

République Algérienne Démocratique et Populaire
Ministère de L'enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaia
Faculté des Sciences Exactes
Département Informatique



Mémoire de Fin de Cycle

En vue de l'obtention du diplôme Master recherche en
Informatique

Spécialité : Réseaux et Systèmes Distribués

THÈME

Adaptation des architectures P2P pour les
applications des réseaux de capteurs sans fil

Réalisé par :

Promoteur : M. AMAD Mourad.

Mme Maouche Zineb.

Mme Djermoune Yasmina.

Président Mr : KHENOUC H
Examineur 1 Mr : DJEBARIN
Examineur 2 Mr : CHEKLAT M

U. A/Mira Béjaia.
U. A/Mira Béjaia.
U. A/Mira Béjaia.

PROMOTION 2017

Remerciement

Nos remerciements les plus sincères vont au bon Dieu qui nous a procuré le courage, la volonté et surtout la santé pour pouvoir réaliser ce travail.

Nous tenons à exprimer nos vifs remerciements à notre Promoteur Mr M. AMAD et Co-promoteur Mr M. ATMANI, pour nous avoir proposé ce thème et tous leurs conseils et leurs orientations tout le long de notre travail.

Nous remercions les membres de jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer ce travail.

Notre profonde gratitude s'adresse à tous nos enseignants qui ont contribué à notre formation.

Nous remercions vivement toutes les personnes qui nous ont aidés à élaborer et réaliser ce mémoire, ainsi que tous ceux qui nous ont assistés de près ou de loin à l'accomplir.

Dédicace

Je dédie ce modeste travail :

A mes très chers parent

Mon très cher père pour sa patience et tous ses efforts

*A ma mère pour m'avoir épaulé, encouragé à reprendre les
études et motivé dans les moments les plus difficiles*

A mes soeurs :TILA, LINDA , TOUTA ,FOUZIA,NADJAT et MAYA

*A mes frères :YASSINE, NOUNOUR, LOTFI et TOUFIK pour leur
encouragements*

A mon marie KARIM pour son

soutien, sa compréhension et ces encouragements

A toute ma famille : DJERMOUNE ,MOKRANE , IDIR

*A notre encadreur Monsieur M. AMAD et co-encadreur monsieur
ATMANI pour*

*Leur excellence accompagnement et la confiance qu'il nous on
Accordé.*

A ma binôme ZINA avec qui j'ai eu l'honneur de les finir mes études.

*A tous mes professeurs ainsi que tous les étudiants de la
Promotion RSD-2017, qui nous ont accueillis si généreusement et
aidés tout au long années universitaire*

*A mes amies : SIHAM , MANEL, KARIMA ,NASSIMA,
MOUNIA.....*

Ainsi qu'à tous ceux qui ont participé de loin ou de près à réaliser ce travail

Dédicaces

Je dédie ce modeste travail:

•A mes chers parents que dieu le tout puissant leurs accorde une place dans son vaste paradis

•A mon très cher mari ilies, qui m'a toujours soutenu et encouragé

•A mes adorables enfants, mes anges, Moussa et Anaisse

•A ma chère belle-mère Malika qui m'a toujours soutenue et aidée

le long de ce travail

•A mes chères sœurs et belle sœurs

•A mes chers frères Hichem et Kadir

•A mon cher grand frère (Sidi Rachid), sa femme et leurs adorables enfants

•A toute ma famille

•A mes fidèles amis (es)

•A ma chère binôme Yasmîna, qui a partagé avec moi ce modeste travail

•A toute la promotion informatique ,

•A tous ceux que j'aime...

•Ainsi qu'à tous ceux qui ont participé de loin ou de près à réaliser ce travail

Zina.

Table des matières

Table des matières

Liste des figures

Liste des Tableaux

Listes des algorithmes

Glossaire

Introduction générale

1 Généralités sur les réseaux de capteurs sans fil.....	1
1.1 Introduction	3
1.2 Capteurs sans fil	3
1.3 Architecture de capteur sans fil	4
1.3.1 Unité de traitement (Mote, processeur, RAM et Flash)	5
1.3.2 Unité de transmission	5
1.3.3 Unités de captage(LED, interface, capteur)	5
1.3.4 Unités de contrôle d'énergie	6
1.4 Applications des RCSF	6
1.4.1 Découvertes de catastrophes naturelles	6
1.4.2 Détection d'intrusions	6
1.4.3 Applications métier	6
1.4.4 Contrôle de la pollution	6
1.4.5 Agriculture	7
1.4.6 Surveillance médicale	7
1.4.7 Contrôle d'édifices	7
1.5 Caractéristiques des réseaux des capteurs	8
1.5.1 Systèmes d'exploitation pour les réseaux de capteurs	8
1.5.1.1 TinyOS	8
1.5.1.2 MOS	8
1.5.1.3 SOS	8
1.5.2 Mode de consommation d'énergie	9
1.6 Ordonnancement de réseaux de capteur	9
1.7 Déploiement des capteurs	10
1.7.1 Déploiement déterministe	10
1.7.2 Déploiement aléatoires	10
1.8 Contraintes de conception des RCSF	11
1.8.1 Tolérance aux fautes	11
1.8.2 Echelle	11
1.8.3 Coûts de production	11
1.8.4 Environnement	11

1.8.5	Contraintes matérielles	11
1.8.6	Topologie de réseau	12
1.9	Conclusion	12
2	 Routage dans les réseaux de Capteurs sans fil	13
2.1	Introduction	13
2.2	Les principaux protocoles de routage dans les RCSF	14
2.2.1	Les Protocoles hiérarchiques	15
2.2.1.1	Protocole de routage "LEACH "	15
2.2.1.2	Protocoles de routage "TEEN et APTEEN"	15
2.2.2	protocoles de routage basés sur la localisation	16
2.2.2.1	protocole de routage " MECN "	16
2.2.2.2	protocole de routage " GAF "	17
2.2.3	protocoles de routage 'data-centric'	17
2.2.3.1	protocole de routage " SPIN "	18
2.2.3.2	Diffusion dirigée	19
2.3	Classification des protocoles de routages dans les réseaux de capteurs sans fil	20
2.4	Contraintes de routage dans les réseaux de capteurs sans fil	21
2.5	Routage dans les réseaux de capteurs à base d'architecture P2P	22
2.5.1	Paradigme de communication pair-à-pair	22
2.5.1.1	Exemple de Réseaux P2P structurés	22
2.6	Les points communs entre les deux types de réseau RCSF ET P2P	26
2.7	Travaux relatifs aux RCSF basés sur des architectureP2P	27
2.7.1	Architecture sans puits basée sur une surcouche pair à pair	27
2.7.1.1	Principe de fonctionnement	27
2.7.1.2	Présentation de l'architecture	28
2.7.1.3	Schéma global de l'architecture	28
2.7.2	P2P-RPL (Point To Point Routing Protocol for Low Power Lossy Networks)	29
2.7.2.1	Principe de fonctionnement	32
2.7.2.2	la mise en œuvre	32
2.7.2.3	Conclusion	32
3	 Proposition d'une Adaptation d'une architecture P2P pour une Appli- cation de Réseaux de Capteurs	34
3.1	Introduction	34
3.2	Hypothèses et exigences	35
3.2.1	Hypothèses de travail	35
3.2.2	Exigences	36
3.3	Présentation de l'architecture	36
3.4	Les différents algorithmes de l'architectureproposée basée sur CAN adapté au RCSF	38
3.4.1	Algorithme d'initiation	38
3.4.2	Algorithme de processus de routage	39
3.4.3	Algorithme d'insertion d'un nœud	42
3.4.4	Algorithme de suppression d'un nœud	43
3.5	Conclusion	43

4	Conception et réalisation	44
4.1	Introduction	44
4.2	Modélisation de l'application avec l'UML	44
4.2.1	Diagramme de cas d'utilisation	45
4.2.2	Diagramme de séquence	46
4.2.3	Diagramme de classes	47
4.3	Réalisation	48
4.3.1	Outils de développement	48
4.3.2	Interface de l'application	48
	Conclusion.....	52
	Bibliographie.....	53
	Résumé	

LISTE DES FIGURES

1.1	Capteur sans fil [19].	4
1.2	Architecture d'un réseau de capteurs [23].	4
1.3	Architecture fonctionnelle d'un capteur sans fil [9].	5
1.4	Applications des RCSF [15].	8
1.5	Trois approches algorithmiques pour l'ordonnancement d'activité [11].	10
2.1	Les principaux protocoles de routages dans les RCSF [5].	14
2.2	Topologie hiérarchique [2].	15
2.3	Routage data-centric [12].	18
2.4	Les phases de communication du protocole SPIN [14].	19
2.5	Les phases de communication du protocole Directed Diffision [14].	20
2.6	Classification des protocoles de routages dans les réseaux de capteurs sans fil [22].	21
2.7	Exemple de réseau CAN d'espace2-d [1].	23
2.8	Routage CAN [1].	24
2.9	Comment un nœud se connecte au réseau CAN [1].	25
2.10	Réseaux réels et surcouche pair à pair [21].	28
2.11	Architecture sans puits basée sur une surcouche pair à pair pour réseau de capteurs sans fil, (a) réseau réel des dispositifs ; (b) surcouche pair à pair [21].	29
2.12	Destination Oriented Directed Acyclic Graph(DODAG) rooted at router S [8].	30
2.13	Communication between router B and routerA through the sink, provided by RPL [8].	31
3.1	Réseau réel et surcouche pair à pair (CAN adapté au RCSF) [20].	37
3.2	Exemple de l'architecture proposée basée sur CAN (nombre de Cluster-Heads égale à11 et nombre de Cluster-Heads connectés égale à 2) [20].	38
3.3	Etape 1 : transmettre l'information au Cluster-Head (1) [20].	40
3.4	Etape 2 : Choisir aléatoirement leCluster-Head connectée (11) [20].	41
3.5	Etape3 : les étapes à suivre pour router l'information au CCH (11) [20].	41
3.6	Etape 4 : Envoie de l'information au Sink [20].	42
4.1	Diagramme de cas d'utilisation	45
4.2	Diagramme de séquence.	46
4.3	Diagramme de classe.	47

4.4	Fenêtre d'accueil.	48
4.5	Interface de la boîte de dialogue.	49
4.6	Génération aléatoire des identifiants et des énergies des nœuds	49
4.7	Construction de réseau.	50
4.8	Fenêtre de la recherche.	50

LISTE DES TABLEAUX

Tab.1.1 Notations utilisées dans différents algorithmes.....	6
--	---

LISTE DES ALGORITHMES

Algorithme 3.1	Algorithme d'initiation.....	17
Algorithme 3.2	Algorithme de processus de routage.....	18
Algorithme 3.3	Algorithme d'insertion d'un nœud.....	19
Algorithme 3.4	Algorithme de suppression d'un nœud.....	28

GLOSSAIRE

- **RCSF** : Réseaux de capteur sans fils
- **GPS** : système de positionnement global
- **TinyOS** : tiny operating system
- **DSDV**: Destination Sequenced Distance Vector
- **OLSR**: Optimized Link State Routing
- **DSR**: Dynamic Source Routing
- **AODV**: Ad hoc On-Demand Distance Vector
- **LEACH**: Low Energy Adaptive Clustering Hierarchy
- **TEEN et APTEEN**: Threshold sensitive Energy Efficient sensor Network protocol
- **MECN**: Minimum Energie Communication Network
- **GAF**: Geographic Adaptive Fidelity
- **SPIN**: Sensor Protocols for Information via Negotiation
- **SPEED**: Destination Sequenced Distance Vector
- **DD**: Directed- Diffusion
- **RR**: Rumor Routing
- **SAR** : Rumor Routing
- **P2P** : pair-à-pai
- **CAN** : Est un réseau structuré où les clés sont associées à des coordonnées dans un
espace de dimension d , et chaque pair est responsable d'une zone de clés.
- **DHT** : Table de hachage distribué (*Distributed Hash Table*), structure décentralisé de la table de hachage classique.
- **P2P-RPL** : Point To Point Routing Protocol for Low Power Lossy Networks
- **DODAG** :Destination Oriented Directed Acyclic Graph
- **DIO** :DODAG Information Object
- **DIO** :DODAG Information Object

Introduction générale

Les progrès technologiques dans le domaine de l'électronique et le domaine des communications sans fil ont donné naissance à la création d'équipements miniaturisés à faible coût, appelés nœuds capteurs. Ces derniers peuvent communiquer via des ondes hertziennes (la radio) et collaborer entre eux pour former un réseau de capteurs.

Les réseaux de capteurs sans fil (RCSFs) sont généralement composés d'un grand nombre de nœuds capteurs qui sont déployés aléatoirement dans une zone d'intérêt pour surveiller un phénomène donné. Ces capteurs collaborent entre eux et remontent l'information de nœud en nœud selon le mode multi-sauts et via une connexion sans fil jusqu'à l'aboutissement à la station de base. Or, ce schéma de communication est coûteux en termes d'énergie puisque tout nœud joue le rôle d'un nœud relai puis qu'il renvoie l'information reçue à ses voisins. Ce processus de communication se répète jusqu'à l'arrivée de l'information à la station de base. De ce fait, pour remédier à cet inconvénient, plusieurs protocoles de routage ont été proposés dans la littérature pour communiquer l'information rapidement à un moindre coût à la station de base.

Un des plus importants des thèmes critiques dans les réseaux de capteurs sans fil est l'optimisation de consommation d'énergie en raison du fait que la vie globale de réseau dépend largement de la vie des batteries des capteurs.

En outre, il est cher de remplacer les ressources énergétiques des capteurs ou même impossible. Par conséquent, on lui recommande de développer de nouveaux mécanismes et techniques pour surmonter cette limitation d'énergie dans les WSNs. Les nœuds de capteur peuvent envoyer, recevoir, ou transmettre par relais également les données. Ceci mène à comprendre que des nœuds sont liés par l'intermédiaire des communications peer-to-peer dans lesquels les pair jouant à la fois le rôle de client et de serveur. Plusieurs variétés de protocoles P2P ont été développées pour chaque catégorie de systèmes P2P.

Notre objectif est d'adapter le protocole CAN pour les réseaux de capteurs sans fil prenant en considération l'optimisation de la consommation d'énergie. Le but visé est d'exploiter Les avantages de DHTs dans les WSNs tel que la bonne gestion et le développement totalement décentralisés, extensibilités et auto-organisation des applications.

Afin de mener à bien notre travail, nous l'avons scindé en quatre chapitres : Chacun de

ces chapitres aborde des points spécifiques, et il est structuré comme suit :

Chapitre 1 : présentation d'une étude générale des réseaux de capteurs en en illustrant leurs caractéristiques et leurs spécificités.

Chapitre 2 : description d'un état de l'art sur les protocoles de routage dans les réseaux de capteurs, les différentes classifications qui existent d'autre pars présentation des systèmes pair-à-pair, ainsi les différents concepts liés à ces réseaux, afin d'exploiter les points communs entre les deux architectures et la mis en point sur les avantages de ce dernier pour réaliser une amélioration dans un réseau WSN. Chapitre 3 : description d'une solution proposée apportant des modifications sur le protocole CAN pour l'adapter aux réseaux de capteurs sans fil prenant en considération l'optimisation de la consommation d'énergie.

Chapitre 4 : description de la conception et de la réalisation de notre application qui consiste en l'adaptation du protocole CAN pour les réseaux de capteurs sans fil prenant en considération l'optimisation de la consommation d'énergie.

Une conclusion générale viendra clore ce travail en résumant les grands points qui ont été abordés dans ce mémoire, ainsi que les perspectives que l'on souhaite accomplir prochainement.

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

1.1 Introduction

Les avancées technologiques récentes confortent la présence de l'informatique et de l'électronique au cœur du monde réel. De plus en plus d'objets se voient ainsi équipés de processeurs et de moyens de communication mobiles, leur permettant de traiter des informations mais également de les transmettre [11]. Les réseaux de capteurs sans-fil entrent dans ce cadre. En effet, ceux-ci sont constitués d'un ensemble de petits appareils, ou capteurs, possédant des ressources particulièrement limitées, mais qui leur permettent néanmoins d'acquérir des données sur leur environnement immédiat, de les traiter et de les communiquer. Ces capteurs peuvent être reliés formant ainsi un réseau sans fil se basant sur des protocoles et des programmes pour se communiquer.

