REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

MEMOIRE DE FIN D'ETUDES

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

Spécialité: SYSTEMES INFORMATIQUES

THEME

Proposition d'un nouveau protocole de routage avec agrégation des données pour contrôler la congestion dans un réseau de capteurs sans fil

Réalisé par : M^{r} TWAGIRAYEZU Jean Bosco

Soutenu le 13 juillet 2011 devant le jury :

M^{r} DAOUI Mohamed (Président)

M^{me} AOUDJIT Rachida (Examinateur)

M^{r} TAHANOUT Mohamed (Examinateur)

M^{r} LAGHROUCHE Mourad (Promoteur)

Promotion 2010-2011

REMERCIEMENTS

Mes remerciements vont en premier lieu à Monsieur LAGHROUCHE Mourad d'avoir accepté de diriger ce travail et pour ses qualités d'encadrement.

Je tiens également à remercier les membres du jury pour avoir accepté de juger ce travail.

Enfin tous ceux qui m'ont encouragé, aidé et contribué à la réalisation de ce projet.

Un réseau de capteurs sans fil est formé par un ensemble d'équipements électroniques de petite taille appelés capteurs. Ces derniers collectent les données du phénomène étudié, effectuent un certain nombre de traitement sur ces données et les transmettent à un centre de collecte via le canal sans fil en mode multi-sauts. Les réseaux de capteurs sont utilisés dans plusieurs domaines, allant du domaine militaire au domaine médical en passant par plusieurs autres domaines tels que l'industriel, l'écologique etc. Les capteurs sont les éléments limités en ressources, principalement en ressources énergétiques, de calcul, de stockage et en puissance de transmission. L'une des conséquences de cette limite de ressources est qu'un capteur peut se trouver dans un état où il ne peut pas transmettre ou traiter les données .On parle de l'état de congestion et on dit que le réseau est congesté. Cette congestion peut être liée au mode de déploiement des capteurs qui fait que plusieurs capteurs collectent et transmettent les données fortement corrélées ou redondantes. La congestion dégrade les performances du réseau et doit donc être évitée.

Dans ce travail nous avons proposé un nouveau protocole de routage utilisant la technique d'agrégation des données pour contrôler la congestion dans un réseau de capteurs sans fil. Les résultats de simulation ont montré que suivant le degré de corrélation entre les données du phénomène observé, notre protocole permettra d'éviter la congestion.

Mots clés : *Réseau de capteurs sans fil, agrégation des données, contrôle de congestion, protocole de routage.*

Table des matières

Table des matières

Liste des figures

LISTE DES FIGURES

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Chapitre V

Liste des tableaux

LISTE DES TABLEAUX

Chapitre I

Chapitre V

Introduction générale

Depuis quelques décennies, le besoin d'observer et de contrôler des phénomènes physiques tels que la température, la pression ou encore la luminosité est essentiel pour de nombreuses applications industrielles et scientifiques. Par exemple, dans le domaine de l'écologie, la surveillance de polluants comme l'Ozone, le NO2 ou encore le CO2, pourrait considérablement augmenter la qualité de vie dans les villes [91].

Les progrès réalisés ces dernières décennies dans les domaines de la microélectronique et des technologies de communication sans fil, permettent de produire à un coût raisonnable des composants de quelques millimètres cubes de volume. Ces derniers, appelés micro-capteurs sont dotés des moyens qui leur permettent de collecter, de stocker, de traiter et de transmettre les données collectées à un centre de collecte via le canal sans fil. Ainsi, Il devient facile de déployer les capteurs dans des endroits parfois même difficiles d'accès, pour former les réseaux de capteurs sans fil, en évitant l'usage des câbles utilisés depuis longtemps qui avaient comme principaux défauts d'être encombrants et couteux.

En raison des contraintes de miniaturisation, les capteurs sont généralement dotés de ressources très limitées en termes de capacité de calcul, d'espace de stockage de données, de débit de transmission et d'´energie embarquée. L'une des conséquences de cette limitation de ressources est que, dans un réseau de capteurs sans fil, tous les capteurs ne peuvent pas atteindre le collecteur central. Les nœuds proches du collecteur joueront donc le rôle de relais entre le collecteur et les nœuds éloignés.

Etant limité en ressource et jouant le rôle de relais, un capteur peut se trouver dans un état où il ne peut pas traiter ou transmette les données. Dans cet état on dit que le capteur en question est congesté, et on parle d'état de congestion. Quand la congestion se présente, elle cause la dégradation des performances d'un réseau avec une perte de données et une consommation inutile de la bande passante. Pour être performant, les protocoles conçus pour les réseaux de capteurs, principalement les protocoles de routage doivent être capables de contrôler la congestion.

Dans un réseau de capteurs sans fil, les capteurs sont souvent densément déployés dans des zones à surveiller. Ainsi, les capteurs proches peuvent capter la même donnée ou les données très proches. En transmettant ces données redondantes aux nœuds relais, elles peuvent être une source de congestion. La mise en place des techniques capables d'éviter ce type de congestion devient donc une nécessité dans le domaine de la recherche.

Pour répondre à cette problématique, nous proposons un nouveau protocole de routage utilisant la technique d'agrégation des données. Cela consiste à combiner les données redondantes ou fortement corrélées afin de réduire le nombre de messages transmis par les capteurs et donc pouvoir contrôler la congestion.

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

Dans le premier chapitre, nous verrons les concepts généraux relatifs au domaine des réseaux de capteurs sans fil. Dans le deuxième chapitre, nous ferrons une étude bibliographique sur les protocoles de routage proposés pour ce genre de réseaux. Avant de proposer notre protocole dans le cinquième chapitre, nous ferrons un état de l'art sur les techniques de contrôle de congestion pour les réseaux de capteurs dans le troisième chapitre et sur les techniques d'agrégation des données dans le quatrième chapitre.

