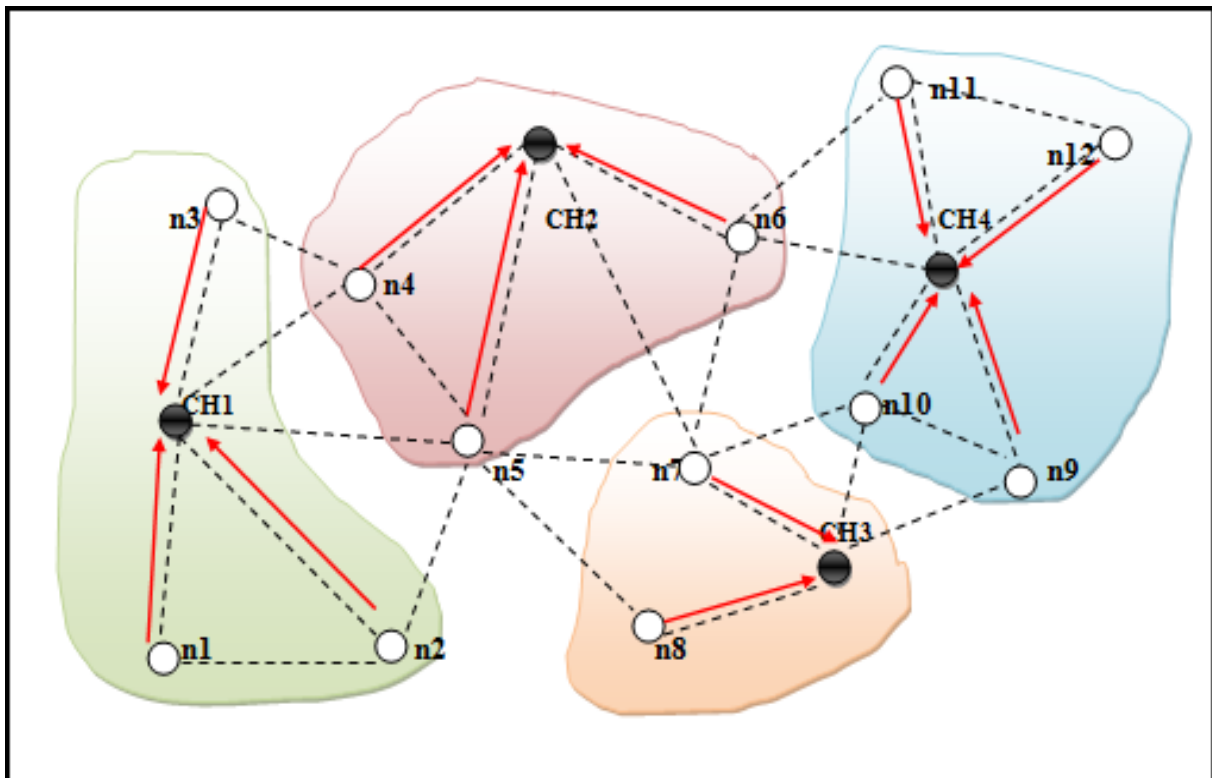


Figure IV.4: Illustration de la collecte des données au niveau CHs

IV.2.4 Phase d'agrégation:

Durant cette phase, la fonction d'agrégation de données f est invoquée. A partir des données qu'un cluster-head ait stockées dans sa `list_Data`, un seul paquet est généré dont le champ "Data" est la moyenne arithmétique de ces dernières.

Lorsqu'un nœud cluster-head ait récolté les différents paquets de données en provenance des membres de son cluster ainsi que sa propre donnée, il applique une fonction qui permet de calculer la moyenne arithmétique. Le résultat sera acheminé à la station de base dans un paquet de données `Msg_Data`.

En effet, la quantité de messages communiqués est considérablement réduite, et puis la charge du réseau entier est allégée. Et donc, cette technique joue un rôle déterminant dans la conservation d'énergie.

IV.3 Organisation:

Dans cette entité du chapitre, nous nous contenterons d'énumérer les données assignées aux nœuds ainsi que décrire les messages échangés entre eux.

IV.3.1 Données:

Les données de chaque nœud diffèrent selon la tâche qu'il assure dans son cluster. Elles fournissent une description de son état et celles des nœuds avec lesquels il interagit. Deux types de données peuvent être distingués: données sur soi et sur les autres. Le tableau suivant résume ces données.

Données	
Données sur lui-même	Données sur les autres
<ul style="list-style-type: none"> ✓ ID ✓ Rôle ✓ Energie restante ✓ Degré de connectivité ✓ Position ✓ Poids 	<ul style="list-style-type: none"> ✓ ID_ch ✓ Position_ch ✓ Liste_neighbor ✓ List_ch ✓ List_member (*) ✓ List_Data(*)

Tableau IV.3: Les différentes données d'un nœud

(*) Si le nœud est CHs

IV.3.2 Les structures de données utilisées:

Notre protocole utilise plusieurs structures de données pour assurer son fonctionnement:

- **La liste list_neighbor:** cette structure est utilisée par chaque nœud du réseau afin d'enregistrer les identifiants de ses1-voisins durant la phase de découverte de voisinage et par la suite est mise à jour en éditant le poids de chacun d'eux.
- **La liste list_ch:** cette liste est utilisée par les nœuds non-CHs afin de sauvegarder les Msg_ch_adv en provenance des cluster-heads voisins et choisir auquel ils doivent adhérer. Chaque entrée de la liste contient l'identifiant du CH, son poids et sa position afin de pouvoir calculer la distance qui sépare le nœud du CH en cas de conflit de poids.

- **La liste list_member:** cette liste est utilisée par les nœuds CHs pour maintenir les nœuds ayant répondu à leurs messages Msg_CH_adv et pouvoir allouer un intervalle de temps pour la phase de transmission des données. Chaque entrée de la liste contient l'identifiant du nœud.
- **La liste list_Data:** cette structure est utilisée par un nœud cluster-head afin de sauvegarder les données reçues des membres de son cluster, qui seront agrégées par la suite dans un seul paquet en appliquant la fonction d'agrégation.

IV.3.3 Interactions: Description des messages

- **Msg_hello:** ce message est diffusé par chaque nœud dans le réseau durant la phase de découverte de voisinage, afin que chacun d'eux puissent sauvegarder ses 1-voisins dans la liste list_neighbor et calculer son poids par la suite.

Champ	Description
Src_ID	Identifiant du nœud émetteur
Src_pos	Position du nœud émetteur

Tableau IV.4: Entête du paquet Hello

- **Msg_weight:** après que chaque nœud ait calculé son poids, le diffuse à l'aide de ce message à ses voisins directs et les met en connaissance pour de prochaines phases de clustering.

Champ	Description
Src_ID	Identifiant du nœud émetteur
Dest_ID	Identifiant du nœud voisin de destination
Weight	Poids du nœud émetteur

Tableau IV.5: Entête du paquet Weight

- **Msg_ch_adv:** ce type de messages est envoyé par un nœud qui a décroché le rôle de CH afin de faire savoir son nouveau statut à son 1-voisinage et les inviter à rejoindre son cluster.

Champ	Description
ID_ch	Identifiant du nœud cluster-head
Pos_ch	Position du nœud cluster-head
Dest_ID	Identifiant du 1-voisin destination
Weight_ch	Poids du nœud cluster-head

Tableau IV.6: Entête du message d'annonce

- **Msg_join_req:** ce message est transmis par les nœuds non-CHs qui ont déjà reçu un ou plusieurs messages `Msg_ch_adv` pour faire savoir leurs décisions au cluster-head qu'ils rejoignent. Les nœuds choisissent un cluster-head qui possède le poids le plus élevé.