1.2 Capteurs sans fil

Les capteurs sont des dispositifs de taille extrêmement réduite avec des ressources très limitées, autonomes, capables de traiter des informations et de les transmettre, via les ondes radio, à une autre entité (capteurs, unité de traitements etc.) sur une distance limitée à quelques mètres [10]. Les réseaux de capteurs utilisent un très grand nombre de capteurs, pour former un réseau. Un capteur analyse son environnement, et propage les données récoltées aux capteurs appartenant à sa zone de couverture. Chaque capteur relayant l'information sur sa propre zone de couverture, le réseau se trouve entièrement couvert. Les FIGURE (1.1) et (1.2) donnent respectivement un capteur et un réseau de capteurs.



FIGURE 1.1 – Capteur sans fil [19].

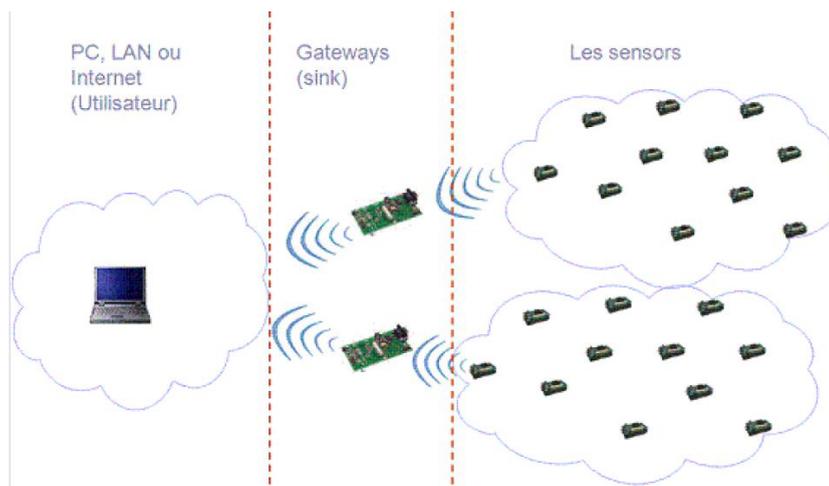


FIGURE 1.2 – Architecture d'un réseau de capteurs [23].

1.3 Architecture de capteur sans fil

Un nœud capteur contient quatre unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission, et l'unité de contrôle d'énergie. Il peut contenir également, suivant son domaine d'application, des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire), où un module de mobilité. La FIGURE(1.3) donne l'architecture fonctionnelle d'un capteur [3].

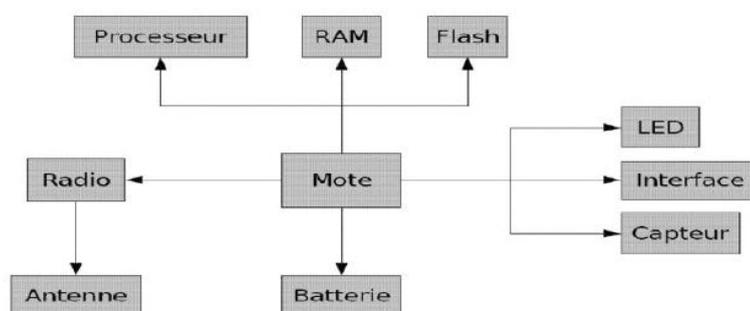


FIGURE 1.3 – Architecture fonctionnelle d'un capteur sans fil [9].

1.3.1 Unité de traitement (Mote, processeur, RAM et Flash)

On appelle généralement Mote la carte physique utilisant le système d'exploitation pour fonctionner. Celle-ci a pour cœur le bloc constitué du processeur et des mémoires RAM et Flash. Cet ensemble est à la base du calcul binaire et du stockage, temporaire pour les données et définitif pour le système d'exploitation. Cette unité est chargée d'exécuter les protocoles de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits .

1.3.2 Unité de transmission

Radio et antenne : les équipements étudiés sont donc généralement équipés d'une radio ainsi que d'une antenne. Cette unité est responsable d'effectuer toutes les émissions et réceptions des données sur un medium sans fil. Elle peut être de type optique (comme dans les nœuds Smart Dust), ou de type radiofréquence. Les communications de type optique sont robustes vis-à-vis des interférences électriques. Néanmoins, elles présentent l'inconvénient d'exiger une ligne de vue permanente entre les entités communicantes. Par conséquent, elles ne peuvent pas établir de liaisons à travers des obstacles [11].

1.3.3 Unités de captage(LED, interface, capteur)

Le capteur est généralement composé de deux sous-unités : le récepteur (reconnaissant l'analyste) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur est responsable de fournir des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique. Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement [9].

1.3.4 Unités de contrôle d'énergie

Batterie : capteur est muni d'une ressource énergétique (généralement une batterie de type AAA) pour alimenter tous ses composants. Cependant, en conséquence de sa taille réduite, la ressource énergétique dont il dispose est limitée et généralement irremplaçable. Cette unité peut aussi gérer des systèmes de rechargement d'énergie à partir de l'environnement observé telles que les cellules solaires, afin d'étendre la durée de vie totale du réseau. Cependant quelques différences existent suivant les fabricants. Chacun d'eux développe son type de capteurs, ces types peuvent être mica, mica2, telos ou telosb par exemple.

1.4 Applications des RCSF

Les RCSFs peuvent avoir beaucoup d'applications voir la FIGURE(1.4). Parmi elles,nous citons :

1.4.1 Découvertes de catastrophes naturelles

On peut créer un réseau autonome en dispersant des nœuds dans la nature. Des capteurs peuvent ainsi signaler des événements tels que les feux de forêts, les tempêtes ou les inondations. Ceci permet une intervention des secours beaucoup plus rapide et efficace.

1.4.2 Détection d'intrusions

En plaçant, à différents points stratégiques,descapteurs, on peut ainsi prévenir des cambriolages ou des passages de gibiersur une voie de chemin de fer par exemple sans avoir à recourir à decoûteux dispositifs de surveillance vidéo [6].

1.4.3 Applications métier

On pourrait stocker des enrées nécessitant un certain taux d'humidité et une certaine température (min ou max). Dans ces applications, le réseau doit pouvoir collecter ces différentes informations et alerter en temps réel si les seuils critiques sont dépassés.

1.4.4 Contrôle de la pollution

On pourrait disperser des capteurs au-dessus d'un emplacement industriel pour détecter et contrôler des fuites de gaz ou deproduits chimiques. Ces applications permettent de

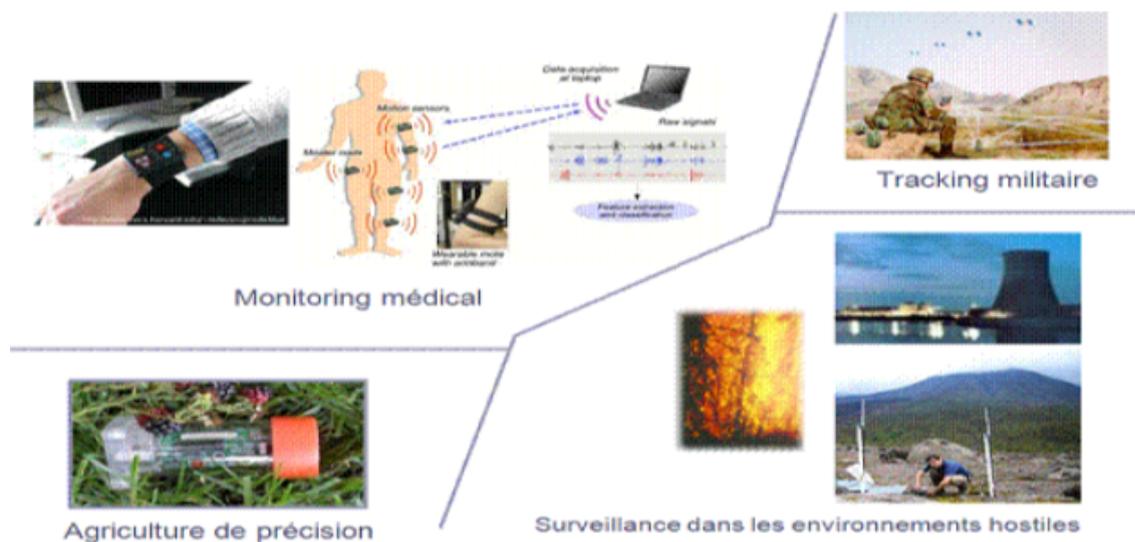


FIGURE 1.4 – Applications des WCSF [15].

1.5 Caractéristiques des réseaux des capteurs

1.5.1 Systèmes d'exploitation pour les réseaux de capteurs

Voici quelques exemples pour les systèmes d'exploitation pour les réseaux de capteurs sans fil.

1.5.1.1 TinyOS

TinyOS est un système d'exploitation open-source pour les réseaux des capteurs. Le caractère open source permet à ce système d'être régulièrement enrichie par une multitude d'utilisateurs.

1.5.1.2 MOS

MOS est un système d'exploitation open-source aussi pour les capteurs. Il est développé en C. Il supporte les plateformes de la famille MICA et de la famille Telos [20].

1.5.1.3 SOS

SOS est un système d'exploitation open-source pour les réseaux de capteurs développés par le laboratoire réseaux et systèmes embarqués de l'université de Los Angeles [10].

1.5.2 Mode de consommation d'énergie

Les capteurs sans fil sont des éléments indépendants les uns des autres, comme leur nom l'indique. Par conséquent, ils doivent également disposer d'une alimentation autonome. Leur durée de vie est limitée par la durée de vie de leur batterie. Cette contrainte forte a une influence majeure sur l'ensemble des techniques, mises en place pour le déploiement de tels réseaux. Un effet majeur de cette limitation énergétique est la limitation maximale des transmissions par voie hertzienne, très coûteuses. Il est donc primordial d'effectuer tant que possible le traitement de l'information localement au niveau du nœud. L'enjeu est donc d'étendre la durée de vie du système et sa robustesse, en cas de chute de certains nœuds seulement. Les problématiques sont donc très éloignées de celles des réseaux classiques, telle la maximisation du débit. Dans les réseaux de capteur sans fil, il faut assurer une consommation répartie de l'énergie au sein du réseau. Cet énergie est consommée par les diverses fonctionnalités de réseaux qui sont donc par ordre décroissant de consommation d'énergie Radio (Communication) , protocoles (MAC, routage), CPU (calcul, agrégation) et Acquisition [4].

1.6 Ordonnancement de réseaux de capteur

Les réseaux de capteurs sont généralement denses et redondants. En effet, suivant l'application, on déploiera plus ou moins des capteurs dans un souci d'allongement de la durée de vie de l'application. À tout moment, il existe donc des capteurs qui observent une même portion de la zone de déploiement. Cette redondance est exploitée par l'ordonnancement d'activité : Ordonnancer l'activité dans un réseau de capteurs consiste à alterner les charges de façon à épuiser les nœuds équitablement [5] . Pendant qu'une partie participe à l'application, les autres sont dans un mode passif, économisant ainsi leur énergie L'ordonnancement d'activité peut se faire de diverses façons. Nous distinguons ici les approches centralisées (où une entité centrale connaît chaque nœud et est capable d'influer sur chacun pour lui assigner ses tâches). Des approches hiérarchiques (une vision hiérarchisée du réseau où l'autorité centrale est démultipliée selon plusieurs entités responsables d'une sous-partie du réseau) et des approches localisées, par conséquent totalement décentralisées, dans lesquelles un comportement global cohérent doit être obtenu à partir de décisions prises localement. La FIGURE(1.5) illustre les différentes approches d'ordonnancement dans les RCSFs.

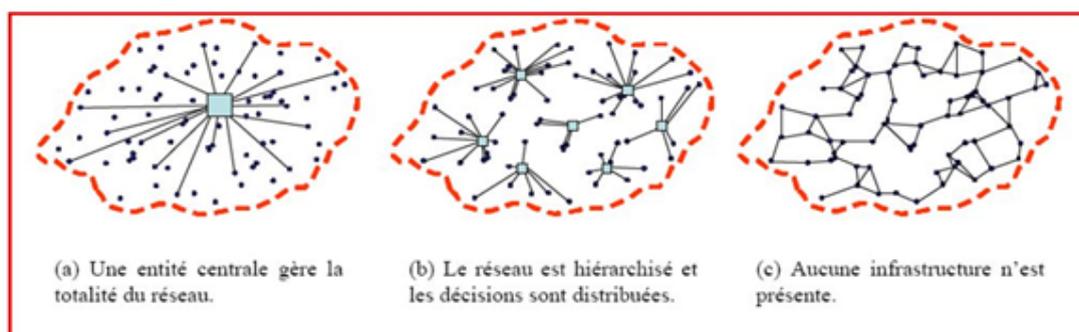


FIGURE 1.5 – Trois approches algorithmiques pour l’ordonnancement d’activité [11].

1.7 Déploiement des capteurs

Les capteurs sont au préalable déployés sur une zone à surveiller. Pour satisfaire de nouvelles contraintes ou pour pallier des pannes, un déploiement de nœuds supplémentaires, dit itératif, peut être requis. Différents modes de déploiement sont envisageables et dépendent essentiellement de l’application de surveillance.

1.7.1 Déploiement déterministe

Lorsque l’environnement est accessible ou connu, il est possible de placer précisément les nœuds sur la zone, il est aussi possible de programmer leurs activités au préalable. C’est ainsi, par exemple, que sont mis en place les capteurs chargés de réguler la climatisation d’un immeuble ou de surveiller les constantes médicales de malades à distance. On parle alors de déploiement déterministe.

1.7.2 Déploiement aléatoires

L’utilisation des capteurs dans des zones inaccessibles ou sensibles rend impossible un déploiement déterministe, au cours duquel chaque objet serait placé à une position prédéterminée. Les nœuds peuvent alors être déployés à l’aide de moyens divers. Il a souvent été question d’un déploiement aléatoire des capteurs, effectué comme un jeté de graines. Il semble pourtant difficile, vu la fragilité des capteurs existants, d’envisager un déploiement par avion par exemple. Néanmoins, un déploiement aléatoire peut être obtenu à partir d’une distribution de capteurs à des individus.

1.8 Contraintes de conception des RCSF

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit :

1.8.1 Tolérance aux fautes

Certain nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. Ces problèmes n'affectent pas le reste du réseau, c'est le principe de la tolérance de fautes. La tolérance aux fautes est la capacité de maintenir les fonctionnalités du réseau sans interruptions dues à une erreur intervenue sur un ou plusieurs capteurs.

1.8.2 Echelle

Le nombre de nœuds déployés pour un projet peut atteindre le million. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que le puit "sink" soit équipé de beaucoup de mémoire pour stocker les informations reçues.

1.8.3 Coûts de production

Souvent, les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un nœuds est critique afin de pouvoir concurrencer un réseau de surveillance traditionnel. Actuellement un nœuds ne coûte souvent pas beaucoup. A titre de comparaison, Un nœuds bluetooth, pourtant déjà connu pour être un système low-cost.

1.8.4 Environnement

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

1.8.5 Contraintes matérielles

La principale contrainte matérielle est la taille du capteur. Les autres contraintes sont que la consommation d'énergie.

1.8.6 Topologie de réseau

Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases : Déploiement, Post-déploiement (les capteurs peuvent bouger, ne plus fonctionner, etc.), Redéploiement de nœuds additionnels.

1.9 Conclusion

Dans ce premier chapitre, nous avons présenté les RCSFs, leurs spécificités et les concepts nécessaires à la compréhension des réseaux de capteurs. Cependant, nous avons remarqué que plusieurs facteurs et contraintes compliquent la gestion de ce type de réseaux. Un choix de protocole de routage et d'un algorithme adéquat pour le routage d'information dans notre réseau ainsi qu'une approche algorithmique pour l'ordonnancement d'activité s'impose pour bien simuler les réseaux de capteurs sans fil. Ceci va être étudié et présenté dans le prochain chapitre.