Chapitre I

I.1 Introduction

Dans le monde industriel, un capteur est un dispositif qui transforme une grandeur physique observée (température, pression, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur).Grâce aux avancées technologiques de ces dernières années, principalement dans le domaine de la miniaturisation, les capteurs sont devenus des éléments de très petite taille. Ils sont ainsi dotés de moyens leur permettant: de stocker les résultats de leurs observations, d'effectuer un certain nombre de traitement sur ces résultats et de les transmettre au monde réel via une communication sans fil. Toutes ces caractéristiques apportent au monde scientifique l'opportunité de déployer les capteurs à grande échelle dans des milieux, parfois difficiles voire impossible d'y avoir l'accès, pour surveiller, collecter et transmettre les données collectées à un système capable de les analyser et prendre (si possible ou nécessaire) les décisions. Ce déploiement aboutit à un réseau de capteurs sans fils(RCSF), ou wireless sensor network(WSN) en anglais.

Pour leurs caractéristiques très variées et leur faible coût de production, les RCSFs sont très utilisés dans plusieurs domaines, allant du domaine militaire au domaine médical, en passant par l'industrie, l'écologie, la domotique, l'agriculture de précision, etc. Pour cela, les RCSFs forment un domaine de recherche très vaste et en pleine croissance. Ainsi, mener un travail de recherche dans ce domaine nécessite la connaissance d'un certain nombre de concepts généraux.

Dans la suite de ce chapitre, nous verrons ces concepts à travers un certains nombre de points et un plan méthodologique que nous avons adopté. Nous commencerons par donner quelques définitions d'un capteur, l'architecture matérielle et voir comment ces capteurs sont déployés pour former des réseaux de capteurs sans fil, qui seront classifiés et comparés aux réseaux ad hoc classiques. Ensuite, les facteurs et contraintes de conception des RCSFs ainsi que les domaines d'applications de ces réseaux seront étudiés. Avant de conclure ce chapitre, nous ferrons une étude un peu détaillée sur les protocoles de communication destinés aux réseaux de capteurs sans fil.

I.2 Définitions

I.2.1 Un capteur

Un capteur est le dispositif qui transforme une grandeur physique observée (température, pression, humidité, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur) [18]. Pour cela, il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre.

I.2.2 Un capteur intelligent

Le terme *capteur intelligent* (smart sensor ou intelligent sensor) a été utilisé dans l'industrie des capteurs pour désigner des capteurs qui ne fournissent pas seulement des mesures, mais aussi une fonctionnalité aux mesures spécifiques [19]. Par rapport à un capteur classique, un capteur intelligent intègre de nombreux éléments électroniques additionnels, ainsi que des unités programmables et des aspects logiciels nécessaires au traitement des données, aux calculs, à la communication numérique [18].Il est donc caractérisé par [19] sa capacité à effectuer une collecte des mesures ,les traiter et à les communiquer au monde extérieur.

La deuxième définition étant la plus utilisée pour définir un capteur, on constate que le mot capteur est utilisé par abus de langage pour désigner un capteur intelligent. Pour ne pas rompre avec la terminologie usuelle, dans le reste de notre travail, lorsque nous parlerons d'un capteur, nous sous-entendrons un capteur intelligent.

I.3 Architecture matérielle d'un capteur

Suivant le type d'application, il existe une multitude de capteurs sur le marché :les capteurs de température, d'humidité, de pression, etc. Cependant, malgré cette diversité apparente, ils restent dotés d'une architecture matérielle similaire. Un capteur est composé principalement d'une unité de : captage, traitement, stockage, communication, et énergie. Des composants additionnels peuvent être ajoutés selon le domaine d'application [20], comme par exemple un système de localisation tels qu'un GPS (Global Positioning System), un générateur d'énergie (exemple : cellules solaires) ou un mobilisateur lui permettant de se déplacer. Ces éléments principaux et optionnels (représentés par les traits discontinus) sont visibles sur la figure I.1.

1. Unité d'énergie

Un capteur est muni d'une source d'énergie, généralement une batterie [21], pour alimenter tous ses composants. Les batteries utilisées sont soit rechargeables ou non. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs et donc d'un réseau de capteurs.

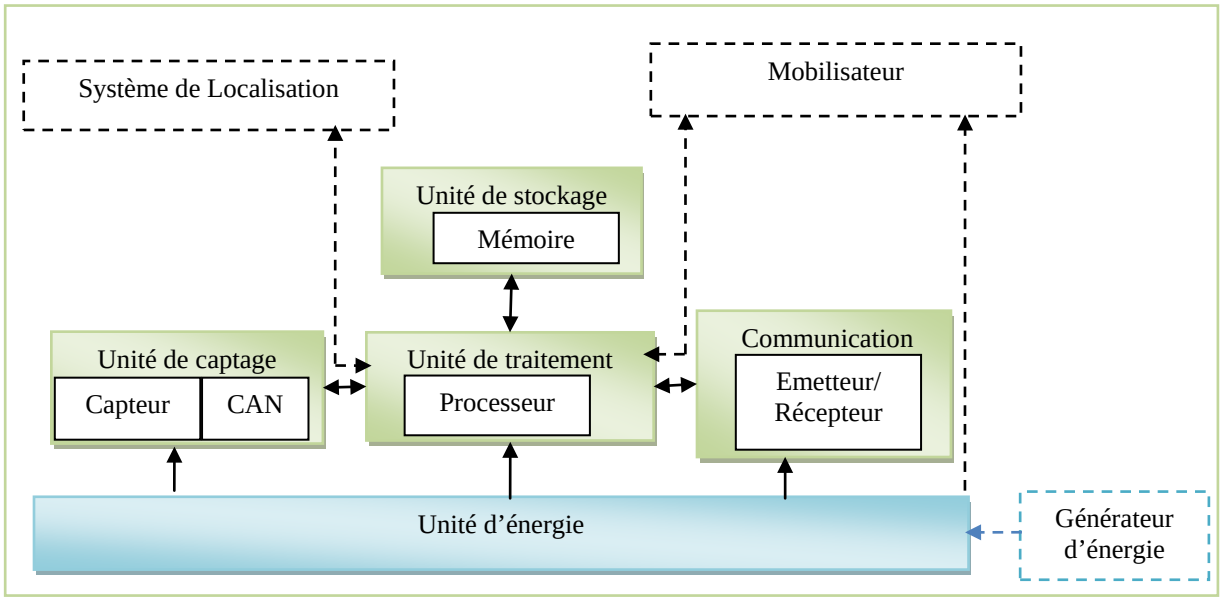

Figure I. 1. Architecture matérielle d'un capteur.