Champ	Description
Src_ID	Identifiant du nœud émetteur (membre)
Ch_ID	Identifiant du cluster-head à joindre

Tableau IV.7: Entête du message de jointure

- **Msg_Data:** les nœuds membres d'un cluster font appel à ce type de message pour communiquer leurs données, à leur propre cluster-head, sur lesquelles la fonction d'agrégation est appliquée. De même, utilisé par un nœud cluster-head afin de communiquer le paquet résultant de la phase d'agrégation à la station de base.

Champ	Description
Src_ID	Identifiant du nœud émetteur (membre ou CH)
Dest_ID	Identifiant du nœud destination (CH ou BS)
Data	Données captée ou résultat d'agrégation
Time_stamp	Temps d'envoi de la donnée

Tableau IV.8: Entête du message de données

Conclusion

Lors de ce chapitre, nous avons présenté une description détaillée de notre protocole, en exposant son principe de fonctionnement, la démarche suivie pour structurer le réseau en clusters, et la technique d'agrégation appliquée afin de réduire et alléger le trafic généré dans le réseau. Après cette phase de description, nous allons passer à l'implémentation de toutes les étapes de notre étude ainsi que la présentation des résultats démonstratifs qui les justifient. Donc le prochain chapitre est consacré pour l'implémentation du protocole sous l'environnement J-Sim afin de simuler et évaluer ses performances.

**IMPLEMENTATION ET
SIMULATION DU
PROTOCOLE**

Introduction

Après avoir présenté une description détaillée de notre protocole d'agrégation de données basé sur l'approche de clustering au cours du chapitre précédent, une implémentation de celui-ci est réalisée.

L'objectif de ce chapitre est de démontrer l'efficacité du protocole proposé en termes d'énergie et de délai de transmission. Donc, les simulations effectuées permettent d'évaluer les performances de la solution. Pour cela, nous commencerons par définir les outils nécessaires pour l'implémentation et la simulation du protocole. Ensuite, nous décrirons la mise en œuvre des processus décrits lors de la conception. Nous terminerons ce chapitre par une présentation des résultats relevés lors des tests.

V.1 Différentes approches de test

Avant la mise en place du protocole, le déploiement d'un réseau de capteurs nécessite une phase de test afin de s'assurer du bon fonctionnement de tous ses composants; matériel ou logiciel. Pour se faire, trois grandes solutions [57] seront exposées: le test en environnement réel, la simulation et l'émulation.

V.1.1 Test en environnement réel

Une première approche pour tester un protocole ou une application consiste à mettre réellement en œuvre le réseau sans fil sur lequel on doit le ou la tester. Cette solution est a priori la meilleure dans la mesure où l'on utilise vraiment le système opérationnel ainsi que l'implémentation finale du protocole. Cependant, cette solution peut être assez difficile à mettre en œuvre et peut être très coûteuse dans certains cas.

Cette solution présente plusieurs d'autres points négatifs, citons: l'impossibilité d'évaluer le protocole aux conditions limites de propagation (pendant une tempête par exemple) ou de mobilité (par exemple avec des vitesses de déplacement très importantes). De même, il est très difficile de réussir à mettre en œuvre des cas rares (tous les nœuds qui émettent simultanément par exemple). Enfin, cette solution n'autorise pas l'exacte reproduction des conditions d'une expérience à l'autre, ce qui peut être très gênant pour réussir à mettre au point son protocole.

V.1.2 Simulation

La seconde solution, souvent privilégiée par les chercheurs, est la simulation. Celle-ci permet d'évaluer un modèle d'application ou un protocole dans un environnement gérable. Pour cela, la simulation s'appuie sur des modèles décrivant l'environnement, les couches de communication utilisées par les terminaux sans fil ainsi que d'autres équipements du réseau et un modèle du trafic circulant sur le réseau. Cependant, la simulation ne fonctionne pas en temps réel ce qui empêche, par exemple, l'évaluation d'applications interactives. Utiliser un modèle au lieu de l'implémentation réelle est également pénalisant, car la validité du modèle ne garantit pas le bon déroulement de son implémentation et son déploiement. Au fait, l'implémentation n'est pas privée des erreurs de programmation, donc elle doit ensuite être évaluée en environnement réel pour vérifier qu'elle se comporte selon son modèle.

V.1.3 Emulation

L'émulation peut être vue comme un compromis entre les deux solutions précédentes en permettant d'évaluer un protocole ou une application dans un environnement contrôlable qui simule en temps réel les conditions telles que: les débits, les délais et les pertes que l'on observe dans le réseau cible. Pour cela, l'émulation va simuler les effets des couches basses de telle manière qu'un protocole ou une application s'exécute dans les mêmes conditions que celles de l'environnement réel. Cette solution de test peut par ailleurs être vue comme une étape supplémentaire dans le cycle de développement entre la phase de simulation, qui permet de concevoir les mécanismes spécifiques à un protocole ou à une application et le déploiement dans un environnement réel.

Bien que l'émulation paraisse plus intéressante et porte plus d'avantages, elle impose cependant certaines contraintes. Comme l'émulation fonctionne en temps réel, les modèles utilisés pour simuler les couches basses ne peuvent pas être trop complexes ce qui implique un impact négatif sur le réalisme de l'émulation rendue. A cet effet, l'approche de simulation détient toujours sa place comme solution pour le test et la validation des protocoles et d'applications [56].

V.2 Simulateur J-Sim

Plusieurs environnements de simulation sont mis en place pour tester et évaluer les différents protocoles et architectures proposés pour les RCSFs. Certains sont libres et parfois Open source par contre, il y en a d'autres qui sont commercialisés. En outre, aucun de ces simulateurs ne peut répondre à toutes les exigences et besoins. Chacun d'eux présente des avantages et des inconvénients. Dans cette entité, nous n'allons pas aborder l'étude comparative de ces outils car cela a fait l'objet de plusieurs travaux de recherche dont [49] et [50]; mais nous nous contenterons de la description du simulateur que nous avons choisi pour simuler notre protocole: J-Sim.

V.2.1 Présentation de J-Sim

J-SIM [49] [51] [7] (appelé autrefois JavaSim), a été développé par une équipe du laboratoire Distributed Realtime Computing Laboratory (DRCL) de l'université d'État d'Ohio. Ce simulateur est gratuit et disponible avec le code source, développé entièrement en langage Java, repose sur une architecture logicielle basée sur les composants autonomes, appelée « Autonomous Component Architecture » (ACA). Le code source est organisé en paquetages relatifs à un type de composants.

Un composant est une entité indépendante représentant un objet physique (une batterie, un module radio, une couche logicielle, etc.), ou logique (un protocole de routage, un modèle de mobilité, etc.). Ces composants seront ensuite connectés à l'aide de ports afin de générer un réseau simulé.

La simulation du fonctionnement d'un réseau de capteurs, qui exige la définition des composants et leur mise en relation, est réalisée grâce à un langage spécifique complètement supporté par le simulateur J-Sim, TCL (Tool Command Line) [52]. Il s'agit d'un langage de script dans lequel on spécifie l'architecture du réseau ainsi que les différents paramètres de simulation et d'analyse. Les commandes de script peuvent également être fournies en ligne de commande, instruction par instruction.

A l'aide de TCL, on définit les composants puis on les connecte. Tous les composants sont hébergés dans un conteneur, qui est à son tour un composant. La définition des composants est en fait la création des objets. Cette création est réalisée par la commande TCL *mkdir*. Chaque composant est d'une entité indépendante (la couche MAC par exemple) qui fonctionne indépendamment des autres entités. Les composants possèdent des ports par défaut pour qu'ils puissent communiquer entre eux.