ROUTAGE DANS LES RÉSEAUX DE CAPTEURS SANS FIL

2.1 Introduction

Un réseau de capteurs sans fil consiste en un ensemble de nœuds capteurs déployés à grande échelle dans un champ de captage pour détecter, recueillir et transmettre les données, concernant un phénomène observé, vers le nœud puits via les liens sans fil. Cependant, suivant le nombre de nœuds du réseau et l'étendu du champ de captage, certains nœuds ne pourront pas transmettre directement leurs messages au nœud collecteur. Ainsi, la collaboration entre les nœuds pour garantir cette transmission est une exigence. De cette manière, les messages sont propagés par les nœuds intermédiaires en établissant les chemins multi-sauts entre la source lointaine et le puits. Ce processus d'acheminement des messages d'un nœud source du réseau vers un nœud destinataire s'appelle le routage. En tenant compte des capacités réduites des nœuds capteurs (calcul, énergie, mémoire), la communication avec le puits devrait se faire sans protocole de routage. Dans ce cas, la solution la plus simple serait, pour chaque nœud capteur, d'envoyer ses messages par diffusions jusqu'à ce qu'ils arrivent au collecteur. Cependant cette simplicité provoque des désavantages significatifs tels que l'implosion et le chevauchement des messages. Une implosion est détectée parce que les nœuds reçoivent des copies multiples du même message (problème de redondance de données). De plus les nœuds ne tiennent pas compte de leurs ressources pour limiter leurs opérations (émission, calcul). Ainsi, pour qu'un réseau de capteur soit efficace, la mise en place d'un algorithme de routage devient inévitable. Néanmoins, vu les contraintes imposées par ces réseaux, la mise en place d'un algorithme de routage n'est pas une tâche facile. Plusieurs travaux de recherche dans le domaine des RCSF ont été effectués récemment et ont abouti à une multitude de protocoles de routage destinés à ces réseaux. Dans ce chapitre,

nous dressons un état de l'art sur le routage dans les RCSF ainsi les différentes classifications des protocoles qui existent.

2.2 Les principaux protocoles de routage dans les RCSF

Les réseaux de capteurs sans fil étant basés sur les réseaux ad hoc, on retrouve le même problème, avec cependant quelques contraintes différentes. En effet, on connaît généralement le type de données qui vont circuler, et dans une architecture traditionnelle des réseaux de capteurs, les données sont toutes renvoyées vers un puits [7]. La FIGURE(2.1)résume les principaux protocoles de routage .

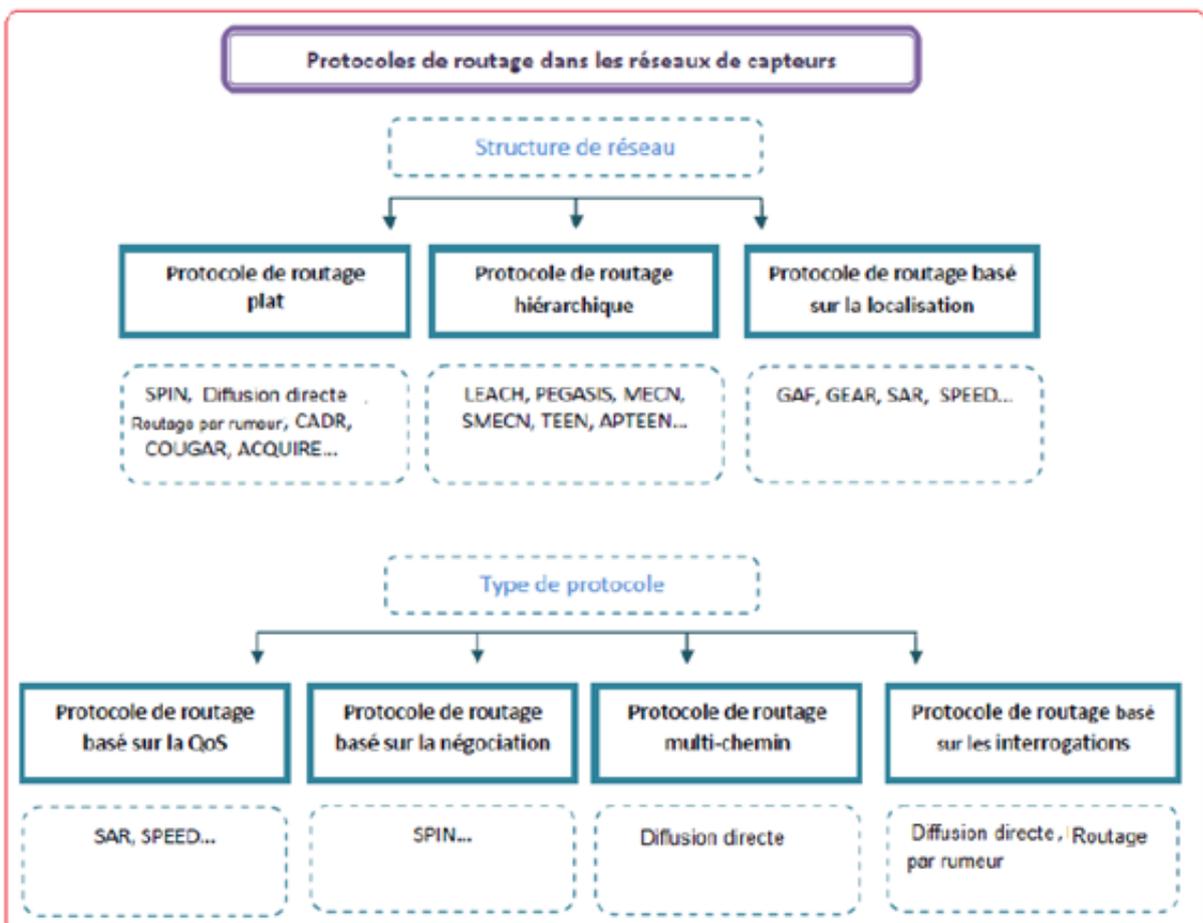


FIGURE 2.1 – Les principaux protocoles de routages dans les RCSF [5].

2.2.1 Les Protocoles hiérarchiques

L'objectif principal du routage hiérarchique est de maintenir efficacement la consommation d'énergie de nœuds de capteurs en les impliquant dans la communication multi-hop au sein d'un cluster et en effectuant l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis à la destination. La formation de clusters est généralement fondée sur la réserve d'énergie des capteurs et sur les capteurs qui sont à proximité de cluster-head voir la FIGURE(2.2). LEACH (LowEnergy Adaptive Clustering Hierarchical) [2] est l'une de premières approches de routage pour les réseaux de capteurs. L'idée proposée par LEACH a été une inspiration pour de nombreux protocoles de routage hiérarchique, bien que certains protocoles aient été développés de manière indépendante.

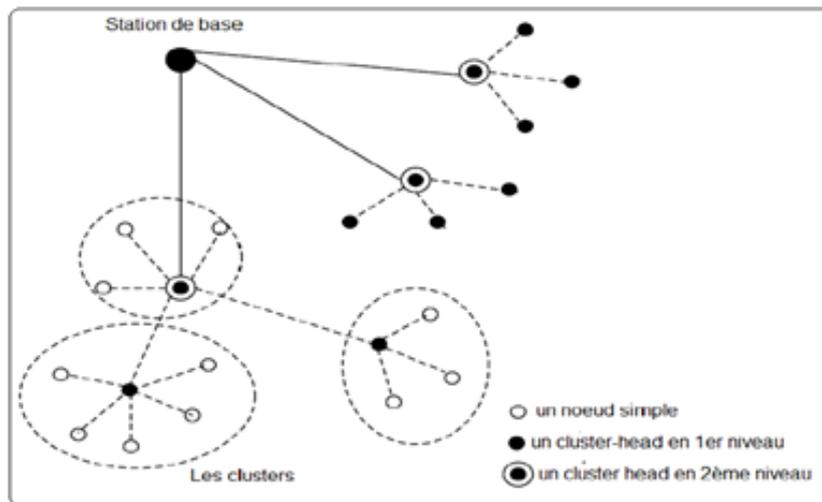


FIGURE 2.2 – Topologie hiérarchique [2].

2.2.1.1 Protocole de routage "LEACH "

LEACH est l'un des algorithmes de routage hiérarchique le plus populaire pour les réseaux de capteurs [24]. L'idée est de former des clusters de nœuds de capteurs basés sur les zones où il y a un fort signal reçu, puis utiliser des clusters-heads locaux comme passerelle pour atteindre la destination. Cela permet d'économiser de l'énergie car les transmissions ne sont effectuées que par les cluster-head plutôt que par tous les nœuds de capteurs.

2.2.1.2 Protocoles de routage "TEEN et APTEEN"

Les protocoles Threshold sensitive Energy Efficient sensor Network protocol (TEEN)[18] et Adaptive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) conviennent

pour les applications critiques. Dans les deux protocoles, le facteur clé est la valeur de l'attribut mesuré. La caractéristique supplémentaire d'APTEEN est la capacité de changer la périodicité et les paramètres de TEEN en fonction des besoins des utilisateurs et des applications. TEEN est conçu pour être sensible à des changements soudains des attributs tels que la température. La réactivité est importante pour les applications critiques dont le réseau fonctionne dans un mode réactif. L'architecture du réseau de capteurs est basée sur un groupement hiérarchique où les nœuds forment des clusters et ce processus va se répéter jusqu'à ce que la station de base soit atteinte [12].

APTEEN est une extension de TEEN qui fait à la fois la collection des captures périodique de données et qui réagit aux événements critiques. Quand la station de base forme des clusters, les clusters head diffusent les attributs, les valeurs des seuils, ainsi que le calendrier de transmission à tous les nœuds. Le cluster-head effectue également l'agrégation de données afin d'économiser l'énergie.

2.2.2 protocoles de routage basés sur la localisation

Les protocoles de routage basés sur la localisation [12] utilisent les informations d'emplacement pour guider la découverte de routage et la transmission des données. Ils permettent la transmission directionnelle de l'information en évitant l'inondation d'information dans l'ensemble du réseau. Par conséquent, le coût de contrôle de l'algorithme est réduit et le routage est optimisé. De plus, avec la topologie réseau basée sur des informations de localisation de nœuds, la gestion du réseau devient simple. L'inconvénient de ces protocoles de routage est que chaque nœud doit connaître les emplacements des autres nœuds.

2.2.2.1 protocole de routage " MECN "

Minimum Energy Communication Network (MECN) [13] est un protocole de routage qui cherche à établir et à entretenir une énergie minimale pour les réseaux sans fil en utilisant des GPS de faible puissance. MECN utilise une station de base comme destination de l'information, ce qui est toujours le cas pour les réseaux de capteurs. MECN identifie une région de relais pour chaque nœud. La région de relais se compose de nœuds dans une zone périphérique où la transmission à travers ces nœuds est plus économe en énergie que la transmission directe. L'idée principale de MECN est de trouver un sous-réseau qui a moins de nœuds et qui nécessite moins d'énergie pour la transmission entre deux nœuds quelconques. Cela est effectué en utilisant une recherche localisée pour chaque nœud en prenant en considération sa région de relais [12].

2.2.2.2 protocole de routage " GAF "

GAF (Geographic Adaptive Fidelity) est un protocole de routage basé sur la localisation des nœuds. Il est conçu principalement pour les réseaux mobiles ad hoc, mais peut être applicable aux réseaux de capteurs. La localisation des nœuds dans GAF pourrait être fournie à l'aide d'un GPS ou d'autres techniques de localisation [14]. Il consiste à former des grilles virtuelles de la zone concernée en partitionnant cette zone où les nœuds sont déployés en de petites zones telles que, pour deux grilles adjacentes G_x et G_y , tous les nœuds de G_x peuvent communiquer avec tous les nœuds G_y . Ainsi, ce système de partitionnement GAF assure la fidélité du routage car il existe au moins un chemin entre un nœud et la station de base. GAF peut augmenter considérablement la durée de vie du réseau. En effet, un seul nœud dans chaque grille reste à l'état actif en faisant passer les autres nœuds de la grille à l'état de sommeil pour une certaine période de temps tout en assurant la fidélité du routage. Cependant, dans certains environnements où les nœuds sont fortement mobiles, la fidélité du routage pourrait être réduite si un nœud actif quitte la grille. Ainsi, le nombre de données perdues sera important.

2.2.3 protocoles de routage 'data-centric'

Dans de nombreuses applications de réseaux de capteurs, vu le nombre élevé de nœuds déployés, il n'est pas possible d'attribuer des identificateurs globaux à chaque nœud. Cette absence d'identification globale avec le déploiement aléatoire de nœuds de capteurs font qu'il est difficile de sélectionner un ensemble spécifique de nœuds de capteurs à interroger. Par conséquent, les données sont généralement transmises de chaque nœud de capteurs dans la région de déploiement avec une redondance importante. Cette réflexion a conduit au routage data-centric qui est différent du traditionnel routage où les routes sont créées entre les nœuds adressables gérée dans la couche réseau. Le destinataire envoie des requêtes à certaines régions et attend à recevoir des données provenant des capteurs situés dans les régions sélectionnées. Comme les données sont demandées à travers des requêtes, le nommage est nécessaire pour préciser les propriétés des données . Comme la montre l'exemple d'une approche data-centric dans la FIGURE(2.3), les données provenant des deux sources sont agrégées au nœud B. Ensuite, la donnée combinée (1+2) est envoyée de B vers la destination [12].

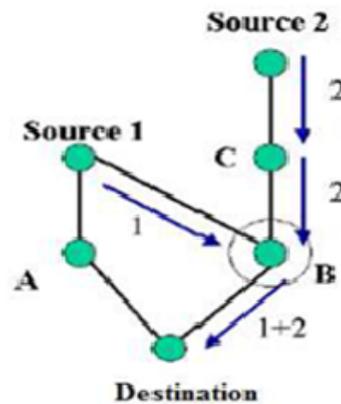


FIGURE 2.3 – Routage data-centric [12].

2.2.3.1 protocole de routage " SPIN "

Un réseau de capteurs sans fil se compose de plusieurs capteurs qui sont déployés dans différentes régions. L'accès aux données d'un événement particulier pour une région ou une zone spécifique peut faire une grande différence. Des mesures peuvent être prises pour l'obtention de données à partir d'un endroit ciblé. Les données sont recueillies auprès de tous les capteurs et transmises de manière redondante sur le réseau, ce qui entraîne une utilisation inefficace de l'énergie. Afin de résoudre ces types de problèmes, les systèmes de routage data- centricont évolué : la destination envoie des requêtes pour les capteurs du réseau dans un endroit choisi. Les attributs sont utilisés pour demander des données provenant des capteurs. SPIN (Sensor Protocols for Information via Negotiation) [14]. Est le premier protocole data-centric qui a été conçu pour les réseaux de capteurs sans fil. Il présente de nombreuses similitudes par rapport à la diffusion dirigée. Il est efficace dans la réduction des données redondantes et économise l'énergie. La diffusion est le processus de collecte des observations de l'ensemble des capteurs individuel qui sont déployées dans le réseau et où tous les capteurs sont traités comme des nœuds destination [22]. Les tâches confiées à ces capteurs sont de recueillir le point de vue complet de l'environnement sous la forme de données et de renforcer une structure de réseau avec une tolérance aux fautes. La consommation d'énergie durant les calculs et les communications doit être contrôlée afin de prolonger la durée de vie des capteurs au sein du réseau.

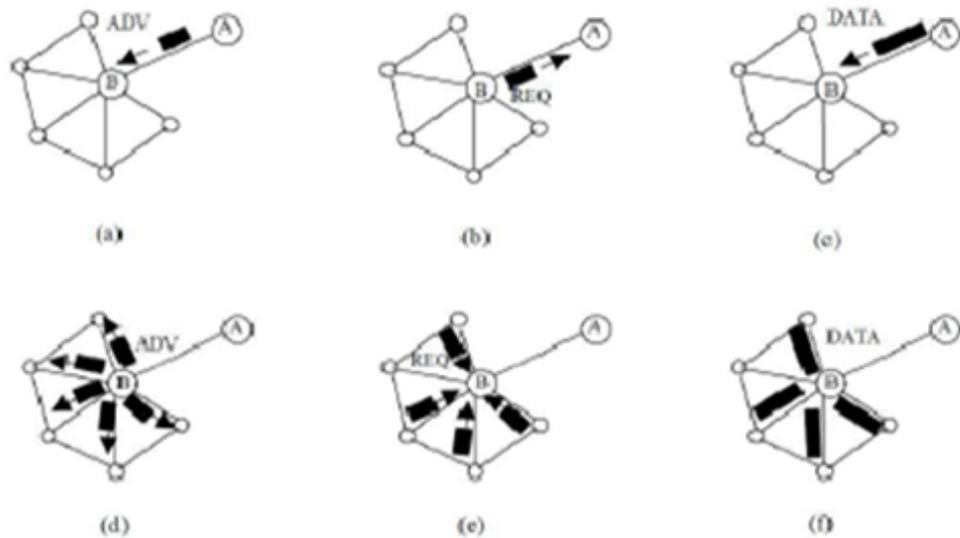


FIGURE 2.4 – Les phases de communication du protocole SPIN [14].