2. Unité de captage

La fonction principale de l'unité de captage est de capturer ou mesurer les données physiques à partir de l'objet cible. Il est composé de 2 sous-unités : le récepteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique(CAN). Ce dernier transforme ces signaux en données numériques et les transmet à l'unité de traitement. Un capteur peut avoir un ou plusieurs unités de captage [22].

3. Unité de traitements (processeur)

Il recueille des données des de l'unité de captage ou d'autres capteurs, effectue un traitement sur ces données (si nécessaire) et décide quand et où les envoyer. Il doit exécuter des programmes et des protocoles de communication différents. Les types de processeurs qui peuvent être utilisés dans un capteur incluent le Microcontrôleur, les DSP (Digital Signal Processors), les FPGA (Field Programmable Gate Array) et les ASIC (Application Specific Integrated Circuit) [22]. Parmi toutes ces alternatives, le Microcontrôleur a été le processeur le plus utilisé pour les capteurs à cause de sa flexibilité à être reliés à d'autres composants (comme par exemple l'unité de communication), à son bon prix et sa faible consommation énergétique [22] [23] [24].

4. Unité de communication

Cette unité est responsable de toutes les émissions et réceptions de données via un support de communication sans fil. Les différents choix de média de transmission sans incluent la Radiofréquence (RF), le Laser et l'Infrarouge [22].

5. Unité de stockage(Mémoire)

L'unité de stockage inclut la mémoire de programme (dont les instructions sont exécutées par le processeur) et la mémoire de données (pour conserver des données fournies par l'unité de captage et d'autres données locales). La taille de cette mémoire est souvent limitée essentiellement par les considérations économiques et s'améliorera aussi probablement au fil des années [25].

I.4 Architecture d'un réseau de capteurs sans fil

Un réseau de capteur sans fil(RCSF),ou Wireless Sensor Network (WSN) en anglais, consiste en un ensemble de nœuds capteurs variant de quelques dizaines d'éléments à plusieurs milliers, placés de manière plus ou moins aléatoire (par exemple par largage depuis un hélicoptère),dans une zone géographique appelée zone de captage, ou zone d'intérêt, afin de surveiller un phénomène physique et de récolter leurs données d'une manière autonome. Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées vers un nœud collecteur appelé nœud puits (*sink* en anglais), ou station de base (base station). Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur central « Gestionnaire de tâches » pour analyser ces données et prendre des décisions. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de

données requises, puis récolter les données environnementales captées par le biais du nœud puits.

En plus des nœuds capteurs, le modèle peut introduire les super-nœuds, appelés des passerelles(Gateways) [26]. Ces derniers possèdent une source d'énergie importante, la capacité de traitement et de stockage plus élevées comparativement aux nœuds capteurs [26].Il peuvent ainsi être utilisés pour exécuter les tâches plus complexes comme la fusion des donnés issues des capteurs d'une même zone. Dans le cas le plus simple, les nœuds capteurs seront dans le voisinage direct du puits (communication à un a un saut). Cependant, dans le cas d'un réseau à grande échelle, ils ne sont pas tous dans le voisinage du puits et les données seront acheminées du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication multi-sauts comme l'illustre la figure I.2 ci-après

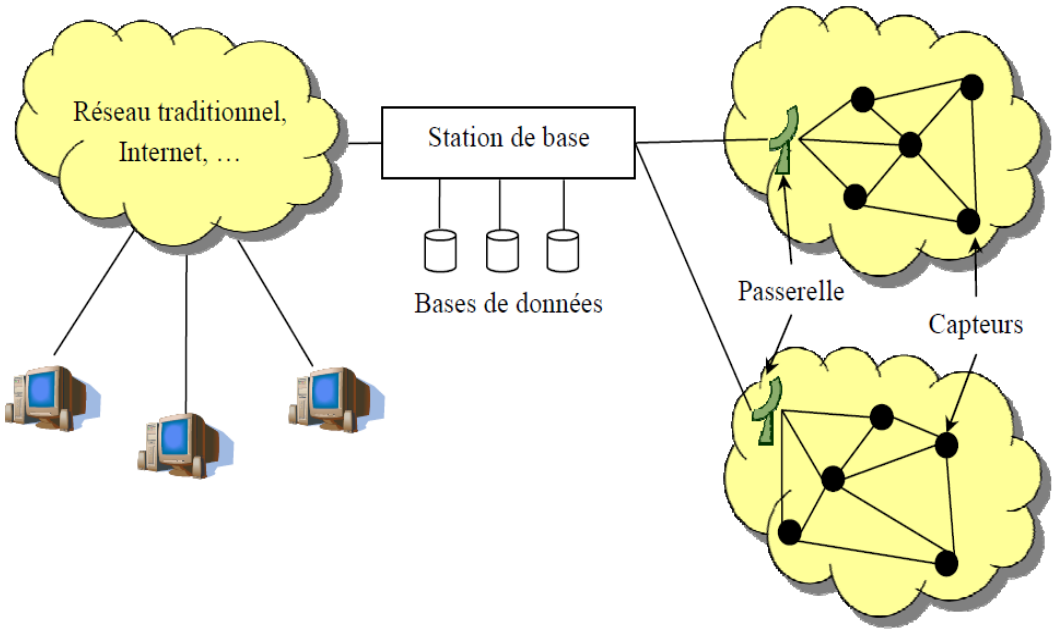

Figure I. 2 . Architecture d'un RCSF

I.5 Classification des RCSFs

Il existe plusieurs critères pour classer les réseaux de capteurs [1][2] [27].En effet, pour chaque type d'application, ces réseaux ont des caractéristiques différentes. Ils se distinguent par le mode d'acquisition et de livraison des données au puits, la distance entre les nœuds capteurs et le puits, le modèle de mobilité dans le réseau, les capacités des nœuds du réseau, etc.