D'autres ports peuvent être créés pour un composant. Une connexion entre deux composants est réalisée par l'intermédiaire de deux ports dédiés, un dans chaque composant (voir la Figure V.1). Cette connexion est matérialisée par la commande TCL *connect*. Il suffit de suivre un schéma qui indique l'interconnexion entre les différents types de composants que l'on veut utiliser. On obtient ainsi l'architecture du nœud. Toujours dans ce script TCL, on y définit les paramètres globaux (par exemple la taille du champ de simulation), les outils de visualisation des résultats de la simulation.

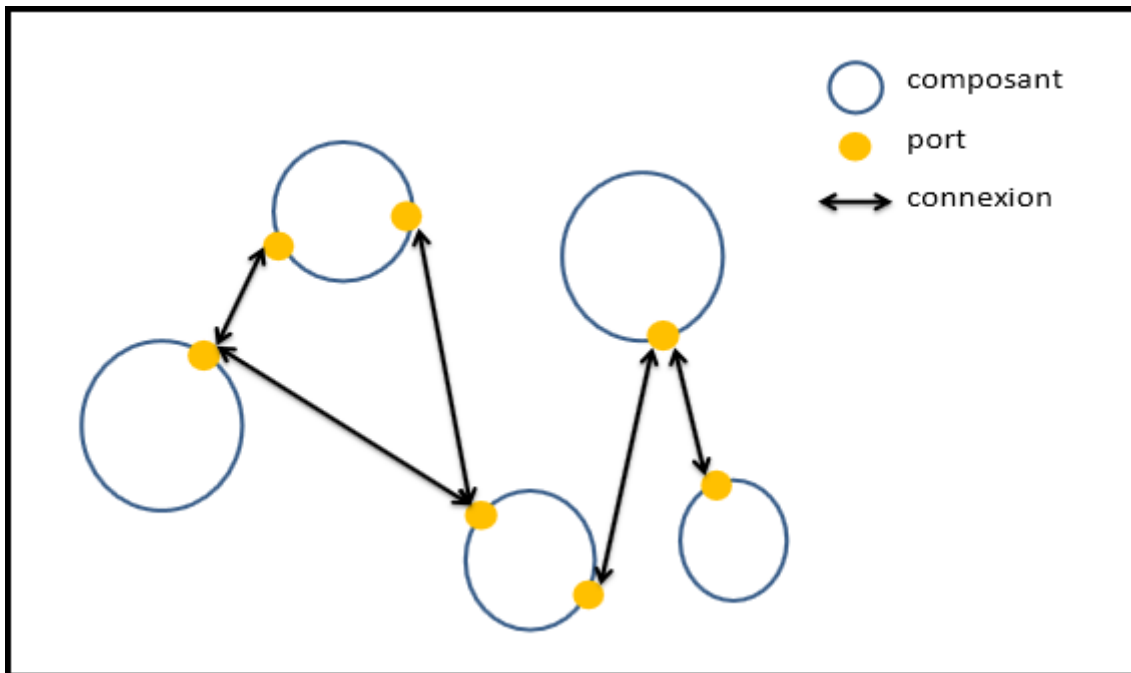


Figure V.1 Connexion entre les composants dans J-Sim [7]

Dans cet environnement de simulation, on distingue trois types de nœuds (figure V.2) [53] : *target*, *sink* et *sensor*.

Les nœuds *target* ont pour rôle de générer des signaux (stimuli) qui peuvent être, par exemple, une secousse sismique, un bruit, etc. Le nœud *sink* représente le nœud destination où les différentes informations collectées dans le réseau doivent arriver. Le reste du réseau est formé par les nœuds *sensor* qui ont pour fonction de capter le signal généré par les nœuds *target*, former les paquets contenant l'information de mesure, et puis les acheminer jusqu'au nœud *sink*.

Lorsqu'un nœud *target* envoie un stimulus et qu'il est capté par un nœud *sensor*, ce dernier crée un paquet contenant l'information de mesure pour la faire parvenir au nœud *sink* à travers le réseau.

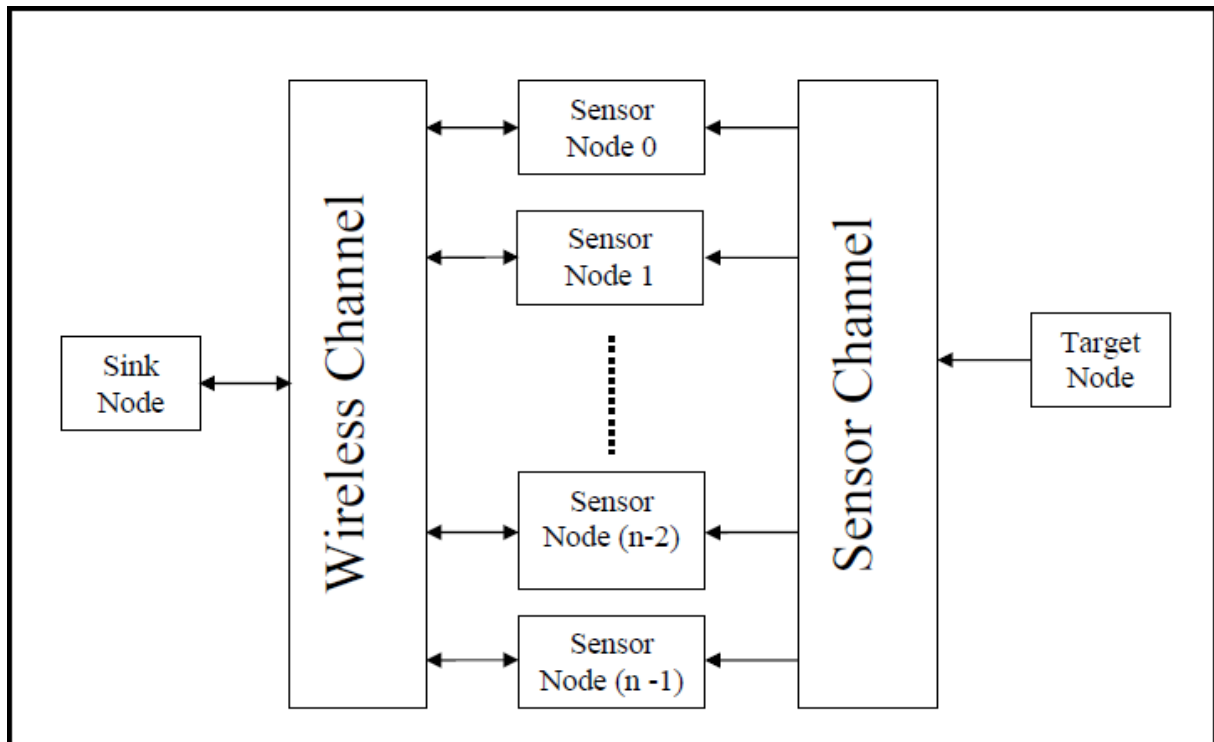


Figure V.2 Vue générale des trois types de nœuds dans J-Sim [53]

La Figure V.3 présente l'architecture interne d'un nœud sensor dans J-Sim. Le modèle d'énergie dans un nœud capteur comprend à la fois le composant producteur d'énergie (batterie), et les composants consommateurs d'énergie (CPU et la radio).

Les composants CPU et Radio définissent les modèles de la consommation énergétique du microcontrôleur et de l'antenne radio. Le composant CPU décrit en principe le coût de traitement des différents paquets reçus, tandis que le composant Radio décrit le coût de l'envoi et la réception des signaux et la mise en veille.