2.2.3.2 Diffusion dirigée

La diffusion dirigée est un protocole important dans le routage data-centrique des réseaux de capteurs. L'idée vise à diffuser des données aux nœuds en utilisant un schéma de nommage pour les données. La raison principale derrière l'utilisation d'un tel système est de se débarrasser des opérations inutiles de routage de couche réseau afin d'économiser l'énergie. La diffusion dirigée suggère l'utilisation de paires attribut-valeur pour les données et les requêtes des capteurs. Afin de créer une requête, un nœud est défini à l'aide d'une liste de paires attribut-valeur comme le nom des objets, l'intervalle, la durée, la zone géographique, etc. Un paquet est diffusé par ce nœud vers la destination à travers ses voisins. Chaque nœud qui reçoit les paquets peut les stocker pour une utilisation ultérieure [14]. Les paquets stockés sont ensuite utilisés pour comparer les données reçues. La requête contient aussi plusieurs champs de gradient. Un gradient est un lien réponse avec un voisin dont le paquet a été reçu et qui est caractérisé par le débit, la durée et la date d'expiration de données. Ainsi, en utilisant les intérêts et les gradients, les routes sont établies entre la destination et les sources. Plusieurs routes peuvent être établies de telle sorte que l'une d'elle est choisie par renforcement. La destination renvoie le message d'intérêt initial à travers la route choisie. Un intervalle plus petit renforce donc le nœud source sur ce chemin pour envoyer des données plus fréquemment.

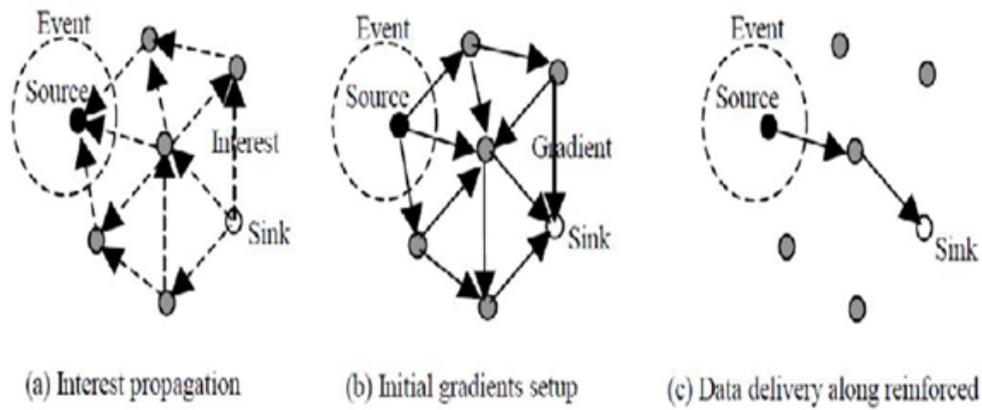


FIGURE 2.5 – Les phases de communication du protocole Directed Diffusion [14].

2.3 Classification des protocoles de routages dans les réseaux de capteurs sans fil

le tableau suivant montre différents protocoles de routage dans les RCSF ainsi que leurs classification

Protocole de routage	Data-Centric	Hierarchical	Location-based	QoS	Network-flow	Data aggregation
SPIN	*					*
Directed-Diffusion	*					*
Rumor Routing	*					*
LEACH		*			*	
Teen and Apteen	*	*				*
Pegasis		*				*
MECN and SMECN			*			
GAF		*	*			
GEAR			*			
SAR				*		
SPEED			*	*		

FIGURE 2.6 – Classification des protocoles de routages dans les réseaux de capteurs sans fil [22].

2.4 Contraintes de routage dans les réseaux de capteurs sans fil

Le routage dans les réseaux de capteurs diffère de celui des réseaux Ad Hoc dans les points suivants :

- il n'est pas possible d'établir un système d'adressage global pour le grand nombre de nœuds .
- les applications des réseaux de capteurs exigent l'écoulement de données mesurées depuis des sources multiples vers la destination finale " Sink".
- les nœuds capteurs exigent ainsi une gestion soigneuse des ressources.

2.5 Routage dans les réseaux de capteurs à base d'architecture P2P

La spontanéité, l'adaptabilité du réseau et la dynamique de la topologie dans le déploiement des WSN soulèvent de nombreuses questions ouvertes. De plus, compte tenu du nombre potentiellement important de capteurs participants et de leurs ressources réduites, il apparaît indispensable de développer des solutions entièrement décentralisées, répartissant la charge entre les participants et permettant de passer à l'échelle. De nombreux procédés des systèmes distribués, et plus particulièrement ceux issus du paradigme pair-à-pair (P2P), nous apparaissent comme adaptables aux réseaux de capteurs sans fil [14].

2.5.1 Paradigme de communication pair-à-pair

Le paradigme de communication pair-à-pair en particulier possède une capacité intrinsèque de passage à l'échelle. Les systèmes reposant sur celui-ci (dit systèmes P2P par la suite) sont totalement décentralisés, chaque nœud jouant à la fois le rôle de client et de serveur. La charge étant répartie sur tous les nœuds du système, augmenter le nombre de participants n'induit pas de goulot d'étranglement ni de point critique de défaillance, couramment observés lors de l'utilisation du paradigme client/serveur. Les notions de serveur et de client disparaissent pour laisser la place à celle de pair. Le terme reflète particulièrement bien la notion d'égalité entre les différents nœuds du réseau, tous les nœuds jouant le même rôle. Les réseaux de pairs sont des réseaux logiques et non physiques. Ainsi, l'organisation des pairs dans le réseau logique ne reflète pas du tout la topologie du réseau physique : deux pairs connectés directement peuvent être physiquement très éloignés alors que deux physiquement proches ne se sont pas mis en relation [22]. De plus, la vision uniquement locale des nœuds des systèmes P2P (ie. pas de connaissance globale du système, communication avec un sous ensemble restreint de participants) rendent l'adaptation de ceux-ci aux WSN avantageuse. Les architectures P2P peuvent être classées en 3 types : les architectures structurées, les architectures non structurées et les architectures hybrides. Dans ce chapitre, on s'intéresse uniquement aux architectures structurées et plus particulièrement le protocole CAN.

2.5.1.1 Exemple de Réseaux P2P structurés

Dans les systèmes structurés, une organisation de connexion est maintenue entre les nœuds. La plupart sont basées sur les tables de hachage distribuées (DHT), permettant ainsi de réaliser des recherches en un nombre de messages croissant de façon logarithmique avec le nombre d'utilisateurs du réseau, comme CAN, Chord, Freenet, et Pastry.

- CAN

Les nœuds du réseau CAN reposent sur un espace de coordonnées cartésiennes de dimension d , sans rapport avec les coordonnées physiques des nœuds. Tout l'espace CAN est partagé entre les nœuds du réseau, de telle sorte que chaque nœud possède sa propre zone distincte à laquelle sont rattachées des paires (clé, valeur)[1]. Deux nœuds sont voisins si leurs coordonnées se recouvrent sur $d-1$ dimensions et sont contigües sur une dimension. Dans l'exemple de la FIGURE(2.7), le nœud 5 est voisin du nœud 1 car sa zone de coordonnées chevauche celle d'unœud 1 le long de l'axe Y et elle lui est contigüe le long de l'axe X. Par contre, le nœud 6 n'est pas un voisin du nœud 1 car ses zones de coordonnées sont contigües sur les deux axes X et Y ensemble.

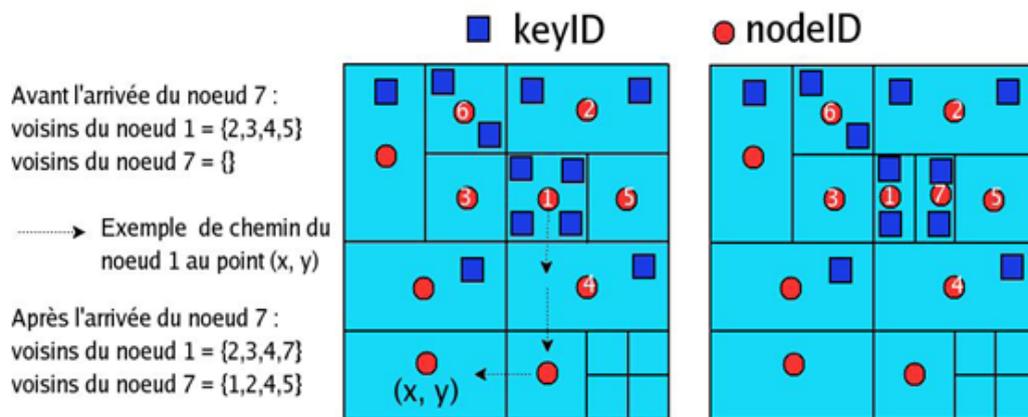


FIGURE 2.7 – Exemple de réseau CAN d'espace2-d [1].

a) Routage Un nœud CAN maintient une table de routage qui contient l'adresse IP et la zone cartésienne virtuelle de chacun de ses voisins. Un nœud route un message vers un point destinataire dans l'espace cartésien. Le nœud détermine d'abord quelle zone voisine est plus proche du point de destination, puis regarde l'adresse IP du nœud de cette zone via la table de routage [1].

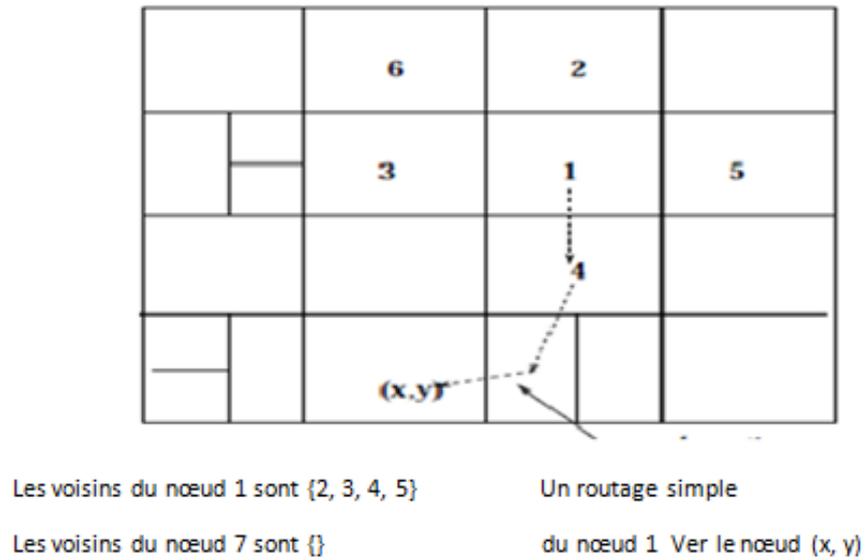


FIGURE 2.8 – Routage CAN [1].

b) Entrée d'un nœud Pour participer à une CAN, un nouveau nœud doit : - Trouver un nœud déjà dans le réseau de recouvrement. - Identifier une zone qui peut être divisée - Mettre à jour les tables de routage des nœuds voisins de la zone nouvellement divisée [22]. Pour trouver un nœud déjà dans le réseau de recouvrement, les nœuds d'amorçage peuvent être utilisés pour fournir le nouveau nœud des adresses IP des nœuds actuellement dans le réseau de recouvrement.

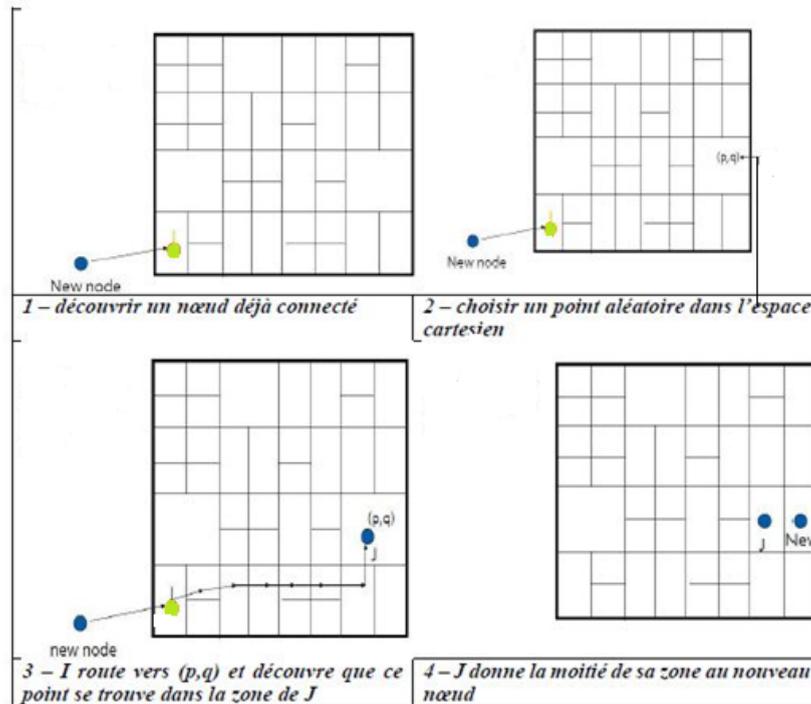


FIGURE 2.9 – Comment un nœud se connecte au réseau CAN [1].

Après avoir reçu une adresse IP d'un nœud déjà dans la CAN, le nouveau nœud peut tenter d'identifier une zone pour lui-même. Le nouveau nœud choisit au hasard un point dans l'espace de coordonnées et envoie une requête JOIN, dirigée vers le point aléatoire, à l'une des adresses IP reçues. Les nœuds déjà dans l'itinéraire de réseau de recouvrement acheminent la demande vers le nœud approprié en utilisant leurs tables de routage zone. Une fois que le nœud de gestion de la zone du point de destination reçoit la demande rejoindre, il peut honorer la demande d'inscription en scindant sa zone en deux, en s'attribuant lui-même la première moitié, et en allouant au nouveau nœud la seconde moitié. Si elle ne respecte pas la demande rejoindre, le nouveau nœud continue à sélectionner des points aléatoires dans l'espace de coordonnées et d'envoyer des requêtes JOIN adressées à ces points aléatoires jusqu'à ce qu'il rejoigne le réseau avec succès [22]. Après la scission de la zone et allocation des nouvelles zones créées, les nœuds voisins sont mis à jour avec les coordonnées des deux nouvelles zones et les adresses IP correspondantes. Les tables de routage sont mises à jour et les mises à jour sont propagées à travers le réseau.

c) Départ d'un nœud Pour gérer le départ d'un nœud, la CAN doit - identifier le nœud au départ - fusionner sa zone avec celle d'un nœud voisin - mettre à jour les tables de routage

à travers le réseau [22]. La détection du départ d'un nœud peut être faite, par exemple, par l'intermédiaire de messages de pulsation qui diffusent régulièrement des informations de table de routage entre voisins. Après une période prédéterminée de silence d'un voisin, ce nœud voisin est déterminé comme ayant échoué et est considéré comme un nœud au départ. Alternativement, un nœud qui aimerait se déconnecter volontairement peut diffuser un tel avis à ses voisins. Après l'identification d'un nœud au départ, sa zone doit être soit fusionnée soit reprise. Tout d'abord la zone du nœud parti est analysée pour déterminer si elle peut être fusionnée avec celle d'un nœud voisin pour former une zone valide. Par exemple, une zone dans un espace de coordonnées 2d doit être un carré ou un rectangle et ne peut pas être en forme de L. Le test de validation peut parcourir toutes les zones voisines afin de déterminer si une fusion valide peut se produire. Si l'une des fusions potentielles est considérée comme une fusion valide, les zones sont ensuite fusionnées. Si aucune des fusions potentielles n'est jugée valable, alors le nœud voisin avec la zone plus petite prend le contrôle de la zone du nœud au départ [1]. Après une prise de contrôle, le nœud de prise en charge peut périodiquement tenter de fusionner ses zones contrôlées avec des zones voisines respectives. Si la fusion est réussie, les tables de routage des nœuds de zones voisines sont mises à jour pour refléter la fusion. Le réseau verra cette section du réseau de recouvrement comme une zone unique après une fusion et traitera tout routage avec cet état d'esprit. Pour effectuer une prise de contrôle, le nœud ayant pris le contrôle d'une nouvelle zone met à jour les tables de routage des nœuds voisins, de sorte que toute requête à l'une des deux zones se dirige au nœud responsable. Mais le réseau voit toujours la sous-section du réseau de recouvrement comme deux zones distinctes et traite tout routage avec cet état d'esprit.