1. Selon le mode d'acquisition et de livraison des données au puits

Dans les réseaux de capteurs, le modèle d'acquisition et de livraison des données au puits dépend de l'application et de ses exigences. Il peut être : continu*(time-driven),* événementiel *(event-driven),* à base de requête *(query-driven),* ou hybride.

Dans le modèle continu, les nœuds doivent périodiquement (intervalle de temps constant) réveiller leurs émetteurs pour envoyer les données captées au puits. Le type d'application visé concerne les applications de type "surveillance" où le but principal est d'avoir une information régulière de la zone surveillée. Dans le modèle orienté événements, les capteurs envoient leurs mesures seulement lorsqu'il y a un évènement qui se produit. Ce type de modèle est recommandé pour les applications de surveillance d'évènements critiques où le but principal est l'obtention d'une information sur l'évènement le plus rapidement possible. Dans le modèle orienté requêtes, les capteurs mesurent des phénomènes et stockent ces mesures dans leur mémoire. Ils envoient ces mesures seulement lorsqu'ils reçoivent des requêtes de la station de base. Le modèle hybride est une combinaison des trois modèles précédents.

2. Selon la distance entre les nœuds capteurs et le puits

Dans cette classification, on distingue les réseaux multi-sauts (multi-hop WSN) des réseaux à un seul saut (Single-hop WSN). Dans un réseau de capteur à un simple saut, les nœuds capteurs sont dans le voisinage immédiat du puits. Ils envoient alors leurs données captées directement au puits sans passer par aucun autre nœud intermédiaire. Dans le réseau de capteur multi-sauts, à distance entre les quelques nœuds capteurs et le puits dépasse leur portée maximale. Pour envoyer leurs données au puits, ils doivent le faire par l'intermédiaire d'autres nœuds. Ce type de réseau a une large gamme d'application mais est difficile à mettre en œuvre [27].

3. Selon le modèle de mobilité dans le réseau

Cette classification consiste en une combinaison entre la mobilité des nœuds capteurs et celle du puits. Par cette combinaison, nous pouvons distinguer deux grandes catégories de réseaux : *réseaux statiques* et *réseaux dynamiques ou mobiles* (static and mobile networks).On peut par exemple avoir un réseau constitué d'un ensemble de nœuds capteurs mobiles et d'un puits fixe (réseau à puits statique et à nœuds capteurs mobiles). Le but de tels

réseaux est la plupart du temps l'exploration de zones inaccessibles ou dangereuses. Un autre exemple est un réseau constitué de capteurs fixes servant à la surveillance d'occurrence d'événements sur une zone géographique et d'un puits fixe (réseau statique).

4. Selon les capacités des nœuds du réseau

Dans cette classe, on distingue les réseaux homogènes des hétérogènes. [27][28].

Dans un réseau de capteurs homogène, tous les nœuds du réseau (nœuds capteurs, le(s) puits, les passerelles) ont les mêmes capacités du point de vue énergie, calcul et stockage. Alors que, dans un réseau de capteurs hétérogène il y a quelques nœuds sophistiqués qui ont plus de capacité de traitement et de communication que les nœuds normaux. Cela améliore l'efficacité énergétique et prolonge la vie de réseau .L'avantage d'un tel réseau est que ces nœuds sophistiqués peuvent être utilisés pour exécuter les tâches plus complexes comme les coordinateurs, les chefs du cluster[1] (cluster head), etc. Son inconvénient est qu'il est difficile de mettre en place un tel réseau du fait qu'au moins chaque type de nœuds du réseau aura un code (programme) propre à lui. Ce qui augmente le coût de développement.

I.6 Comparaison entre les RCSFs et les réseaux Ad hoc classiques.

Les RCFS sont souvent comparés aux réseaux ad hoc[2] (ou MANET, pour Mobile Ad hoc NETwork [3]) traditionnels car ces deux types de réseaux partagent beaucoup de points communs [29] [20]:

- ✓ Tous les deux sont des réseaux sans fil, ce qui fait que la portée de communications est limitée par la capacité de rayonnement des antennes utilisées et les puissances mises en jeu. Ainsi, les nœuds dans ces types de réseaux sont souvent dans des configurations multi-sauts,
- ✓ Tous les deux sont des réseaux ad hoc, c'est `a dire, ils fonctionnent sans avoir besoin d'une infrastructure pour la gestion des échanges. De ce fait, ils ont besoin d'être auto-configurables,
- ✓ Ces réseaux travaillent sur une bande de fréquences non propriétaire, ce qui rend leurs communications vulnérables aux problèmes d'interférences,

[1] La notion du cluster et cluster-head seront vues dans le chapitre sur le routage.
[2]Les réseaux ad hoc sont des réseaux sans fil capables de s'organiser sans infrastructure définie au préalable.

- ✓ Les entités de ces réseaux sont souvent alimentées par des batteries,

Malgré les points sur lesquels ces deux type de réseaux convergent, ils différent sur plusieurs aspects. Parmi les points sur lesquels ils divergent, on cite [29] [20] **[5] :**

- ✓ Les entités d'un réseau MANET sont souvent utilisées directement par des êtres humains, comme les portables, les PDA, etc. Dans les réseaux de capteurs sans fil les entités interagissent essentiellement avec la nature ou l'environnement ou entre elles,
- ✓ Le nombre de noeuds dans un réseau de capteur peut-être beaucoup plus important que dans les réseaux ad hoc (forte scalabilité),
- ✓ Les capteurs peuvent être densément déployés (forte densité),
- ✓ Le fait que les nœuds du réseau dans un RCSF sont souvent déployés dans des environnements hostiles (forêts, volcans, etc.) les rend vulnérables et risque de tomber en panne beaucoup plus souvent que les nœuds dans un réseau MANET (tolérance aux panes),
- ✓ La topologie des capteurs change fréquemment du fait des pannes des noeuds ou de leur mobilité (flexibilité),
- ✓ Les échanges de données dans les applications d'un réseau RCSF sont souvent du type collecte de données. Les nœuds doivent envoyer vers un puits des informations sur des phénomènes observés (*modèle de communication many-to-one*) alors que les applications des réseaux MANET sont plus orientés calcul distribué et donc le trafic circule entre tous les nœuds du réseau et dans tous les sens *(modèle de communication any-to-any*),
- ✓ Les réseaux de capteurs utilisent principalement les communications broadcast alors que la plupart des réseaux ad hoc sont basés sur les communications point à point,
- ✓ Les noeuds capteurs sont caractérisés par des ressources plus limitées (ressource d'énergie, puissance de calcul et mémoire). Les nœuds dans un MANET sont plus puissants et possèdent des capacités beaucoup plus importantes,
- ✓ Les RCFS sont qualifiés de réseaux `a basse consommation et `a bas débit *(Low-Power Low-Rate Wireless Personal Area Network, LP-LR-WPAN),* ceci n'est pas le cas des MANET, même si les nœuds d'un MANET sont souvent alimentés sur batteries, ils sont facilement rechargeables,