Le composant Battery définit le modèle de la batterie d'un nœud. Ce composant calcule la consommation énergétique due au traitement des instructions, envois et réceptions des signaux en se basant sur les modèles du CPU et de Radio. Un programmeur peut facilement créer ses propres modèles (CPU, Radio, Battery) et les insérer dans un scénario de simulation afin de les tester. A noter que seuls les nœuds sensor sont dotés d'une batterie [7].

La Figure V.3 montre ainsi que les composants des différentes couches d'un nœud sont séparés (Network layer, Mac Layer, Physical Layer). Ils communiquent entre eux via des ports. Les connexions (désignées par des flèches) entre les composants peuvent être unidirectionnelles ou bidirectionnelles. Une connexion unidirectionnelle d'un composant C1 vers un composant C2 signifie que les informations peuvent être envoyées dans une seule direction (de C1 à C2). D'autre part, une connexion bidirectionnelle signifie un échange d'information dans les deux directions.

Le composant "Wireless Channel" est le composant commun pour tous les nœuds sensor du réseau. Il fait communiquer tous les nœuds sensor entre eux. Chaque nœud sensor du réseau doit avoir une connexion, réalisée par le programmeur, avec ce composant. Le cas échéant, le nœud ne peut pas communiquer avec les autres nœuds.

Le composant "Sensor Channel" est responsable de la communication entre les nœuds target et les nœuds sensor. De même, chaque nœud (sensor ou target) doit avoir une connexion avec ce composant. Si, par exemple, un tel événement est créé par un nœud target, ce composant envoie l'information à tous les nœuds sensor qui entourent le target (la distribution des nœuds et leurs positions dans le champ de simulation sont définies dans le fichier TCL).

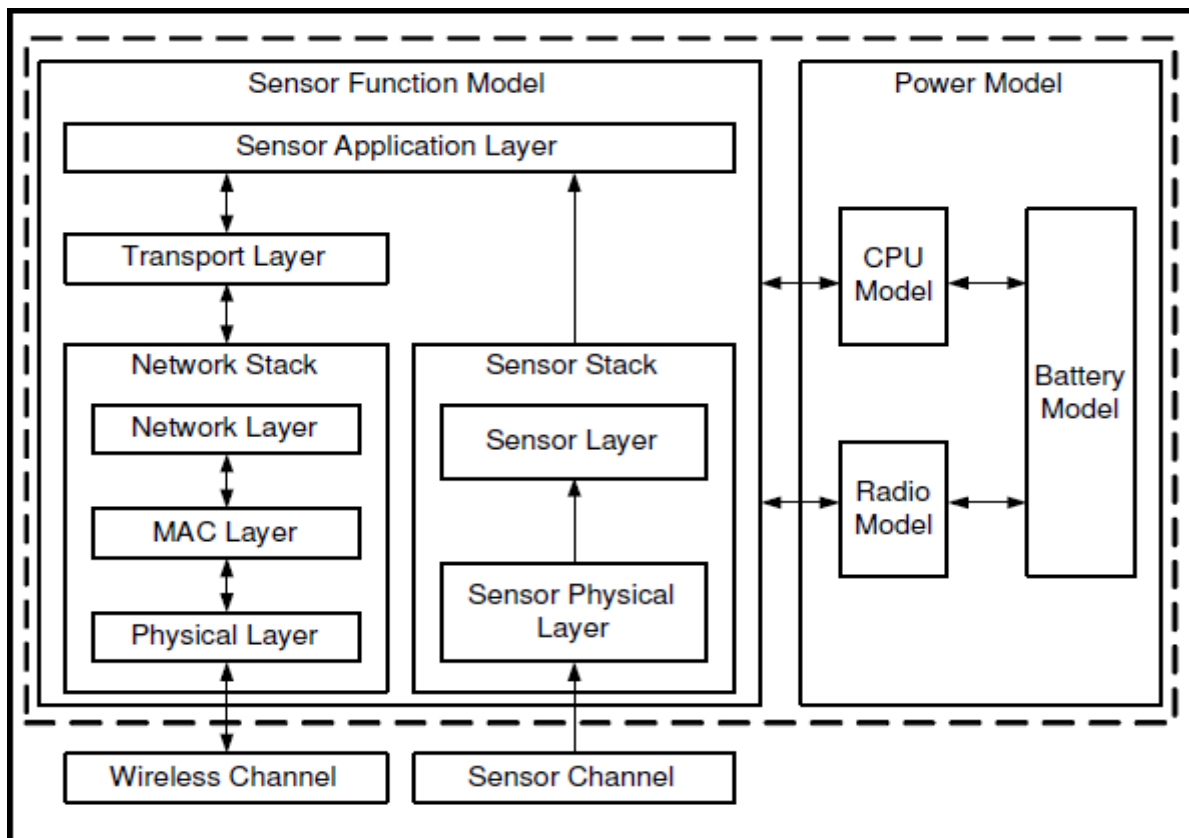


Figure V.3 Architecture interne du nœud sensor dans J-Sim [53]

V.2.2 Architecture détaillée de J-Sim

– Composant

L'entité de base dans l'architecture logicielle de J-Sim est le composant. Nous définissons une application en réalisant une composition de composants. Les composants dans J-Sim sont faiblement connectés. Ils communiquent entre eux en connectant leurs ports ensemble, et sont liés par des contrats. Les contrats indiquent la causalité des informations envoyées/reçues entre les composants, mais n'indiquent pas les composants qui participent à la communication.

– Port

Les ports dans J-Sim assurent la communication entre les différents composants. Un composant peut posséder plus d'un port, telle que l'interface de programmation entre un composant et son port est bien définie. Un composant peut être développé sans la présence d'autres composants (il se doit d'être autonome). En outre, le mécanisme réel de communication qu'un composant emploie pour communiquer avec le reste du monde est complètement caché par la notion de port.

– Contrat

La communication entre composants est décrite par les contrats. Le contrat impose les conditions sur les ports d'entrée/sortie des composants afin de les faire communiquer. Un contrat indique comment un initiateur (visiteur) et un réacteur (appelé) accomplissent une certaine communication.

Il décrit comment un composant répond aux données qui arrivent à chacun de ses ports (par exemple, comment le composant traite les données, certaines structures de données de mises à jour, et produit des sorties à certains ports).

– Langage utilisé pour définir un scénario de simulation

Bien que J-Sim soit écrit en Java, les scénarios de simulation ne sont pas décrits en Java car ce processus serait trop complexe. En outre, la combinaison des langages Java et TCL ne facilite pas cette tâche. Pour développer un grand projet, il peut devenir encombrant d'employer des commandes TCL/Java parce que les références des objets Java doivent être stockées dans des variables TCL afin d'y accéder. Afin de simplifier la syntaxe des simulations, un système appelé RUntime Virtual (RUV) est mis en place. Ce système s'appuie sur la similitude entre les systèmes composants et les systèmes de fichiers d'UNIX.

L'analogie entre un composant/port et un chemin de fichier permet d'accéder au composant de la même manière que l'on accède à un dossier dans un système de fichiers. D'ailleurs des commandes systèmes UNIX, telles que `ls`, `cd`, `pwd`, `mkdir`, `cp`, `mv`, et `rm` peuvent être utilisées pour manœuvrer des composants et des ports dans le système. Les ports ont une utilisation particulière pour le nommage : `a@b` (où `a` est le nom du port et `b` le groupe du port). En résumant, des composants peuvent être créés et ils peuvent être liés grâce à une interface de commande qui utilise la syntaxe des invites de commandes (shells).