2.6 Les points communs entre les deux types de réseau RCSF ET P2P

Nous présentons ici quelques points communs entre les réseaux P2P et les réseaux de capteurs. Ces ressemblances sont importantes pour nos travaux. Les ressemblances sont à exploiter afin de mettre en relation la partie physique du réseau, représentée par les capteurs, et la partie logique du réseau, représentée par le réseau de pairs. Les réseaux de capteurs et les réseaux P2P sont des architectures décentralisées. Il n'existe pas de serveur nécessaire au bon fonctionnement du réseau. Les rôles sont répartis sur les différents nœuds tout comme les ressources. Nous avons vu que le terme de pair se justifie par l'égalité entre les différents nœuds du réseau. Les nœuds d'un WSF sont également à égalité puisqu'ils n'ont pas de rôle prédéfini. Ils peuvent prendre tour à tour les rôles de client et de serveur et être routeur dans le réseau. La spontanéité est aussi un point commun. Dans les deux cas, il n'y a pas de configuration initiale à effectuer. En effet, dans les réseaux P2P, un pair qui se connecte

recherche un autre pair dans son voisinage et accède au réseau par cette voie. La volatilité des nœuds est une donnée commune. Dans les deux cas il est difficile de faire confiance à une relation établie entre deux composants du réseau. L'instabilité des communications a des origines différentes mais présente les mêmes conséquences. Les problèmes de sécurité dans les deux réseaux sont proches car ils ont des sources communes : décentralisation, volatilité, anonymat. Les solutions de sécurité adaptées aux deux types de réseaux ont des points communs. Les algorithmes de routage dans les deux types de réseau présentent également certaines similarités.

2.7 Travaux relatifs aux RCSF basés sur des architectures P2P

2.7.1 Architecture sans puits basée sur une surcouche pair à pair

Contrairement aux réseaux de capteurs sans fil traditionnels, dans cette architecture les dispositifs des utilisateurs finaux puissent se connecter directement aux capteurs. Dans cette optique ils ont défini quatre rôles différents [21] :

- **Rôle de capteur** : qui définit la capacité à faire des mesures.
- **Rôle d'agrégateur** : qui définit la capacité à agréger du contenu.
- **Rôle de gestionnaire d'historique** : qui définit la capacité à sauvegarder des valeurs.
- **L'application de l'utilisateur final** : qui caractérise l'application qui a la capacité d'utiliser les valeurs captées.

Par rapport à une architecture classique, seul le gestionnaire d'historique est ajouté, les autres jouant des rôles similaires à ce qu'on peut voir habituellement. Ces rôles sont implémentés par deux types de dispositifs :

1. Dispositif dit capteur / agrégateur : cela correspond aux capteurs physiques qu'on trouve sur le marché. Ils jouent au choix un ou plusieurs rôles parmi ceux de capteurs, d'agrégateur, ou de gestionnaire d'historique.

2. Dispositif de l'utilisateur final : c'est le dispositif sur lequel l'application de l'utilisateur final tourne. Il pourra aussi éventuellement implémenter le rôle de gestionnaire d'historique.

2.7.1.1 Principe de fonctionnement

Dans cette architecture tous les dispositifs ont la même pile de protocole. En effet, compte tenu du caractère totalement décentralisé de l'architecture et des faibles ressources des nœuds, il n'est pas acceptable d'utiliser certains nœuds comme relais de communication.

le principe c'est que chacun des nœuds peut être soit fixe soit mobile. En effet, dans le scénario du système de prévention des accidents tous les nœuds sont mobiles, tandis que dans le scénario du système de prévention d'incendie, seuls les dispositifs des utilisateurs finaux sont mobiles. Il est donc important de couvrir le maximum de cas, c'est pourquoi cette architecture prend en compte les deux possibilités de mobilité ou non des nœuds. l'ensemble des nœuds du réseau sont adressables. C'est-à-dire qu'on pourra établir une communication directe entre deux nœuds. C'est un élément très important dans l'architecture décentralisée, puisque cela va nous permettre d'assurer la fiabilité du réseau quel que soit l'état de celui-ci.

2.7.1.2 Présentation de l'architecture

cette architecture est composée de deux principales entités représentées sur la FIGURE(2.11) Tout d'abord le nuage (a) qui représente le réseau réel, constitué des différents dispositifs utilisés, à savoir les capteurs / agrégateurs et les équipements des utilisateurs finaux. Le nuage (b) représente la surcouche pair à pair réalisée à l'aide de l'intergiciel. Les traits gris représentent le fait que les équipements se connectent à l'intergiciel pair à pair.

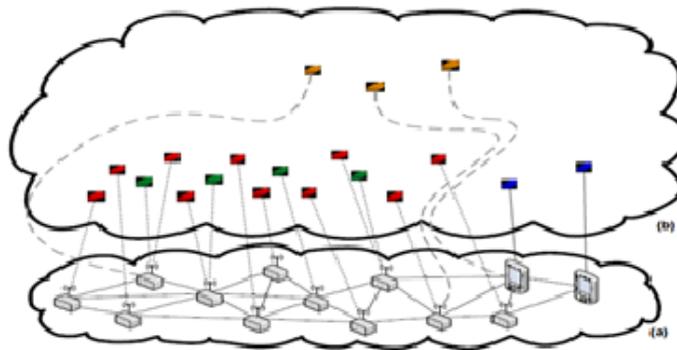


FIGURE 2.10 – Réseaux réels et surcouche pair à pair [21].

2.7.1.3 Schéma global de l'architecture

Voici donc un schéma représentant l'ensemble de l'architecture que nous proposons. Elle est sans puits et permet une connexion directe entre un agrégateur et un capteur.

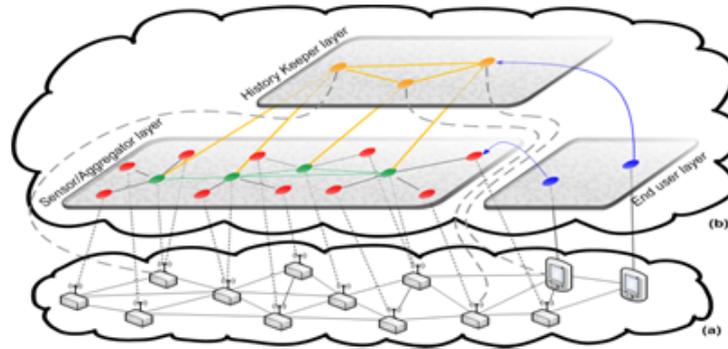


FIGURE 2.11 – Architecture sans puits basée sur une surcouche pair à pair pour réseau de capteurs sans fil, (a) réseau réel des dispositifs; (b) surcouche pair à pair [21].

Le principe général de fonctionnement est le suivant : les capteurs envoient leurs mesures à la tête de la grappe qui va ensuite s'occuper de les faire sauvegarder par un gestionnaire d'historique. Les utilisateurs finaux peuvent alors accéder aux données via l'historique ou par connexion directe avec un capteur.

2.7.2 P2P-RPL (Point To Point Routing Protocol for Low Power Lossy Networks)

- RPL

RPL[8] est un protocole de routage qui organise les routeurs le long d'un graphe acyclique dirigé orienté par destination (DODAG) (Destination Oriented Directed Acyclic Graph), le graphe acyclique dirigé, enracinée par un puits Routeur S sur la FIGURE (2.13). La racine lance le DODAG information par DODAG information d'objet périodiquement (DIO) (DODAG Information Object). les messages qui sont publiés via le lien local Multicast. Les messages DIO portent des informations telles que L'identité de la racine DODAG, les paramètres de routage utilisés, ainsi que la profondeur du routeur d'origine (appelé le "rang") dans le DODAG. Un routeur se joint au DODAG en tenant compte Ces facteurs, détermine son propre rang dans le cadre du DODAG basés sur les informations annoncées par ses voisins dans leur DIOs. Le routeur choisit comme parent dans le DODAG le Voisin dont le rang résultant est le plus petit, Parmi les voisins qui publient des messages DIO.

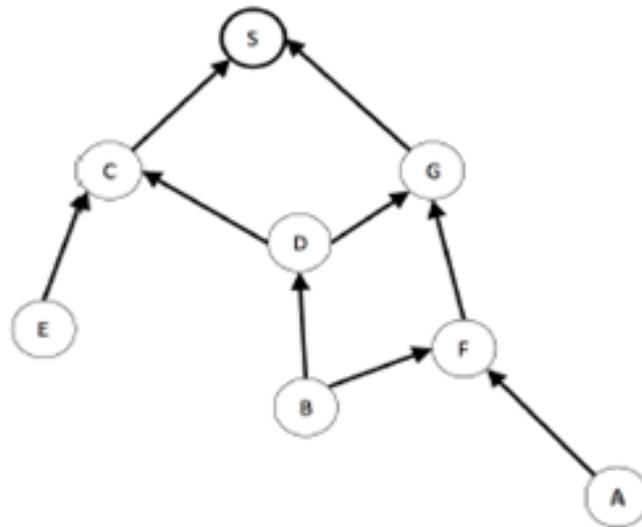


FIGURE 2.12 – Destination Oriented Directed Acyclic Graph(DODAG) rooted at router S [8].

A donc rejoint le DODAG, il a un chemin vers le puit à travers son parent (parents, et le routeur peut alors créer son propre DIO messages. RPL fournit ainsi des chemins allant des routeurs à un puit nécessitant des routeurs pour stocker les informations de la table de routage, les informations nécessaires sur son parent dans le DODAG. Cette caractéristique est compatible avec la technologie sans fil. Les capteurs, généralement sont peu coûteux et plutôt peu fiables des dispositifs qui ont des contraintes drastique en termes de CPU et mémoire (généralement quelques kilobytes de RAM et de ROM dans le total). Le taux de transmission des messages DIO suit en outre une politique Trickle qui vise à l'élagage inutile, transmissions en surveillant la cohérence de l'information entre voisins. Lorsque les données d'un nœud ne sont pas d'accord avec ses voisins, ce nœud communique rapidement pour résoudre le problème d'incohérence. Lorsque les nœuds sont d'accord, ils ralentissent leur communication, de sorte que les nœuds envoient des paquets très rarement. Cette caractéristique est également compatible au sans fil.

RPL fournit des chemins pour point à point (P2P) Communication entre capteurs arbitrairement dans le réseau. Puisque la communication P2P est une exigence fondamentale pour plusieurs applications, une extension de Protocole, appelé P2P-RPL est actuellement considéré. Afin de fournir des chemins P2P plus courts entre les capteurs, lorsque ils sont disponibles.

2.7.2.1 Principe de fonctionnement

P2P-RPL permet aux routeurs de découvrir et d'établir les chemins pour un autre routeur, basé sur un simple mécanisme réactif. lorsqu'un routeur S a besoin de découvrir un chemin vers Un autre routeur D, le routeur S provient d'un message similaire dans la fonctionnalité a une demande de chemin Route-Request AODV. Des messages sont diffusés sur l'ensemble du réseau, créant efficacement un DODAG temporaire enraciné dans S. En parcour le réseau, le message installe les information temporairement du prochain nœud vers S sur les Routeurs Traverse et peut accumuler des informations sur le chemin parcouru jusqu'à présent. Après avoir reçu un tel message, le routeur D envoie un message à S, similaire à la fonctionnalité d'une Route-Réponse AODV, le long du chemin enregistré, donc établir un chemin entre S et D, et le temporaire DODAG finit par expirer [16]. P2P-RPL utilise les mêmes mécanismes que le RPL basique forme le DODAG. Il introduit une nouvelle option DIO qui Spécifie l'adresse qui doit être découvrir et enregistre le chemin traversé. La durée de vie du DODAG est restreinte au temps de la recherche du route. P2P-RPL permet d'utiliser Les routes sources ainsi que les routes hop-by-hop Pour spécifier les contraintes métriques pour les routes découvrir.

2.7.2.2 la mise en œuvre

P2P-RPL a été mise en œuvre sur Contiki, un système d'exploitation pour les réseaux de capteurs sans fil utilisés et développé activement par un large industriel et communauté universitaire.

2.7.2.3 Conclusion

Dans ce chapitre, nous avons étudié le routage dans les réseaux WSN, on a traite quelques protocoles de routage dans ces réseaux, on a constaté qu'il sont fragile face à l'utilisation d'une energie limitée par les capteurs et le passage à l'échelle qui est complexe à mettre en place. On a pris un autre type de réseaux qui est le P2P, afin d'exploiter les points commun entre les deux architectures ainsi que les avantage de ce dernier pour réaliser une amélioration dans un réseau WSN. Dans le chapitre suivant, nous allons proposer une Adaptation d'une architecture P2P pour une Application de Réseaux de Capteurs basée sur

une architecture structurée (CAN).

PROPOSITION D'UNE ADAPTATION D'UNE ARCHITECTURE P2P POUR UNE APPLICATION DE RÉSEAUX DE CAPTEURS

3.1 Introduction

Le déploiement des réseaux de capteurs sans fil soulève diverses questions. Quelle sera la spontanéité, l'adaptabilité du réseau ou encore la dynamique de la topologie. Ces éléments restent encore à définir. Compte tenu du nombre potentiellement important de capteurs participants et de leurs ressources réduites, il paraît indispensable de développer des solutions entièrement décentralisées, répartissant la charge entre les participants et permettant de passer à l'échelle. De nombreux procédés de systèmes distribués, et plus particulièrement ceux issus du modèle de pair à pair (P2P), apparaissent adaptables aux réseaux de capteurs sans fil. La mise en œuvre des techniques pair à pair pour les réseaux de capteurs est un axe de recherche stimulant, prometteur et non trivial. En effet, les caractéristiques de connexions et de vision locale permettent de mettre en évidence les similarités de ces deux domaines. D'autre part, les faibles ressources des capteurs et leur voisinage imposé par leur portée de communication rendent difficile une application directe. Plusieurs facteurs entrent en jeu, notamment l'adaptation des méthodes P2P aux contraintes énergétiques, et ainsi, une maximisation de la durée de vie du réseau. Les systèmes P2P sont de manière inhérente auto-organisables, résistants aux défaillances et capables de passer à l'échelle. De nombreux axes de recherche émergent à la frontière de ces deux domaines et se focalisent actuellement sur des objectifs tels que la diffusion, l'agrégation, la collecte mais aussi la base de données physiques.

Notre travail consiste à proposer et adapter une application pour les réseaux de capteurs

basée sur une architecture P2P et plus précisément sur l'architecture du protocole CAN.

3.2 Hypothèses et exigences

Afin de déterminer le travail que nous avons à faire, il est important de fixer le cadre dans lequel nous allons évoluer. Nous allons donc poser certaines hypothèses. Comme dans une architecture traditionnelle de réseaux de capteurs, les données sont toutes renvoyées vers un puits dans notre architecture nous souhaitons que les transmissions de données se font en mode structuré et non par inondation ou aléatoire afin d'éviter la consommation d'énergie dans des transmissions inutiles et ce en ce basant sur un type particulier de protocole (CAN).