- ✓ Les noeuds capteurs ne possèdent aucune identification (ID) globale tel que les adresses IP dans les réseaux ad hoc.

I.7 Facteurs et contraintes de conception d'un RCSF.

La conception et la mise en place des RCSFs sont influencées par plusieurs contraintes qui peuvent être des contraintes conceptuelles ou matérielles. Ces facteurs importants servent comme directives pour le développent des algorithmes et protocoles utilisés dans les réseaux de capteurs ; ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine.

I.7.1 Contraintes conceptuelles

La conception des RCSF, leurs protocoles et algorithmes sont guidés par plusieurs facteurs:

1. La tolérance aux panes

La défaillance ou le blocage de certains nœuds dans un réseau de capteurs peut être engendrés par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique, ou les interférences liées à l'environnement. Ces problèmes ne devraient pas affecter le reste du réseau. C'est le principe de la tolérance aux panes.

2. L'extensibilité (passage à l'échelle)

L'une des caractéristiques des RCSF est qu'ils peuvent contenir des centaines voire des milliers de nœuds capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs. Les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre élevé de capteurs. Ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.

3. Le Coût de production

Le coût de production d'un seul micro-capteur est très important pour l'évaluation du coût global du réseau, si ce dernier est supérieur à celui nécessaire pour le déploiement des capteurs classiques, l'utilisation de cette nouvelle technologie ne serait pas rentable. Par

conséquent, réduire le coût de production jusqu'à moins de 1$ par nœud est un objectif important pour la faisabilité de la solution des réseaux de capteurs sans fil [20].

4. Environnement

Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie,...

5. Média de transmission

Les nœuds communicants sont reliés sans fil. Ce lien peut être réalisé par radio, signal infrarouge ou un média optique.

6. Consommation énergétique

Les nœuds capteurs, étant des dispositifs microélectroniques, peuvent être équipés seulement d'une source d'énergie limitée (<0.5 Ah, 1.2 V). Dans certains scénarios d'application, il est impossible de réapprovisionner de l'énergie. La durée de vie d'un capteur est donc dépendante de la durée de vie de sa batterie. D'autre part, la retransmission des données, la réorganisation du réseau ainsi que le changement de sa topologie rendent la gestion et la conservation d'énergie d'une haute importance. Cette énergie est consommée par les différentes unités du capteur afin de réaliser les tâches de captage, traitement de données et communication. Cette dernière est l'opération qui consomme le plus d'énergie.

7. Agrégation de données

Dans les RCSF, les données produites par les nœuds capteurs voisins sont très corrélées spatialement et temporellement (voir chapitre IV). Ceci peut engendrer la réception par la station de base d'informations redondantes. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau et ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données, appelée aussi fusion de données. Cette technique sera étudiée en détails dans le quatrième chapitre de ce travail.

I.7.2. Contraintes matérielles

Parmi les contraintes matérielles liées aux RCSFs, on peut citer :

1. Dimension

La taille réduite des capteurs peut présenter de nombreux avantage, elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée, par la petite taille de ces derniers.

2. Puissance de calcul

Les processeurs des réseaux de capteurs sont différents de ceux d'une machine classique. Car ils utilisent souvent des microcontrôleurs de faibles fréquences.

I.8 Domaines d'application des RCSF

La taille de plus en plus réduite des micro-capteurs, leur coût de plus en plus faible, la large gamme des types de capteurs disponibles (thermique, optique, de vibrations, etc.) ainsi que le support de communication sans fil utilisé, permettent aux réseaux de capteurs d'envahir rapidement plusieurs domaines d'applications [4]. En effet, ces réseaux se révèlent très utiles dans de nombreuses applications lorsqu'il s'agit de collecter et de traiter des informations provenant de l'environnement. Parmi ces domaines, nous citons les domaines : militaire, environnemental, écologique, domestique, de santé, de sécurité, etc. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous.

1. Applications militaires

Comme dans le cas de plusieurs technologies, certaines des premières utilisations proposées de réseaux de capteurs sans fil étaient destinées aux applications militaires[6]. Une grande partie de la croissance rapide dans la recherche et le développement des réseaux de capteurs sans fil a été apportée par des programmes financés par l'Agence américaine pour les Projets de Recherche Avancée de Défense (DARPA pour Defense Advanced Research Projects Agency), notamment grâce à un programme connu sous le nom de "SensIT" (Sensor Information Technology) [7] de 1999 à 2002.

Dans ce domaine les RCSFs peuvent être utilisés pour remplacer des gardes et des sentinelles autour des périmètres défensifs. De cette façon, ils peuvent servir la même fonction que les mines antipersonnel. En plus de telles applications défensives, le déploiement

des RCSFs dans une zone stratégique ou difficile d'accès, permettrait par exemple d'y surveiller tous les mouvements, le nombre et l'identité des soldats (amis ou ennemis), ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou nucléaires).

2. Application médicales (santé)

Le champ de contrôle de santé (Health monitoring) représente un grand marché pour les réseaux de capteurs sans fil qui a tendance à croître très rapidement. Ces réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain (surveillance de la glycémie, détection de cancers, ..) grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, le rythme cardiaque, ... à l'aide des capteurs ayant chacun une tâche bien particulière.