V.2.3 Installation de J-Sim

Le fait que J-Sim soit développé en java lui offre l'avantage d'être un simulateur multiplateforme. Cela veut dire qu'il s'installe sur toute plateforme (Système d'exploitation) dotée d'une machine virtuelle Java (JVM : Java Virtual Machine). Pour cela avant d'installer J-Sim, on doit disposer de cette JVM. Le guide d'installation de J-Sim sous Windows et UNIX/Linux, est disponible sur le site officiel de J-Sim [51].

V.2.4 Intégration du code source du protocole dans J-Sim

Le code source du protocole à simuler, qui doit être écrit en langage Java est copié dans l'un des sous-répertoires du répertoire `/src` de J-Sim, à condition de respecter le principe de développement imposé par le langage Java (notion de packages).

Pour faciliter le développement du code source, il est nécessaire de choisir l'un des environnements de développements pour le langage Java. Il en existe plusieurs mais les plus populaires sont Eclipse IDE [54] et NetBean IDE [55]. Dans notre cas, nous avons choisi NetBeans IDE, pour sa facilité à créer les projets, qui nous a permis d'ailleurs d'installer J-Sim dans le répertoire `/project` de NetBeans IDE.

Avec l'ensemble J-Sim et NetBeans combinés, nous procédons comme suit pour simuler notre protocole:

Nous créons en premier lieu un projet sous NetBeans dans lequel nous installons J-Sim. Ensuite, nous passons à la création des packages (au sens Java) de notre projet dans le répertoire `/src/drcl/inet/sensosim`. Une fois le code source est sauvegardé, sous netBeans, nous le compilons par la commande « *ant compile* ».

V.3 Définition de quelques méthodes du protocole

Plusieurs méthodes ont été utilisées afin de réaliser les traitements de notre protocole, parmi eux:

- ***protected void _start ()***: cette méthode est exécutée au démarrage d'un nœud;
- ***protected void _stop ()***: cette méthode est exécutée à l'arrêt d'un nœud;
- ***public double getWeight(double energy, double energy_max, double degree ,double distance)***: cette méthode est évoquée par chaque nœud du réseau afin de calculer son poids selon la métrique utilisée par notre protocole;
- ***public synchronized void sendHello ()***: est exécutée par tous les nœuds du réseau afin que chacun d'eux puisse découvrir son voisinage;
- ***public synchronized void sendWeight()***: utilisée par tous les nœuds afin d'annoncer leurs poids à leurs voisinages;
- ***public synchronized void recvSensorEvent (Object data_)*** : exécutée à la réception d'un stimulus généré par un nœud target;
- ***public synchronized void recvSensorPacket (Object data_)***: exécutée lorsqu' un nœud du réseau reçoit un paquet data_;
- ***public synchronized void processOther (Object data_ , Port inPort_)***: exécutée par un composant lors de la réception d'un message via un port *inPort_* ;
- ***protected synchronized void timeout (Object data_)***:cette méthode est évoquée par un composant à l'expiration d'un timer relatif à un événement.

V.4 Exécution et simulation

Pour lancer la simulation, nous avons deux possibilités:

Charger le fichier de configuration de notre protocole via l'interface RUV ou utiliser la commande:

« *java drcl.ruv.System -u nom_fichier_de_configuration* ».

Le nom du fichier de configuration est de format *.tcl

V.5 Evaluation du protocole proposé

V.5.1 Environnement de simulation:

Pour effectuer nos simulations, nous avons utilisé une machine qui a les caractéristiques techniques suivantes:

- Système d'exploitation: Windows 7 Professionnel 64bits ;
- Processeur intel® core™ i5 2.40 Ghz;
- Mémoire physique: 04 Go;
- Simulateur J-Sim version 1.3 avec le patch;

V.5.2 Modèle énergétique:

Le modèle de la consommation d'énergie radio utilisé pour notre protocole est celui développé par Heinzelman et al pour le protocole LEACH [59].

L'énergie consommée pour émettre un paquet de k bits avec une distance d est donnée par la formule suivante:

$$E_{Tx}(k,d) = E_{Tx_elec}(k) + E_{Tx_amp}(k,d) = E_{elec} * k + amp * k * d^2$$

Pour recevoir un message de k bits, le récepteur consomme:

$$E_{Rx}(k) = E_{Rx_elec}(k) = k * E_{elec}$$

Où:

E_{elec} est l'énergie de transmission/réception électronique;

k est la taille d'un paquet;

amp représente le facteur d'amplification;

d est la distance entre l'émetteur et le récepteur;

E_{Tx_amp} est l'énergie d'amplification.

V.5.3 Paramètres de simulation

L'évaluation de notre protocole exige la prise en considération des spécificités des réseaux de capteurs sans fil pour qu'elle soit réalisable dans des conditions qui se rapprochent de la réalité. Pour cela, certains paramètres ont été pris en compte lors de la simulation. Ces paramètres sont sélectionnés en se basant sur des tests effectués par nous-mêmes tout en tenant compte de la capacité du simulateur ainsi que des travaux antérieurs pour des applications similaires.

Le tableau ci-dessous résume quelques paramètres utilisés :

Paramètres	Valeur
Surface de simulation (mxm)	100x200
Nombre de nœuds capteurs	Variable (25, 50, 75, 100, 125, 150)
Nombre de nœuds puits	1
Nombre de nœuds target	3
Placement des nœuds	aléatoire
Energie initiale (joules)	0.25
Puissance de transmission radio (watt)	0.096
Données étudiées	Variant entre 0 et 50
Période d'un round (second)	25
Temps de simulation (second)	500

Tableau V.1: Paramètres de simulation

Les capteurs utilisés dans la simulation sont considérés homogènes : possédant la même quantité d'énergie initiale, capacités de calcul et mémoire. De plus, l'énergie de la station de base est considérée comme illimitée. Ces capteurs sont déployés dans une surface (LxL) créant ainsi des topologies contenant un nombre variable de capteurs. La position des nœuds dans les topologies générées est réalisée d'une manière aléatoire.

V.5.4 Métrique d'évaluation de performance

Le choix des métriques pour évaluer un protocole dépend de l'objectif de l'évaluation et des relations entre ces métriques. Pour quantifier les performances du protocole proposé, deux métriques sont sélectionnées: moyenne d'énergie consommée et le délai moyen de transmission des paquets de données à la station de base.

– Moyenne d'énergie consommée

L'un des principaux critères de performances pour un capteur est l'utilisation efficace de son énergie. Une fois qu'elle sera épuisée, il n'y a pas d'autre moyen de recharger à nouveau sa batterie. La moyenne d'énergie consommée est obtenue par la somme des joules de tous les nœuds sur la taille du réseau.

Mathématiquement, on peut l'exprimer ainsi :

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

Où :

E_i représente l'énergie consommée pour un capteur i . Elle représente la différence entre l'énergie initiale et son énergie résiduelle.

n est le nombre de capteurs dans le réseau.

– **Délai moyen**

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

$$Délai = \frac{\sum_{i=1}^n (Tr - Te)}{n}$$

Où:

Tr représente le temps d'arrivée d'un paquet de données à la station de base;

Te est le temps d'émission du paquet par le nœud cluster-head;

n est le nombre total de paquets de données reçus par la station de base.

La moyenne d'énergie consommée ainsi que le délai moyen de transmission sont étudiés sur différentes topologies (25, 50, 75, 100, 125, 150 nœuds) durant une durée de 500 seconds ce qui est équivalent à 20 rounds.

Le protocole a été simulé sur les mêmes topologies avant et après application de la technique d'agrégation.