3.2.1 Hypothèses de travail

Tout d'abord, nous supposons que tous les dispositifs ont la même pile de protocole. En effet, compte tenu du caractère totalement décentralisé de notre architecture et des faibles ressources des nœuds, il n'est pas acceptable d'utiliser certains nœuds comme relais de communication. Nous allons supposer que l'ensemble des nœuds du réseau sont adressables. C'est-à-dire qu'on pourra établir une communication directe entre deux nœuds. C'est un élément très important dans l'architecture décentralisée que nous voulons mettre en place, puisque cela va nous permettre d'assurer la fiabilité du réseau quel que soit l'état de celui-ci. Dans notre solution on utilise une architecture P2P structurée qui est l'architecture du protocole CAN pour un réseau de capteur dont une topologie hiérarchique. Car la structure hiérarchique devient une branche active pour le routage dans WSNs, puis qu'il exécute mieux que le plat. En effet, les protocoles plats ne s'adaptent pas bien avec les réseaux à grande échelle, tandis que les protocoles hiérarchiques offrent des caractéristiques avantageuses telles que la bonne évolutivité, agrégation de données, moins de charge, moins de consommation d'énergie, et ainsi de suite. Notre but dans ce travail est de mieux adapter CAN prenant en considération la contrainte d'énergie afin de maximiser la durée de vie du réseau par l'utilisation d'une technique de groupement et l'exploitation des précieuses caractéristiques du protocole structuré CAN de sorte qu'il accélère la transmission de données au puits, réalise l'équilibre de charge par l'élection dynamique des Cluster-Heads, il peut réduire la quantité de transmission de l'information à l'aide de l'agrégation de données. En outre, la solution proposée exploite l'efficace fonctions de recherche de CAN de base afin de maintenir les tables de routage du nœud dans le réseau.

3.2.2 Exigences

À partir des hypothèses que nous avons formulées au point précédent, nous allons ici définir quelques exigences auxquelles notre architecture devra répondre. Tout d'abord la plus importante, nous souhaitons que notre architecture soit indépendante de l'intergiciel. En effet de nombreux intergiciels pair à pair sont disponibles, chaque intergiciel apportant son lot d'avantages et d'inconvénients. Il est donc important de définir une architecture indépendante de l'intergiciel, cela permettra d'offrir une solution plus généraliste et laissera libre de choisir au mieux l'intergiciel le mieux adapté aux circonstances. D'un point de vue plus général, nous devons assurer la fiabilité du réseau. Nous devons aussi assurer une optimisation de la charge sur le réseau de manière à limiter la consommation d'énergie et par conséquent prolonger la durée de vie du réseau.

1. Topologie du réseau physique

La conception des protocoles pour le routage de données dans un RCSF est étroitement liée à la topologie du réseau considérée. Dans notre travail, on va s'intéresser à la topologie hiérarchique. Chaque nœud dans le réseau est caractérisé par un contrôle d'énergie.

2. Topologie du réseau virtuel

Le réseau virtuel est basé sur le réseau CAN qu'on a étudié dans le chapitre précédent. Les pairs (nœuds) du réseau CAN reposent sur un espace de coordonnées cartésiennes de dimension 2, sans rapport avec les coordonnées physiques des nœuds. Chaque pair dans le réseau est caractérisé par une variable d'énergie. Nous proposons d'organiser l'espace de routage en clusters (groupes), et pour chaque groupe on sélectionne le nœud dont l'énergie est plus élevée dans ce dernier comme chef de groupe (Cluster-Head). Les Cluster-Heads sont reliés entre eux selon le protocole CAN. Ce mécanisme limite les requêtes et attaches de cheminement de données aux Clusters-Heads.

3.3 Présentation de l'architecture

Dans cette partie nous allons présenter une architecture répondant à l'ensemble des exigences que nous avons citées précédemment. La FIGURE(3.1) montre le schéma global de notre architecture proposée.

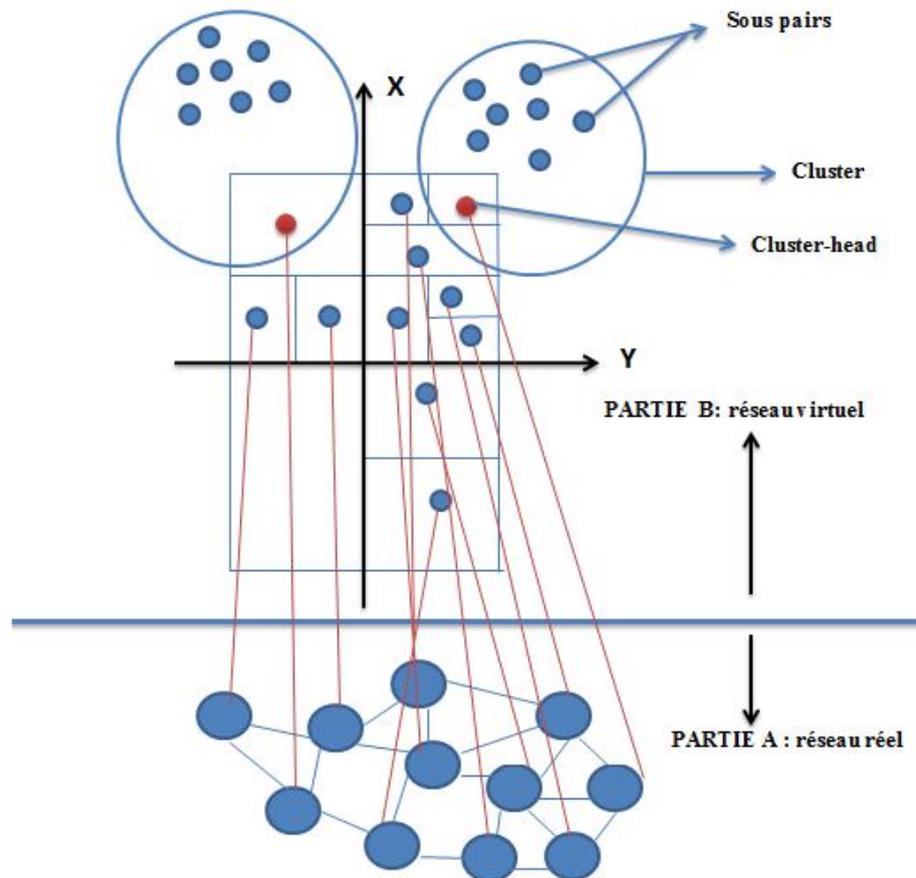


FIGURE 3.1 – Réseau réel et surcouche pair à pair (CAN adapté au RCSF) [20].

Notre architecture est composée de deux principales entités représentées sur la figure ci dessus, tout d'abord la partie (A) qui représente le réseau réel, constitué de plusieurs capteurs. La partie (B) représente la surcouche pair à pair réalisée à l'aide de l'intergiciel. Les traits reliant les deux couches représentent le fait que les équipements se connectent à l'intergiciel pair à pair. Cette architecture réussit le déploiement distribué des capteurs dans un domaine d'application. On utilisant le mécanisme de groupement, on créer plusieurs groupes (Clusters). A l'intérieur de chaque groupe on désigne un élu (Head) de groupe. Un autre point est très important c'est de connecter le Sink au réseau par la diffusion des message " Hello " aux nœuds appartenent à son proximité. Ce dernier contient l'adresse de station de base. Tous les voisins des Cluster-Heads reçoivent ce message et informent les autres Cluster-Heads non connectés qu'ils sont connectés directement au puits (Sink) les Cluster-Heads sont chargés de transmission des données au Sink. La FIGURE (3.1) montre un exemple de l'architecture proposée.

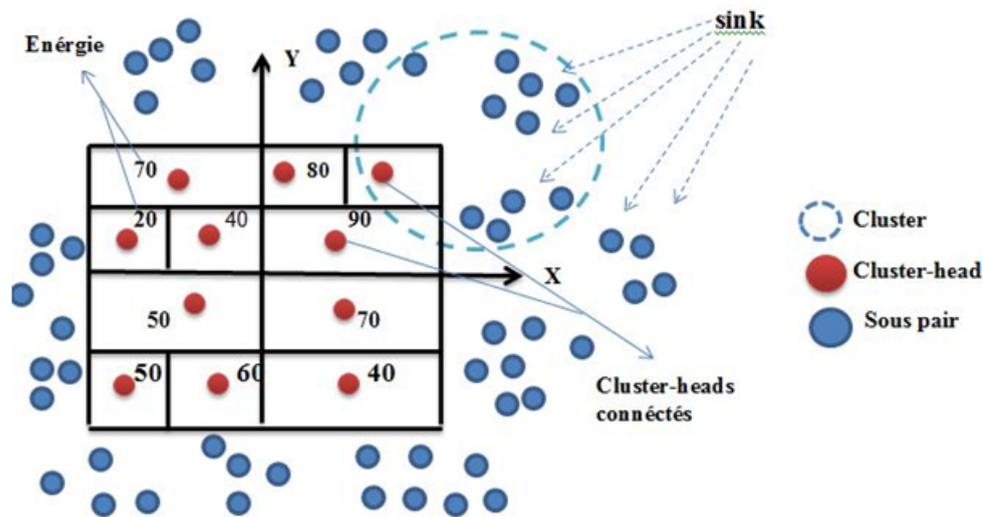


FIGURE 3.2 – Exemple de l'architecture proposée basée sur CAN (nombre de Cluster-Heads égale à 11 et nombre de Cluster-Heads connectés égale à 2) [20].

La FIGURE(3.1) illustre un exemple de l'architecture proposée, dans la figure (FIG3.2) avec plus de précision on considère le nombre de Cluster-Heads égale à 11 et nombre de Cluster-Heads connectés égale à 2.

3.4 Les différents algorithmes de l'architecture proposée basée sur CAN adapté au RCSF

3.4.1 Algorithme d'initiation

- : Début .
- : Construction de l'espace CAN
- : **Pour** $i=1$ a n **faire** // n est le nombre de clusters dans le système.
- : Déterminer le Cluster-Head de i eme cluster.
- : **Finpour** .
- : Creation du CAN.
- : Le Sink envoie le message HELLO pour tous les nœuds voisins. .
- : **Si** le nœud reçoit le message est un Cluster-Head **alors**
- : Le Cluster-Head devient Cluster-Head **connecté**.
- : Informer les Cluster-Heads non connectés par les Cluster-Heads connectés.

- : Sinon supprimer le message.
- : **Finsi.**
- : **Fin.**

Notations	Descriptions
SN	Le nœud source initialise une requête
SP	L'ensemble des sous pairs d'un cluster
CCH	cluster-head connecté
Chi	Ieme cluster-head
K	La requête cherchée
Z	l'ensemble des zones voisines
Zi	Ieme zone voisines
Ni	Nœud veut quitter le réseau
Nj	Nœud veut joindre le réseau
ENi	L'énergie restante d'un nœud
ENCH	L'énergie restante d'un Cluster-Head

TABLE 3.1 – Notations utilisées dans différents algorithmes.

3.4.2 Algorithme de processus de routage

- : **Debut**
- : **Si** SN appartient SP **alors**
- : SN transmettre la requête au Cluster-Head
- : **Sinon**
- : CHi recoit SN
- : **Fin si.**
- : **Si** CHi n'appartient pas à CCH **alors**
- : Déterminer aléatoirement le CCH
- : K recoit CCH-id.
- : **Si** k appartient aZ**alors** / Zl'ensemble des zones voisines de la zone de CHi.
- : Envoyer la donnée au nœud appartient a Zi/ Zi appartient a Z.
- : Envoyer la reponse au sink.
- : **Sinon**
- : Choisir le nœud (CHi) qui a une énergie plus élevé par rapport aux nœuds appartiennent a Z et proche de la destination.
- : K recoit CHi- id.

- : Répéter la procédure jusqu'à trouver K
- : **Fin si.**
- : Envoyer la reponse au sink.
- : **Sinon**
- : Envoyer la donnée au sink.
- : **Fin si.**
- : **Fin.**

Pour illustre le processus de routage, on donne un exemple dans la FIGURE(3.3) nous supposons que le nœud (13) veut envoyer une donnée au Sink. Premièrement le nœud (13) transmette la donnée au Cluster-Head (1), ce dernier choisira léatoirement un Cluster-Head connecté au Sinka partir de la liste des Cluster-Heads connectés, soit le nœud (11), après le nœud (1) transmet la donnée au nœud (11) qui est a son tour la transmette au Sink suivant les étapes suivantes.

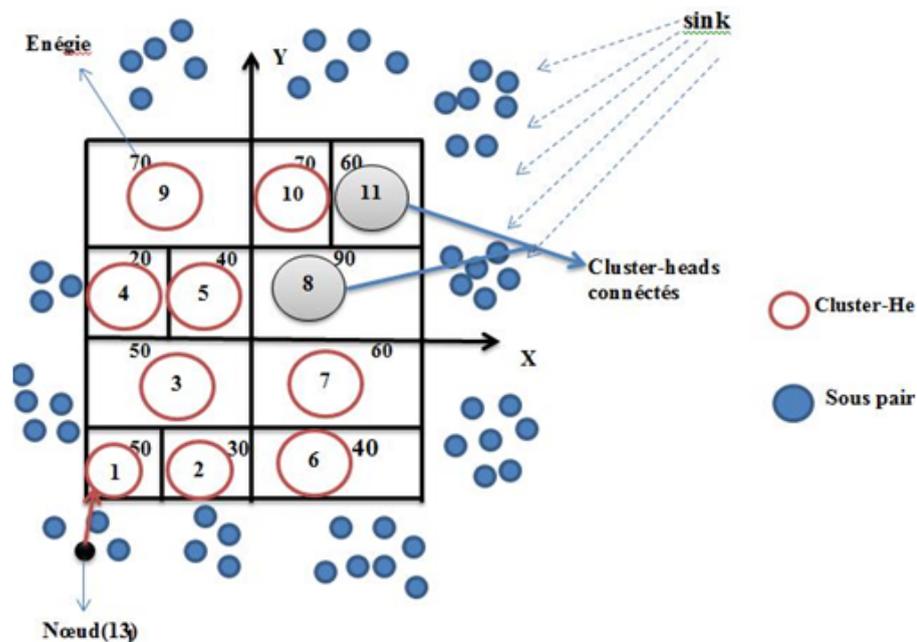


FIGURE 3.3 – Etape 1 : transmettre l'information au Cluster-Head (1) [20].

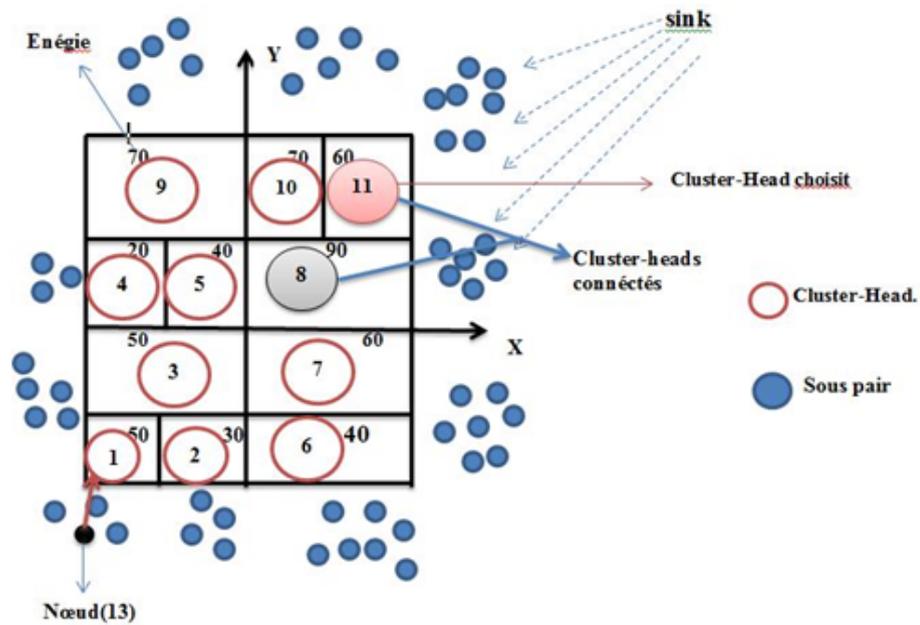


FIGURE 3.4 – Etape 2 : Choisir aléatoirement le Cluster-Head connectée (11) [20].