3. Applications domestiques

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc.[9]. Ces capteurs embarqués peuvent interagir entre eux et avec un réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. En plaçant, sur le plafond ou dans le mur, des capteurs, on peut économiser l'énergie en gérant l'éclairage ou le chauffage en fonction de la localisation des personnes et seulement si c'est nécessaire.

4. Applications environnementales

Les micro-capteurs dispersés à partir d'un avion dans une zone difficile d'accès peuvent permettre de détecter des incendies, surveiller des catastrophes naturelles (inondations, séismes, éruptions volcaniques), surveiller des phénomènes météorologiques, de détecter de la pollution (qualité des eaux, taux d'ensoleillement, taux de radioactivité, fuite du pétrole, taux de CO2…),

5. Applications écologiques

L'intégration de plusieurs micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenchés qu'aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué peut aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait probablement de réduire la demande mondiale en énergie.

6. Agriculture de précision

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

7. Application industrielles

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

La figure I.3 ci-dessous montre quelques domaines d'application cités précédemment.

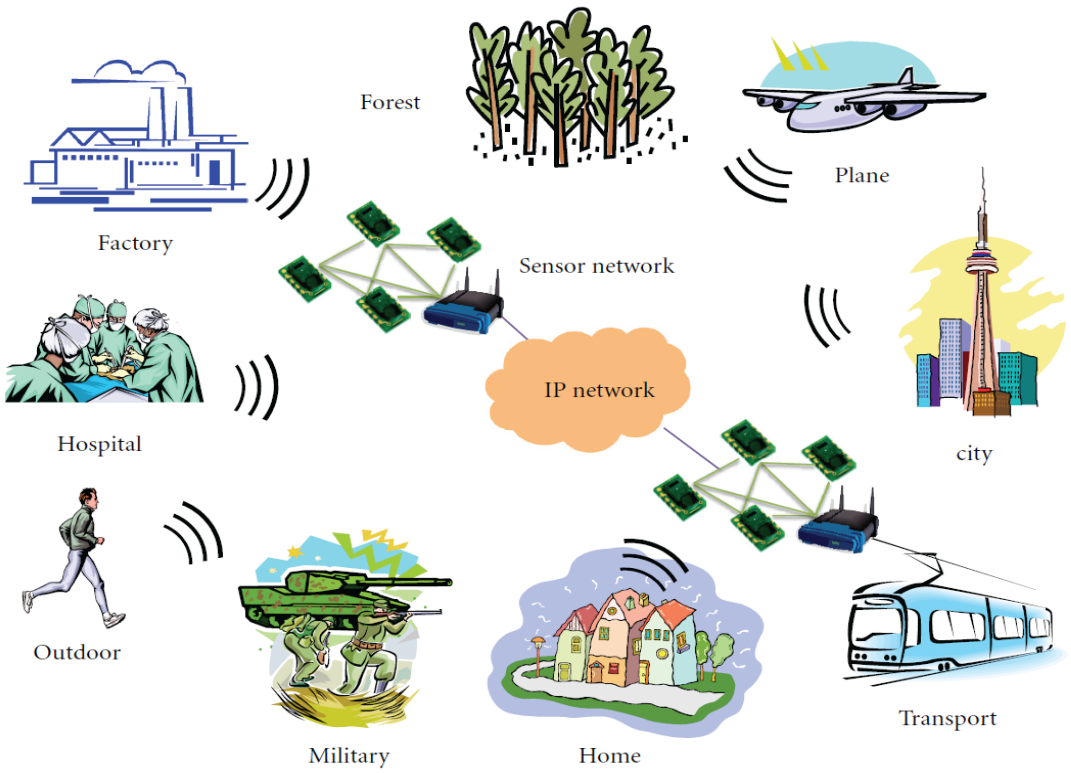

Figure I. 3. Quelques domaines d'application pour les RCSFs [30]

I.9 La communication dans les RCSF

I.9.1 Pile protocolaire

La volonté de ne pas favoriser un industriel, la prise en compte de l'hétérogénéité des équipements, et la possibilité d'une évolution et d'une adaptation facile ont nécessité l'adoption de règles communes de communication et de coopération entre les équipements.ces règles forment un protocole de communication. Traditionnellement, la pile protocolaire utilisée dans le monde des réseaux est organisée suivant un modèle en couches superposées l'une sur l'autre. Chaque couche assure de manière indépendante une partie des fonctionnalités nécessaires à la communication entre les entités du réseau. Elle utilise les services des couches inferieures et en fournit à celle de niveau supérieur.

Différents des réseaux ad hoc traditionnels, les réseaux de capteurs exigent des nouvelles limitations pour la conception des protocoles de communication. Ces protocoles doivent prendre en compte des contraintes de conception propres aux RCSF (voir section I.7).Le modèle OSI(Open Systems Interconnexion) de l'ISO(International Standardization Organization) ne suffit donc plus.

La pile protocolaire [20] utilisée par la station de base ainsi que tous les autres capteurs du réseau est illustrée par la figure I.4. Ce modèle comprend 5 couches (une couche application, une couche transport, une couche réseau, une couche liaison de données, une couche physique) qui ont les mêmes fonctions que celles du modèle OSI, ainsi que 3 niveaux (plans) qui sont : un plan de gestion d'energie, un plan de gestion de mobilité et un plan de gestion des tâches.

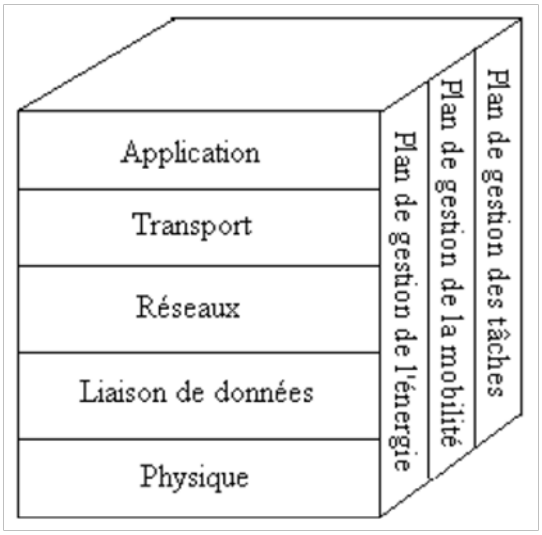

Figure I. 4. Pile protocolaire pour les RCSFs

1. La couche physique

La couche physique est responsable de :

- ✓ La sélection des fréquences porteuses.
- ✓ La détection du signal.
- ✓ La modulation.