V.5.5 Interprétation des résultats

– Moyenne d'énergie consommée:

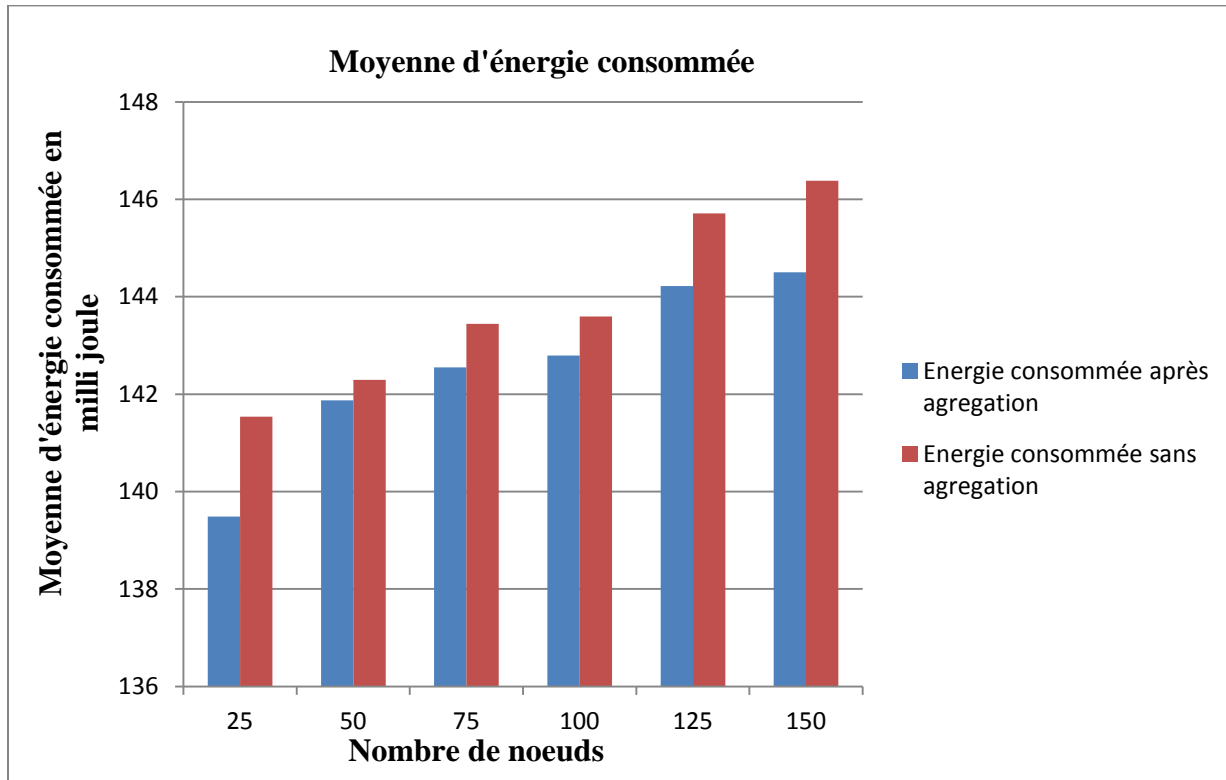


Figure V.4: Moyenne d'énergie consommée

La figure ci-dessus illustre la moyenne d'énergie consommée en fonction du nombre de nœuds du réseau. Nous observons dans toutes les topologies que l'énergie consommée par notre protocole lors de l'application de la fonction d'agrégation est plus petite que celle consommée dans le cas sans agrégation. Ceci est dû à la technique d'agrégation qui a permis une réduction du nombre de paquets transmis dans le réseau et donc gain en terme d'énergie consommée lors des communications.

Ainsi, nous remarquons que l'énergie évolue presque de la même manière dans les deux cas; ceci est dû à l'efficacité de la technique de clustering appliquée qui permet une meilleure rotation du rôle de cluster-head.

– Délai moyen:

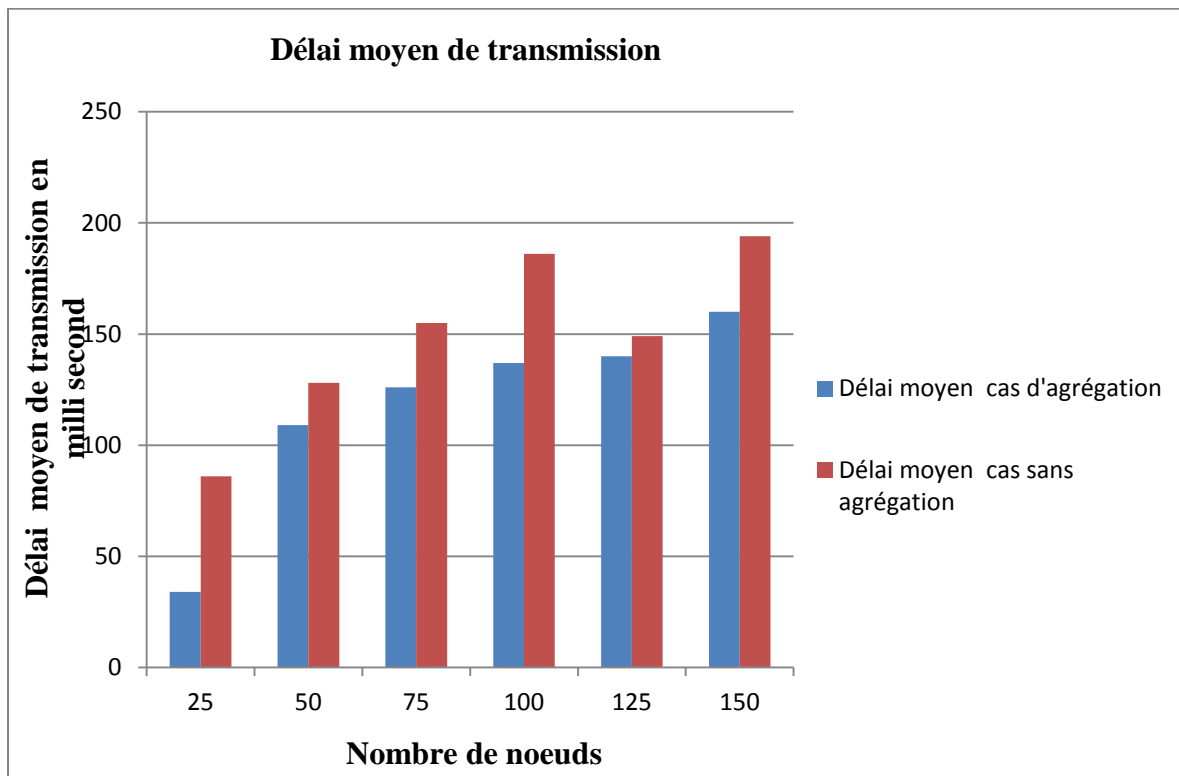


Figure V.5. Délai moyen de transmission des paquets de données

Comme le montre la figure précédente, le protocole dans les deux cas avec et sans agrégation expriment le délai moyen de cheminement des paquets de données à la station de base avec un léger avantage pour le cas d'agrégation. Cet avantage marqué revient au nombre réduit de paquets reçus avec succès par la station de base contrairement au cas sans agrégation qui acheminent un nombre plus important de paquets à cette entité de collecte. Ces résultats confirment l'efficacité du mécanisme d'agrégation utilisé.

Conclusion

Dans ce chapitre, nous avons présenté l'environnement de simulation utilisé J-Sim ainsi que son fonctionnement. Cet environnement a permis l'implémentation de notre approche basée clustering sans et avec agrégation. Les résultats de simulation effectués sous J-Sim ont montré que l'approche avec agrégation de données améliore la consommation d'énergie par rapport à celle sans agrégation, de même pour le délai moyen de transmission. Ceci est dû essentiellement à l'efficacité de la technique d'agrégation qui a permis une réduction du nombre des transmissions effectuées au sein du réseau et donc préservation de la ressource énergétique et réduction des délais moyens de cheminement des données à la station de base.