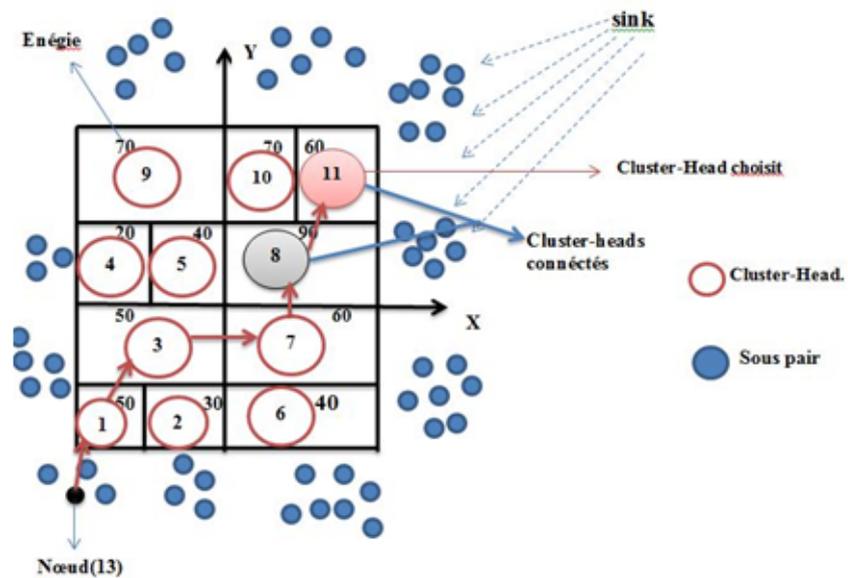


FIGURE 3.5 – Etape3 : les étapes à suivre pour router l'information au CCH (11) [20].

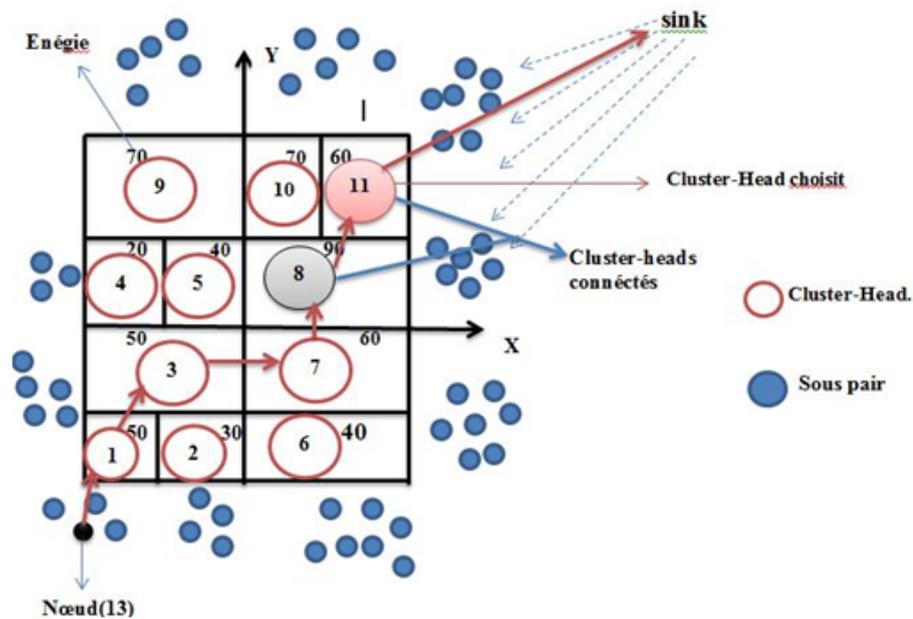


FIGURE 3.6 – Etape 4 : Envoie de l'information au Sink [20].

les quatre figures précédentes resument un Exemple de routage d'un message au Sink.

3.4.3 Algorithme d'insertion d'un nœud

- : **Debut**
- : N_i demande N_j de joindre le réseau.
- : N_i appartient au Cluster-Head de N_j
- : **Si** $E_{N_i} > E_{NCH}$ **alors**
- : Deviser Z_i en deux moitié / Z_i la zone de CH.
- : Insérer N_i dans la deuxième moitié.
- : mettre à jour les voisins.
- : **Fin si**
- : **Si** CH appartient a CCH **alors**
- : N_i informe tous les autres CCH que c'est un nouveau CCH.
- : **Fin si.**
- : **Fin.**

3.4.4 Algorithme de suppression d'un nœud

- : **Debut**
- : **Si** NiI veut quitter le réseau
- : **Si** Ni appartient à CH **alors**
- : Ni envoie l'emplacement de la requêtes aux nœuds qui sont dans son cluster.
- : CHi recoit Nj/ Nj c'est le nœud dont L'énergie la plus élevée dans le cluster.
- : mettre à jour les voisins.
- : **Fin si.**
- : **Si** Ni appartient à CCH**alors**
- : CCHi recoit CHi.
- : Ni informe tous les autres CCH que c'est un nouveau CCH.
- : **Fin si.**
- : **Sinon** Ni quitte le réseau
- : **Fin si.**
- : **Fin.**

3.5 Conclusion

Dans ce chapitre, nous avons proposé une adaptation d'une architecture P2P pour un réseau de capteur. Cette proposition basée sur le réseau(CAN). La solution proposé permet de traiter et gérer la consommation d'énergie lors de la transmission de données. La recherche des données se fait en mode structuré ce qui rend le routage des informations plus organisé et ne permet pas l'utilisation d'énergie dans le vide. La solution prend en considération le changement de la topologie du réseau avec des conditions pour une bonne gestion de réseau. Dans le chapitre suivant, on va décrire la conception de notre proposition et la concrétiser par une application.

CONCEPTION ET RÉALISATION

4.1 Introduction

La conception de notre travail consiste à créer un réseau P2P basé sur CAN adapté aux réseaux de capteurs. Ce travail repose principalement sur la conception d'un réseau virtuel (overlay network) interconnectant les membres présents dans le système et le mécanisme de recherche dans ce réseau. Dans ce chapitre on va traiter la conception et la réalisation d'une architecture CAN adapté aux réseaux de capteurs (proposée dans le chapitre 3) basée sur le CAN standard à partir d'un ensemble des nœuds appartenant au réseau.

4.2 Modélisation de l'application avec l'UML

Afin de représenter le modèle de notre application, nous avons opté pour l'utilisation de l'UML (Unified Modelions Language) qui est un langage de modélisation unifié, devenu désormais la référence en termes de modélisation objet. Il fournit un moyen astucieux permettant de représenter diverses projections d'une même représentation grâce aux vues qui se constituent d'un ou plusieurs diagrammes. Pour modéliser notre application, nous avons utilisé les diagrammes UML de cas d'utilisation, de séquence et de classes. Le choix de ces diagrammes étant justifié du moment qu'ils suffisent pour couvrir tous les aspects. Dans l'UML on distingue deux types de vues :

Les vues statiques : Elles comportent les diagrammes des cas d'utilisation, diagramme de classes, ainsi que le diagramme de déploiement.

Les vues dynamiques : Elles comportent les diagrammes de séquences et les diagrammes d'activités.

4.2.1 Diagramme de cas d'utilisation

Dans cette partie, nous allons évoquer l'essentiel des cas d'utilisation décrits sous la forme d'actions et de réactions entre l'utilisateur et le système, dans le but de définir les grandes fonctionnalités du système à modéliser. La FIGURE(4.1) représente le diagramme de cas d'utilisation.

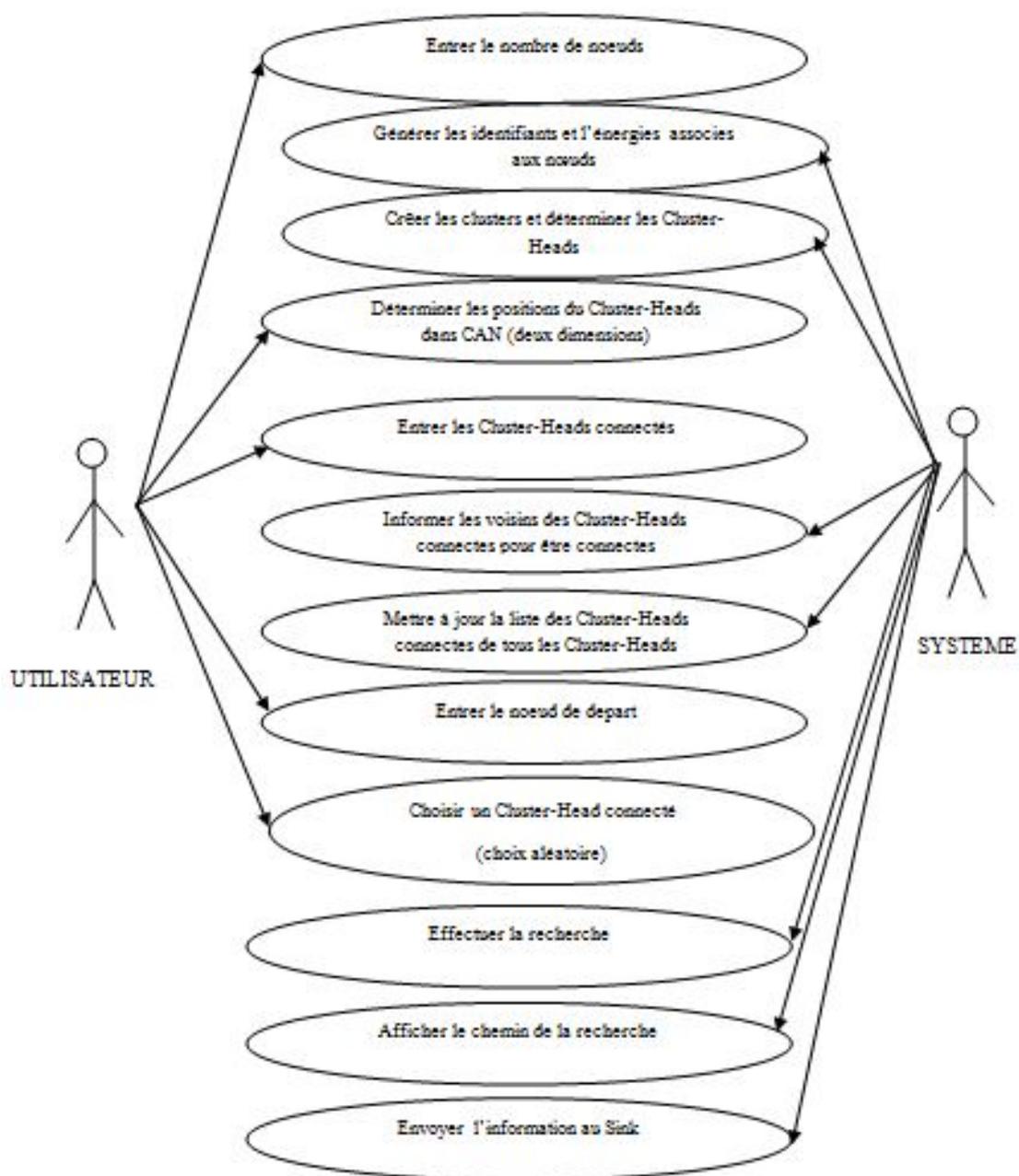


FIGURE.4.1: Diagramme de cas d'utilisation

4.2.2 Diagramme de séquence

Dans ce diagramme, on montre la séquence temporelle des tâches associées à chacun des cas d'utilisation. Comme les diagrammes de séquence sont toujours lus du haut vers le bas, ils illustrent l'ordre d'apparitions des tâches.

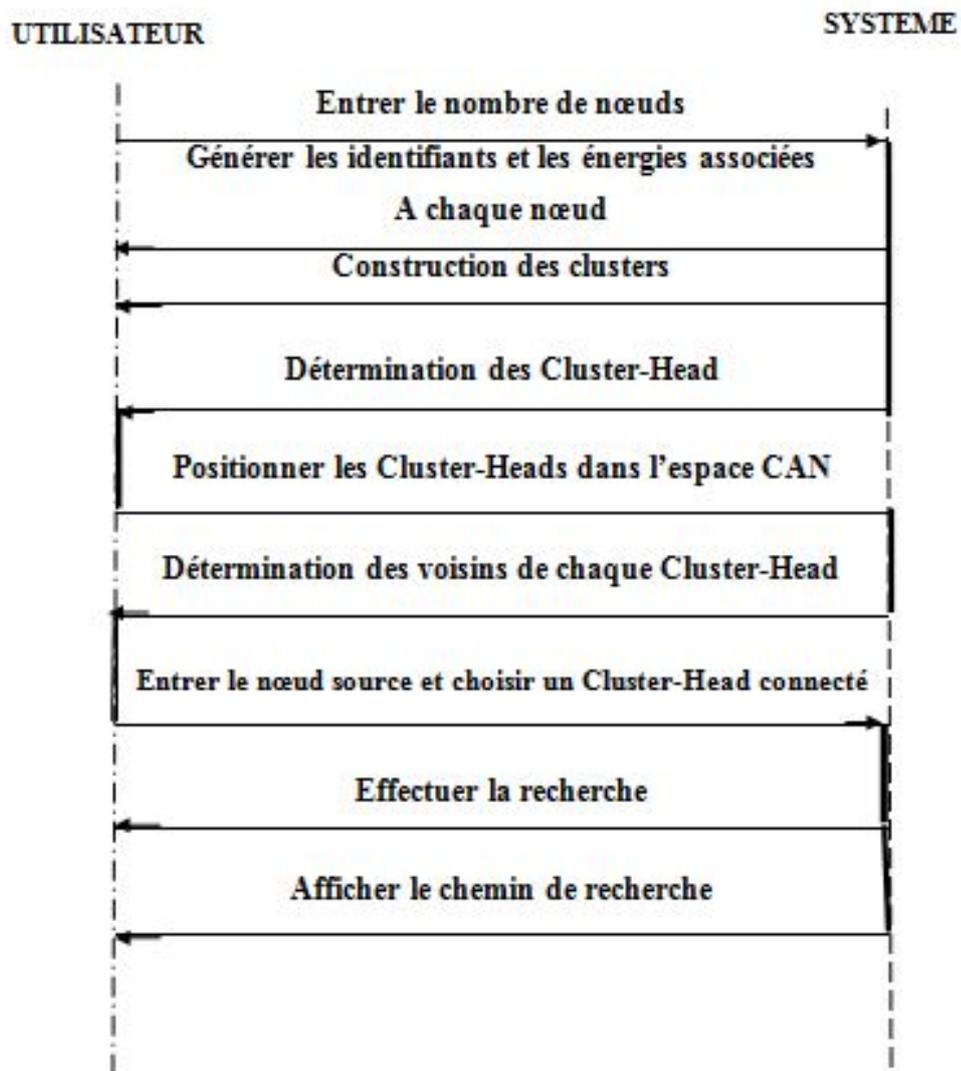


FIGURE 4.2: Diagramme de séquences.

4.2.3 Diagramme de classes

Dans ce diagramme, on montre la structure de système grâce à des classes et à des relations entre ces classes. La figure 4.3 représente le diagramme de classes.

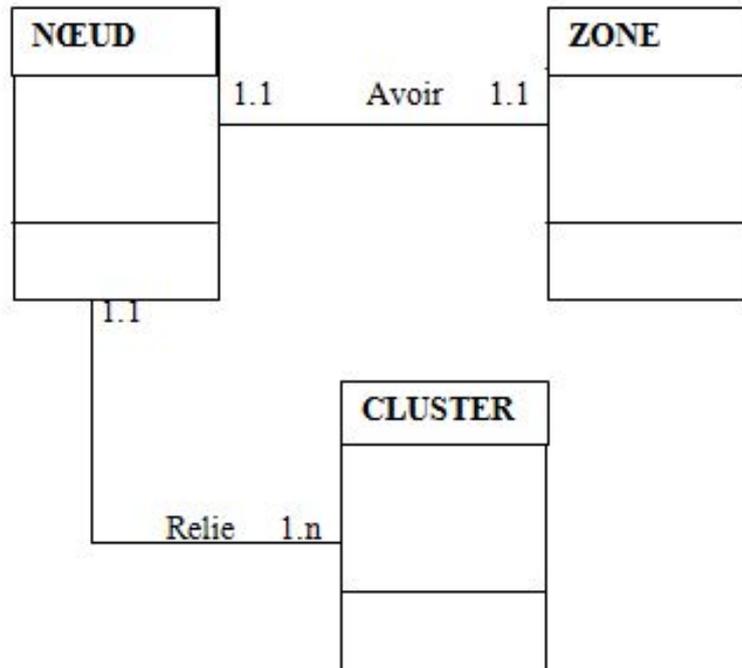


FIGURE 4.3: Diagramme de classes

- **La classe CLUSTER** : les méthodes de cette classe permettent de relier les nœuds et de gérer le routage de l'ensemble des nœuds de réseau.
- **La classe NOEUDS** : les méthodes de cette classe servent à la gestion des positionnements des nœuds dans l'espace CAN et déterminent les zones des noeudnoeudss.
- **La classe ZONNE** : les méthodes de cette classe servent à la création des zones et détermination des zones voisines, .