2. Couche liaison de données

En générale, cette couche est responsable du multiplexage du flux de données, de la détection et le verrouillage des frames de données, du contrôle d'accès aux média (MAC: Media Access Control), et du control des erreurs. Elle assure une connexion fiable (point-à-point ou point-à-multipoints) selon la topologie du réseau de capteurs.

3. Couche réseau

Le but principal de la couche réseau est de trouver une route et une transmission fiable des données, captées, des nœuds capteurs vers le puits en optimisant l'utilisation de l'énergie des capteurs. Les caractéristiques spécifiques aux RCSFs (voir Section I.6) exigent que leurs protocoles de routage diffèrent de ceux des réseaux ad hoc traditionnels. Le routage dans les RCSFs sera étudié en détails dans le deuxième chapitre de ce mémoire.

4. La couche transport

Cette couche est chargée du transport des données, de leur découpage en paquets, du contrôle de flux, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission.

5. La couche application

La couche application constitue l'ensemble des applications implémentées sur un réseau de capteurs. Ces applications doivent fournir des mécanismes permettant à l'utilisateur d'interagir avec le réseau de capteurs à travers différentes interfaces, et éventuellement, par l'intermédiaire d'un réseau étendu (par exemple ; Internet). Il s'agit donc du niveau le plus proche des utilisateurs, géré directement par les logiciels.

Quant aux niveaux (plans) intégrés dans la pile protocolaire, ils ont les fonctions suivantes :

6. Le plan de gestion d'énergie

Un nœud de capteur sans fil, nécessite seulement une source d'énergie limitée (< 0.5Ah, 1.2 V). La vie du nœud montre, une dépendance forte à l'égard de la vie de la batterie. Le plan de gestion d'énergie doit gérer la manière dont les nœuds utilisent leurs énergies. Par exemple le nœud doit se mettre en sommeil après la réception d'un message à partir d'un voisin afin d'éviter la duplication des messages déjà reçus.

7. Le plan de gestion de mobilité :

Puisque les nœuds peuvent être mobiles, un système de gestion de mobilité doit exister. Un tel système doit être capable d'enregistrer les mouvements du nœud afin de l'aider à se localiser.

8. Plan de gestion de tâche :

Lors d'une opération de captage dans une région donnée, les nœuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme, cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans la quelle il a été déployé. Pour cela, le niveau de gestion des tâches assure l'équilibrage et la distribution des tâches sur les différents nœuds du réseau, afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau.

I.9.2 Les standards de communication pour les RCSFs

Il y a une multitude de normes sans fil comme le Wi-Fi(le standard IEEE 802.11)[34] et le WiMax (le standard IEEE 802.16)[35] qui s'adressent au transport des données à haut débit .Certains dispositifs comme les capteurs n'ont pas besoin d'une largeur de bande très élevée, mais plutôt d'un temps de latence faible ainsi qu'une consommation d'énergie très basse, pour un longue durée de vie sur batterie et un grands nombre de dispositifs. D'où la nécessité de concevoir d'autres normes sans fil capables de répondre à ces exigences. Parmi les standards les plus aptes à être exploités dans les réseaux de capteurs sans fil se retrouvent les standards *Bluetooth* et *ZigBee* présentés ci-après.

I.9.2 .1 Bluetooth

Le Bluetooth [10] est un standard de communication sans fil utilisant une technologie radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Il représente l'équivalent de l'USB mais sans fil et sert donc à relier des périphériques avec un ordinateur ou d'autres périphériques (imprimantes, scanners, claviers, souris, téléphones portables, PDA, …). Le succès de la technologie Bluetooth l'a amené à être utilisée dans le cadre des LAN (Local Area Network) sans fil. L'avantage du Bluetooth par rapport à Wi-Fi, qui offre apparemment de meilleures performances, se trouve au niveau de la consommation : en effet, la connexion sans fil est surtout utilisée pour les appareils mobiles, et donc n'étant pas directement reliés au secteur. Par conséquent, la technologie Bluetooth a une plus faible consommation que le Wi-Fi. Le second avantage est le prix attractif de cette technologie grâce à son faible coût de réalisation des puces. Cependant, cette faible consommation d'énergie ne répond pas encore à des exigences des capteurs. En plus de cette consommation d'énergie, la technologie Bluetooth soufre d'un problème de passage à l'échelle.

I.9.2 .2 ZigBee

Beaucoup moins connue que Bluetooth [31], ZigBee [11] est le nom d'une suite de protocoles de hauts niveaux basée sur le standard IEEE 802.15.4 [12]pour les réseaux personnels sans fil(Wireless Personal Area Network :WPAN). La technologie ZigBee a pour but la communication à courte distance telle que le propose déjà la technologie Bluetooth, tout en étant moins chère et plus simple. Les nœuds sont conçus pour fonctionner plusieurs mois (jusqu'à deux ans) en autonomie complète grâce à une simple pile alcaline de 1,5V. Sa très faible consommation électrique et ses coûts de production très bas en font une candidate idéale pour les RCSFs [31]. Le petit débit qu'il offre (250Kbps théorique contre 1Mbps pour Bluetooth), n'est pas vraiment un handicap pour un réseau de capteurs puisque la taille des paquets échangés n'est pas vraiment importante.