CONCLUSION GENERALE

CONCLUSION GENERALE

Les réseaux de capteurs sans fil constituent une voie de recherche très importante à explorer, ils sont adaptés aux divers domaines de notre quotidien. En effet, une multiplicité de travaux a été mise en place pour remédier aux limites et améliorer leurs performances.

L'un des principaux problèmes que nous pouvions rencontrer dans ce type de réseau est la consommation importante de la ressource énergétique, causée par l'existence de données redondantes. Cette surconsommation est liée à la nature de ces réseaux, principalement à leur mode de déploiement.

L'existence ou l'échange de données redondantes ou fortement corrélées dans le réseau influe négativement sur ses différentes ressources: la bande passante et l'énergie qui représente une ressource critique dans les réseaux de capteurs sans fil. Il est donc nécessaire de développer et mettre en place des protocoles permettant de conserver cette ressource et assurer le bon fonctionnement du réseau entier.

Dans ce projet, nous avons mis en place un protocole, appliquant une technique d'agrégation de données sur un réseau structuré en clusters par un mécanisme de clustering bien défini, à l'aide d'un simulateur dédiés aux réseaux de capteurs sans fil: J-Sim.

Travailler sur un domaine de recherche comme les réseaux de capteurs, exige la prise en compte des différentes contraintes qui lui sont liées. Ce qui présente des difficultés lors de la phase d'implémentation. Avec la motivation que nous avons eue pour mener ce travail, nous avons finalement pu atteindre les objectifs tracés et nous espérons que ce projet sera une grande inspiration pour poursuivre dans ce domaine de recherche.

Après avoir réalisé notre travail, des perspectives ont fait apparition. En effet, nous envisageons comparer les performances de notre protocole avec d'autres protocoles d'agrégation de données existants. Nous souhaitons, de même, continuer à l'améliorer en adoptant des techniques permettant de fixer de manière efficace les différents paramètres, et surtout prendre en compte d'autres métriques d'évaluation de son bon fonctionnement. Aussi, nous prévoyons continuer à découvrir et à explorer l'outil de simulation J-Sim qui est récent et disposant d'une bibliothèque très riche. Nous envisageons tous cela dans nos futurs travaux de recherche qu'ils soient d'ordre individuels, professionnels ou académiques.

REFERENCES
BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

- [1] **Vernon S. Somerset**, “Intelligent and Biosensors, Edited by Vernon S. Somerset”, Intech, January 2010.
- [2] **Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam et Erdal Cayirci** “A Survey on Sensor Networks”, IEEE Communications Magazine, August 2002.
- [3] **YACINE CHALLAL**, Réseaux de Capteurs Sans Fils, Version 1 *SIT60*, 2008.
<http://moodle.utc.fr>
- [4] **Holger Karl, Andreas Willig**, PROTOCOLS AND ARCHITECTURES FOR WIRELESS SENSOR NETWORKS, 2005 John Wiley & Sons, Ltd. ISBN: 0-470-09510-5
- [5] **Bhaskar Krishnamachari**, Networking Wireless Sensors, CAMBRIDGE UNIVERSITY PRESS 2005.
- [6] **Jun Zheng, Abbas Jamalipour**, WIRELESS SENSOR NETWORKS A Networking Perspective, A JOHN WILEY & SONS, INC., PUBLICATION 2009 by IEEE.
- [7] **Kamal BEYDOUN**, CONCEPTION D’UN PROTOCOLE DE ROUTAGE HIERARCHIQUE POUR LES RESEAUX DE CAPTEURS, L’U.F.R DES SCIENCES ET TECHNIQUES DE L’UNIVERSITE DE FRANCHE-COMTE, Année 2009.
- [8] **Wassim MASRI**, Dérivation d'exigences de Qualité de Service dans les Réseaux de Capteurs Sans Fil basés sur TDMA, Ecole doctorale : Mathématiques, Informatique et Télécommunications de Toulouse, 2009
- [9] **JAMAL N. AL-KARAKI, AHMED E. KAMAL**, ROUTING TECHNIQUES IN WIRELESS SENSOR NETWORKS: A SURVEY, IEEE Wireless Communications December 2004.
- [10] **Karl Baumgartner**, Réseaux de capteurs sans fil, NOUVELLES DU GITI, 2006
- [11] **Eiko Yoneki, Jean Bacon**, “A survey of Wireless Sensor Network technologies: research trends and middleware’s role”, university of Cambridge, September 2005.
- [12] **Lyes KHELLADI & Nadjib BADACHE**, "Les réseaux de capteurs : état de l'art ", rapport de recherche, Laboratoire des Systèmes Informatique (LSI) USTHB, 2004
- [13] **BECHKIT Walid**, Un nouveau protocole de routage avec conservation d’énergie dans les réseaux de capteurs sans fil, Ecole nationale Supérieure d’Informatique (E.S.I), 2009
- [14] **M. ACHIR**, Technologies basse consommation pour les réseaux ad-hoc, Thèse doctorat, Institut National Polytechnique de Grenoble, juillet 2005.
- [15] **Qinghua Wang et Ilangko Balasingham**, “Wireless Sensor Networks – An Introduction”, Wireless Sensor Networks: Application- Centric Design, Geoff V Merrett and Yen Kheng Tan (Ed.), ISBN: 978-953-307-321-7 InTech, 2010

REFERENCES BIBLIOGRAPHIQUES

- [16] **Ben L. Titzer et Jens Palsberg** , “Nonintrusive Precision Instrumentation of Microcontroller Software”, ACM, New York, NY, ETATS-UNIS, Volume 40 Issue7, Pages 59-68, July 2005.
- [17] **Alcides Montoya, Diana Carolina Restrepo et Demetrio Arturo Ovalle**, "Artificial Intelligence for Wireless Sensor Networks Enhancement”, InTech, 2010.
- [18] **TWAGIRAYEZU Jean Bosco**, "Proposition d'un nouveau protocole de routage avec agrégation des données pour contrôler la congestion dans un RCSF", Mémoire de Master, UMMTO, 2011.
- [19] **Shio Kumar Singh , M P Singh , and D K Singh**, Routing Protocols in Wireless Sensor Networks –A Survey, International Journal of Computer Science & Engineering Survey (IJCSSES) Vol.1, No.2, November 2010.
- [20] **Stanislava Soro , Wendi B. Heinzelman**, Cluster head election techniques for coverage preservation in wireless sensor networks, Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627, United States, 2008.
- [21] **Sanghita Bhattacharya, Subhansu Bandyopadhyay**, A Dynamic Energy Efficient Multi Hop Routing Technique Using Energy Aware Clustering In Wireless Sensor Network, IEEE 2011.
- [22] **D. J. Dechene, A. El Jardali, M. Luccini, and A. Sauer**, A Survey of Clustering Algorithms for Wireless Sensor Networks, Department of Electrical and Computer Engineering, The University Of Western Ontario London, Ontario, Canada, 2011.
- [23] **Ameer Ahmed Abbasi, Mohamed Younis**, A survey on clustering algorithms for wireless sensor networks, Computer Communications 30 (2007) 2826–2841.
- [24] **Khaled BOUCHARAKOUR**, Routage hiérarchique sur les réseaux de capteurs sans fil: Protocole KhLCH (K-hop Layered Clustering Hierarchy), thèse doctorat, Ecole Doctorale Sciences et Technologies de l'Information et de la Communication (STIC), 2012.
- [25] **Badreddine GUIZANI**, ALGORITHME DE CLUSTERISATION ET PROTOCOLES DE ROUTAGE DANS LES RÉSEAUX AD HOC, Université de Technologie de Belfort-Montbéliard École Doctorale Sciences pour l'Ingénieur et Microtechniques, 2012.
- [26] **Cheick Tidjane KONE**, Conception de l'architecture d'un réseau de capteurs sans fil de grande dimension, Ecole Doctorale : IAEM Lorraine, Département de Formation Doctorale : Automatique et Production Automatisée, 2011.