4.3 Réalisation

L'application conçue a pour but la réalisation d'un réseau basé sur le protocole l'architecture P2P CAN adapte adaptée aux RCSFs et permet la gestion de transmission de données et de l'énergie dans ce réseau.

4.3.1 Outils de développement

L'application a été développée avec le langage Java sous Windows Xp, en utilisant le logiciel Eclipse.SDK.3.5 (Java EE IDE) avec le kit de développement de base JDK 6 (Java Développement Kit) sous la licence Sun Microsystems. Le kit de développement comprend plusieurs outils, parmi lesquels : -javac : le compilateur Java. -java : un interpréteur d'application (machine virtuelle). -javadoc : un générateur de documentation. -jar : un compresseur de classes java. Les systèmes Java consistent en plusieurs parties : un environnement, le langage, les interfaces de programmation d'application(API) et divers bibliothèques de classes. Notre choix s'est porté sur Java, car il est un langage de programmation puissant et facile, convenant parfaitement à la construction des différentes applications et des systèmes informatiques.

4.3.2 Interface de l'application

a) Interface de page d'accueil

Lorsqu'on exécute le programme, une fenêtre d'accueil apparaît. Elle contient un Menu qui permet d'accéder à la fenêtre principale de l'application par le menu Application FIGURE(4.4).

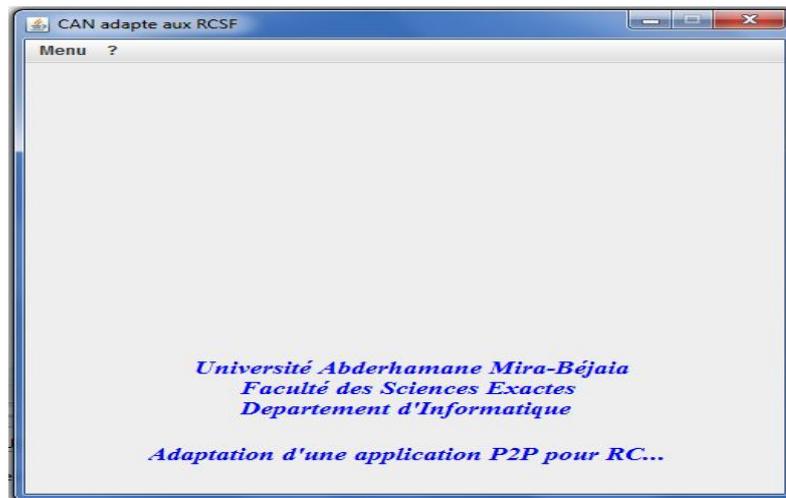


FIGURE 4.4: Fenêtre d'accueil

Le menu Quitter permet à l'apparition de la par le biais d'une boîte de dialogue pour la confirmation ou non de vouloir quitter l'application FIGURE(4.5).

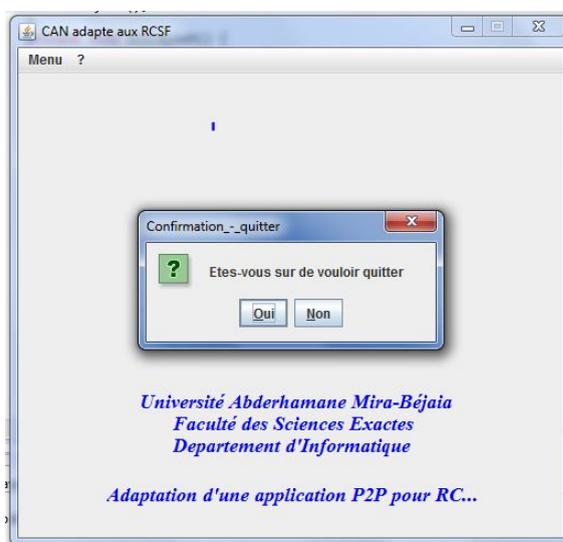


FIGURE 4.5: Interface de la boîte de dialogue

b) Interface de la construction du réseau On introduit le nombre de nœuds (inferieur à 50, limite de notre application), on clique sur le bouton Activer pour générer aléatoirement les identifiants des nœuds et leurs énergies initiales FIGURE(4.6).

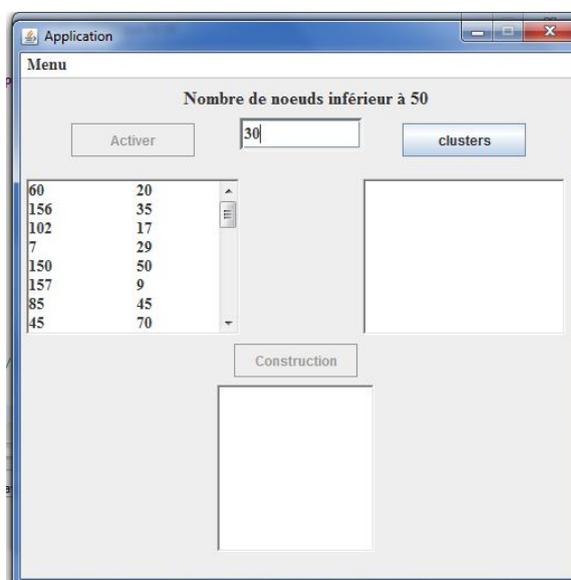


FIGURE 4.6: Génération aléatoire des identifiants et des énergies des nœuds

Lorsqu'on clique sur le bouton clusters, les nœuds seront groupés en cluster (04 nœuds par cluster selon notre application). Chaque cluster contient un Cluster-Head (le nœud possède l'énergie la plus élevée par rapport aux nœuds appartient à son cluster) . Le bouton Construction construit le réseau en déterminant les voisins de chaque Cluster-Head FIGURE(4.7).

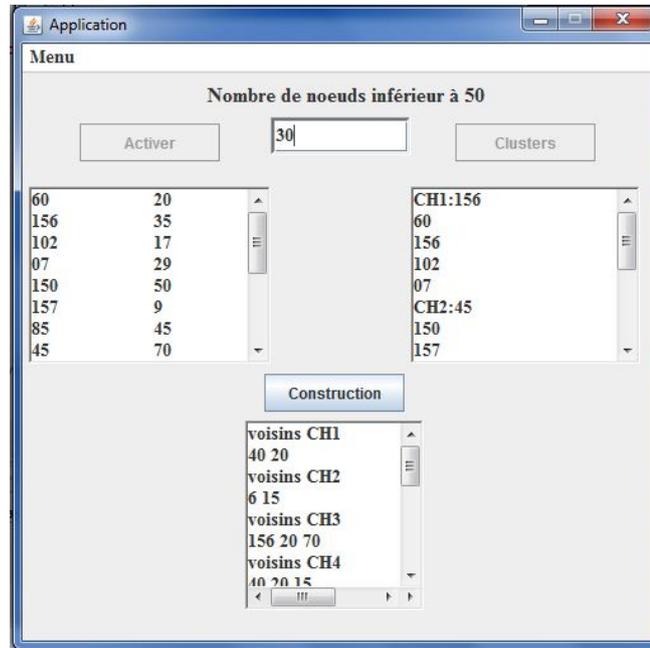


Figure 4.7: Construction de réseau

c) **Interface de la recherche** On accède à la fenêtre Recherche (FIG.4.8) par le menu Recherche de la fenêtre précédente. Pour la recherche, on introduit le nœud du Départ et le nœud d'Arrivée (Cluster-head connecté choisi). Le bouton Rechercher affiche le parcours détaillé et le chemin parcouru FIGURE(4.8).



FIGURE 4.8: fenêtre de la recherche

4.4 Conclusion

Dans ce chapitre, nous avons décrit les différentes étapes de la conception de notre application qui est basée sur le protocole CAN adapté au réseau de RCSF (proposé dans le chapitre 3) suivant une modélisation UML. En second lieu, après avoir décrit les outils utilisés pour la réalisation de cette application, nous avons présenté les fonctionnalités de cette dernière ainsi que les résultats qu'elle fournit à savoir : la gestion de l'énergie afin de maximiser la durée de vie du réseau ainsi que l'utilisation des cluster-Heads afin de bien gérer la transmission des données du CSF proposé.

Conclusion générale et Perspectives

Le coût réduit et la rapidité de déploiement, la tolérance aux fautes, l'auto configuration et l'auto organisation sont des paramètres qui caractérisent un bon réseau. Dans les réseaux de capteurs, les chercheurs ont pensé à la miniaturisation des nœuds constituant de tels réseaux afin de les utiliser dans les applications de sécurité, militaire et la médecine interne.

Une telle technique présente des contraintes d'énergie et de capacités des stockages très insuffisantes qu'on doit bien gérer afin de prolonger le plus long possible la durée de vie du réseau. Pour remédier à ce problème, dans la littérature, ils existent beaucoup de propositions qui interviennent pendant les trois opérations : la capture, le traitement et la communication. Cependant, c'est la phase de communication qui consomme énormément d'énergie.

La diffusion d'information dans les WSNs se fait souvent par inondation (flooding) induisant une forte consommation d'énergie, surtout autour du point d'entrée de la donnée diffusée. L'utilisation d'un protocole épidémique P2P est une solution envisageable pour diffuser de manière équitable l'information tout en conservant une rapidité de convergence. Dans ce cadre, l'objectif de notre travail était de proposer un modèle qui limite la consommation d'énergie dans un modèle P2PWSN basé sur le protocole CAN. Le schéma proposé étend la durée de vie du réseau en exploitant le recouvrement des tables de hachages distribuées DHTs et l'utilisation de l'approche de clustérisation où les clusters heads effectuent le processus de routage.

En guise de perspectives, nous envisageons de faire Une nouvelle méthode de recherche pour une meilleure gestion des liens supplémentaires afin de diminuer plus le nombre de sauts.

- Gestion des arrivées et des départs fréquents des nœuds.
- Transfert de fichiers.
- Gestion des requêtes complexes.

Bibliographie

- [1] S. Athmani. Protocole de sécurité pour les réseaux de capteurs sans fil, thèse de magistère en informatique, université hadj lakhder de batna, département informatique. 2010.
- [2] Y. Busnel. Systèmes d'information pair-à-pair pour les réseaux de capteurs larges échelles. (projet paris). thèse financée par le ministère de la recherche. 2002.
- [3] B.Amina B.Zeyneb. Mise en place d'un réseau de capteurs sans fil pour l'irrigation intelligente, mémoire de master, université de tlemcen.
- [4] N. Cardenas C. B. Abbas, R. González and L. J. G. Villalba. "a proposal of a wireless sensor network routing protocol". *springer science and business media. telecommunication systems*. pp. 61-68. March 2008.
- [5] R. Govindan C. Intanagonwiwat and D. Estrin. "directed diffusion : A scalable and robust communication paradigm for sensor networks". in the proceedings of the 6th annual acm/ieee international conference on mobile computing and networking (mobicom'00), boston, ma. August 2000.
- [6] C.Castelluccia, E.Mykletun, and G.Tsudik. Efficient aggregation of encrypted data in wireless sensor networks , thèse de doctorat, université de limonge, spécialité informatique. pages 109–117.
- [7] J. Heidemann D. Estrin, R. Govindan and Satish Kumar. Next century challenges : scalable coordination in sensor networks". in the proceedings of the 5th annual acm/ieee international conference on mobile computing and networking (mobicom'99), seattle, wa. August 1999.
- [8] Mukul Goyal Emmanuel Baccelli, Matthias Philipp. "the p2p-rpl routing protocol for ipv6 sensor networks : Testbed experiments". 19th international conference on software, telecommunications and computer networks, split, croatia. pp.1 - 6. 2011.
- [9] F.Abdelfatah. Développement d'une bibliothèque de capteurs, projet de fin d'études pour l'obtention du diplôme de master en informatique , université de montpellier2 , spécialité informatique,. Juin 2015.

- [10] P. T. Kuruganti H. Qi and Y. Xu. "the development of localized algorithms in wireless sensor networks". published on 202 sensors issn, pp. 1424- 8220. 22 July 2004.
- [11] <http://membres.lycos.fr/consuter> fevrier 2017.
- [12] <http://www.Capit.Net/> consuter Mars 2017.
- [13] <http://www.labosso.Com/> consuter Avril 2017.
- [14] <http://www.Univ-mlv.Fr/> Boubaker-Lelion-Pierron-P2P.Pdf/ consuter Mars 2017.
- [15] A.E.Kamal J.N.Al-Karaki. Routing techniques in wireless sensor networks : a survey wireless communications, journal of communication and network. pages 6–28, 2004.
- [16] A. Brandt R. Cragie J. Martocci M. Goyal, E. Baccelli. "reactive discovery of point-to-point routes in low power and lossy networks," ietf internet draft draft-ietf-roll-p2prpl. May 2011.
- [17] J. Hui O. Gnawali J. Ko P. Levis, T. Clausen. "rfc6206 : The trickle algorithm," internet engineering task force(ietf) request for comments. March 2011.
- [18] G. Paroux. "une plate-forme pour les échanges p2p sur les réseaux mobiles ad hoc", rennes, pp.275-282. Nov 2005.
- [19] P.Kadionik. Projet avancé : réseau de capteurs sans fils ,université de paris, spécialité électroniques.
- [20] V. Rodoplu and T. H. Ming. "minimum energy mobile wireless networks". iee journal of selected areas in communications, vol. 17, no. 8, 1999, pp. 1333-1344. 2003.
- [21] P. Thubert et al T. Winter. "rpl : Ipv6 routing protocol for low power and lossy networks," ietf internet draft draft-ietf-roll-rpl. March 2011.
- [22] Y.Challal. Les réseaux de capteurs (wireless sensor networks),université de technologie de compiègne, france. 2016.
- [23] B.Ramamurthy Y.wang, G.Attebury. A survey of security issues in wireless sensor networks, in IEEE communication survey tutorials, pp. 2456-2460. Janvier 2006.
- [24] F. Guenounou Z. Maouche. "problèmes de recherche et de localisation des services dans les réseaux p2p .cas skip graphs", thème de fin de cycle pour l'obtention du diplôme d'ingénieur d'état en génie informatique,université abderrahmane mira. Bejaia 2010.

Summary

A current need to use wireless technologies to extract information from hostile and critical environments is the reason for the appearance of wireless sensor networks. However, the sensors' miniaturization creates trouble for networks' lifetime. We propose a limited energy consumption model for P2P wireless sensor networks, which takes advantage of the Can protocol and adapts it so that it will be suitable for P2P WSNs.

Can in its basic form optimizes the path length by minimizing the number of hops, but it does not optimize the energy, since it was designed for wired networks. The key idea of our proposal is to optimize both the number of hops and the energy consumption to provide an energy-efficient routing scheme for P2P WSNs.

Keywords Wireless sensor networks , Peer-to-peer systems , Routing , Can

Résumé

Le besoin du monde actuel d'utiliser les technologies sans fil pour extraire des informations à partir des milieux très sensibles, hostiles et inaccessibles a fait appel au réseau de Capteurs. Cependant, la miniaturisation des capteurs nécessite des mécanismes de conservation d'énergie de ces derniers afin d'étendre la durée de vie du réseau. Nous proposons un modèle limité de consommation d'énergie pour les réseaux de capteurs sans fil basée sur l'architecture p2p.

Ces réseaux bénéficient des propriétés de Can, et l'adapte de sorte qu'elle soit appropriée à p2p WSNs. Can sous sa forme de base optimise la longueur du trajet par la minimisation du nombre de saut mais n'optimise pas l'énergie puisque il a été conçu pour les réseaux câblés. L'idée principale de notre proposition est d'optimiser le nombre de saut et la consommation d'énergie. Pour fournir un plan de rendement optimum de cheminement pour P2P WSNs.

Mots clés réseaux de capteurs sans fil, les systèmes p2p, routage, Can.