REFERENCES BIBLIOGRAPHIQUES

- [27] **Ossama Younis, Marwan Krunz, and Srinivasan Ramasubramanian**, Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges, University of Arizona, 2006.
- [28] **Raluca Marin-Perianu**, Wireless Sensor Networks in Motion Clustering Algorithms for Service Discovery and Provisioning, University of Twente, 2008.
- [29] **Luis Javier García Villalba, Ana Lucila Sandoval Orozco, Alicia Triviño Cabrera and Cláudia Jacy Barenco Abbas**, Routing Protocols in Wireless Sensor Networks, *Sensors* 2009, 9, 8399-8421.
- [30] **XUE Yong, AGUILAR Andres, GONZALEZ Andres, BARROUX Mickael**; Agrégation de données dans les réseaux de capteurs, Projet SR04 Rapport final, UTC, Automne 2010.
- [31] **M. Chatterjee, S. K. Das, and D.Turgut**. Wca: A weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing*, 5(2):193–204, 2002.
- [32] **Stefano Basagni**, Distributed Clustering for Ad Hoc Networks, Center for Advanced Telecommunications Systems and Services (CATSS) Erik Jonsson School of Engineering and Computer Science, The University of Texas at Dallas.
- [33] **Mohamed Tahar DARBOUCHE**, L'utilisation des heuristiques pour la résolution des problèmes de tournées de véhicules, Université Mentouri, Constantine, Algérie, 2009
- [34] **Alan D. Amis Ravi Prakash Thai H.P. Vuong Dung T. Huynh**, Max-Min D-Cluster Formation in Wireless Ad Hoc Networks, Department of Computer Science University of Texas at Dallas Richardson, Texas 75083-0688, 2011.
- [35] **Vaibhav Pandey, Amarjeet Kaur and Narottam Chand**, A review on data aggregation techniques in wireless sensor network, Department of Computer Science & Engineering, National Institute of Technology, Hamirpur, HP, India, *Journal of Electronic and Electrical Engineering* ISSN: 0976–8106 & E-ISSN: 0976–8114, Vol. 1, Issue 2, 2010, pp-01-08.
- [36] **Bhaskar Krishnamachari, Deborah Estrin and Stephen Wicker**, The Impact of Data Aggregation in Wireless Sensor Networks, 2002.
- [37] **Kiran Maraiya, Kamal Kant, Nitin Gupta**, Wireless Sensor Network: A Review on Data Aggregation, *International Journal of Scientific & Engineering Research* Volume 2, Issue 4, April -2011.
- [38] **Ramesh Rajagopalan, Pramod K. Varshney**, Data aggregation techniques in sensor networks: A survey, Syracuse University, Department of Electrical Engineering and Computer Science, 1-1-2006.

REFERENCES BIBLIOGRAPHIQUES

- [39] **Philip P. Mathew**, Data Aggregation Techniques in sensor networks, Based on a paper by Ramesh Rajagopalan, Pramod K Varshney IEEE communications Survey, 2009.
- [40] **M. Thangaraj and P. Punitha Ponnalar**, A Survey on data aggregation techniques in wireless sensor networks, International Journal of Research and Reviews in Wireless Sensor Networks (IJRRWSN) Vol. 1, No. 3, September 2011, ISSN: 2047-0037.
- [41] **Jie Feng**, Performance of Data Aggregation for Wireless Sensor Networks, A Thesis Submitted to the College of Graduate Studies and Research in Partial Fulfillment of the Requirements for the degree of Doctor of Philosophy in the Department of Computer Science University of Saskatchewan Saskatoon , June/2010.
- [42] **George Kollios, John Byers, Jeffrey Considine, Marios Hadjieleftheriou, and Feifei Li**, Robust Aggregation in Sensor Networks, Computer Science Dept., Boston University, 2009.
- [43] Choon-Sung Nam, Kyung-Soo Jang and Dong-Ryeol Shin, A Cluster Head Election Method for Equal Cluster Size in Wireless Sensor Network, Sungkyunkwan University and Kyungin women's college, 2012.
- [44] **K.Ramesh and Dr. K.Somasundaram**, A COMPARATIVE STUDY OF CLUSTERHEAD SELECTION ALGORITHMS IN WIRELESS SENSOR NETWORKS, International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.4, November 2011.
- [45] **EDUARDO F. NAKAMURA, ANTONIO A. F. LOUREIRO and ALEJANDRO C. FRERY**, Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications, ACM Computing Surveys, Vol. 39, No. 3, Article 9, Publication date: August 2007.
- [46] **Hilal Houssain**, Elliptic Curve Cryptography Algorithms Resistant Against Power Analysis Attacks on Resource Constrained Devices, ÉCOLE DOCTORALE SCIENCES POUR L'INGÉNIEUR DE CLERMONT-FERRAND, 2012.
- [47] **Sadaf TANVIR**, Localisation dans les Réseaux de Capteurs - Protocoles de Communication et Efficacité Energetique, l'Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, UNIVERSITE DE GRENOBLE, 2010.
- [48] **Maulin Patel, R. Chandrasekaran and S.Venkatesan**, Efficient Minimum-cost Bandwidth-constrained Routing in Wireless Sensor Networks, Telecommunication Engineering Program, Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, Texas 75083-0688.

REFERENCES BIBLIOGRAPHIQUES

- [49] **Mrs. Saba Siraj, Mr. Ajay Kumar Gupta, Mrs Rinku-Badgujar**, Network Simulation Tools Survey, International Journal of Advanced Research in Computer and Communication Engineering Vol. 1, Issue 4, June 2012.
- [50] **Marek Małowidzki**, NETWORK SIMULATORS: A DEVELOPER'S PERSPECTIVE, Military Communication Institute 05-130 Zegrze, Poland.
- [51] <http://sites.google.com/site/jsimofficial/>
- [52] Tcl Developer Site. <http://www.tcl.tk/>.
- [53] **Ahmed Sobeih, Jennifer C. Hou**, A Simulation Framework for Sensor Networks in J-Sim, Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801.
- [54] <http://www.eclipse.org/>
- [55] <http://netbeans.org/>
- [56] **Gianni A. Di Caro**, « Analysis of simulation environments for mobile ad hoc Networks », Technical Report No. IDSIA-24-03, Dalle Molle Institute for Artificial Intelligence, Galleria 2, 6928 Manno, Switzerland, 2003.
- [57] **Emmanuel CONCHON**, DÉFINITION ET MISE EN OEUVRE D'UNE SOLUTION D'ÉMULATION DE RÉSEAUX SANS FIL, École doctorale: Informatique et Télécommunications, Institut National Polytechnique De Toulouse, 2006.
- [58] **Abdallah Makhoul**, Réseaux de capteurs : localisation, couverture et fusion de données, l'École Doctorale Sciences Pour l'Ingénieur et Microtechniques (SPIM), 2008.
- [59] **W. Heinzelman, A. Chandrakasan, and H. Balakrishnan**. "Energy-efficient communication protocol for wireless sensor networks". Proceedings of the IEEE Hawaii International Conference on System Sciences, pp. 3005-3014, January 2000.