1. Le projet Zigbee

En 2000, un sous-groupe du consortium HomeRF, se baptisant ZigBee coopère avec le groupe de travail 15 de l'IEEE 802 pour adresser le besoin d'un protocole sans fil économique et de très faible consommation, pour les environnements résidentiels et aussi industriels. Au mois de décembre de la même année, l'IEEE érige un nouveau groupe de travail, sur la décision de son comité de création de nouveaux standards, pour initier la définition d'un protocole sans-fil bas débit (Low Rate WPAN : LR-WPAN), appelé IEEE802.15.4 [15] [17]. L'objectif de ce groupe sera de développer la couche physique (PHY) et la couche d'accès au médium (MAC) d'un standard se voulant simple, économique et peu gourmant en énergie. Le standard IEEE802.15.4 sera ainsi définitivement ratifié par l'IEEE en mai 2003. Dans le même temps, l'alliance ZigBee (la ZigBee Alliance [13]), une association composée de grands industriels dont Motorola, Philips , Samsung, Siemens,etc, renforcée de jour en jour par de nouveaux membres, s'est focalisée sur le développement des couches réseaux et applicatives qui forment avec la norme 802.15.4 le protocole ZigBee.

2. La pile protocolaire Zigbee

La pile protocolaire proposée par IEEE 802.15.4 et la ZigBee Alliance suit les recommandations de l'ISO en terme de séparation des rôles attribués aux différentes couches. Comme cela a été dit précédemment, cette pile reprend les couches 1 et 2 normalisées dans la norme IEEE 802.15.4 et ajoute ses propres couches supérieures. L'ensemble de ces couches est montré par la figure I.5 ci-dessous.

Figure I. 5. Pile protocolaire ZigBee

2.1 Couche physique IEEE 802.15.4

a. Description

La couche physique IEEE 802.15.4 décrit les caractéristiques de l'interface radio (fréquences, largeur de bande, modulation, débit binaire, etc.).Conformément à IEEE 802.15.4, ZigBee peut travailler sur trois bandes de fréquences différentes : 868 MHz pour la région Europe, 915 MHz pour l'Amérique du nord, et 2,4 GHz pour une couverture mondiale. La norme prévoit deux couches physiques différentes (PHY), une pour le 868/915MHz (PHY868/915) et une seconde pour le 2,4 GHz (PHY2450).Avec les trois bandes de fréquences, la couche physique offre 27 canaux de transmission différents (numérotés de 0 à 27) dont 16 dans la bande 2,4 GHz séparés de 5MHz, 10 dans la bande de 915 MHz séparés de 2 MHz et 1 canal dans la bande 868.Le tableau I.1 ci –dessous résume les caractéristiques principales des trois bandes de fréquence

Tableau I. 1. Caractéristiques principales de la couche physique

PHY	Bande de fréquence	Paramètres de codage			Etalement de spectre	
		Débit en bits/s	Débit en kbauds	Type modulation	Débit au niveau chip (*) (Mbits/s)	Modulation
868/915 MHz	868.0-868.6 MHz	20	20	BPSK	0,3	BPSK
	902.0-928.0 MHz	40	40	BPSK	0,6	BPSK
2,4 GHz	2,4-2,4835 GHz	250	62,5	16-orthogonal	2,0	O-QPSK

(*) Un « chip » est un vecteur, composé de 0 et de 1, résultant de l'étalement d'un signal.

La couche PHY868/915 est relativement simple et basique : les symboles sont binaires, grâce à l'emploi d'une modulation BPSK (Binary Phase Shift Keying) et un débit peu élevé (20 Kbits/s pour le 868 MHz, 40 Kbits/s pour le 915 MHz). En revanche, la couche PHY2450 propose une modulation plus complexe, O-QPSK (Offset Quadrature Phase ShiftKeying), qui permet un débit plus intéressant.

b. Portée, puissance d'émission et sensibilité du récepteur

IEEE 802.15.4 prévoit une portée classique de quelques dizaines à quelques centaines de mètres suivant l'environnement considéré[3] . La puissance maximale émise par un module 802.15.4 ou ZigBee n'est pas définie par la norme ; celle-ci est laissée d'une part à l'appréciation de l'autorité de régulation de la zone où est effectuée la transmission, et d'autre part au constructeur pour des questions évidentes d'autonomie énergétique du système dans lequel il est implanté. Néanmoins, la puissance typique recommandée est de 1 Mw.

Notons que comme pour toutes les technologies de réseau sans fil, la portée effective d'un transceiver 802.15.4 est très liée à sa puissance d'émission. Certains modules sont dotés d'un étage amplificateur en sortie HF et/ou d'un amplificateur à faible bruit en entrée, ce qui permet d'étendre considérablement la portée radio. A titre d'exemple, les modules XBeePRO fabriqués par la société MaxStream [14] sont vendus pour une portée extérieure à champs libre de 1,6 km.

c. Fonctionnalités

Parmi les fonctionnalités de contrôle de cette couche, nous pouvons disposer de celles qui permettent :

- ✓ d'activer et désactiver le module radio ;
- ✓ La détection de l'énergie et la sélection de la fréquence du canal ;
- ✓ de tester l'occupation du canal ou CCA (Clear Channel Assessment) ;
- ✓ l'indication de la qualité du lien ou LQI (Link Quality Indication) ;
- ✓ la transmission et la réception des données.

[3] L'environnement par rapport à la présence ou sans présence des obstacles.

2 .2 La sous couche MAC IEEE 802.15.4

a. Description

C'est la charnière du standard ZigBee, située entre la gestion bas niveau (fréquences de communication, des débits, du type de modulation…) et la gestion haut niveau des réseaux (association, appairage…). Cette couche, appelée MAC (Medium Access Controller), pose les principes de bases qui permettent aux nœuds de communiquer au sein d'un réseau ZigBee. Lorsqu'un nœud ZigBee possède une information à transmettre, il utilise le mécanisme de détection de porteuse sur le canal (CSMA-CA : Carrier Sense Multiple Access with Collision Avoidance), c'est-à-dire qu'il écoute le canal pour repérer si celui-ci est libre , auquel cas il transmet sa trame. Dans le cas contraire, il attend pendant un laps de temps court, mais de durée aléatoire, avant de faire une autre tentative.

b. Format des trames ZigBee

Comme dans n'importe quel réseau, les données sont transmises en trames (paquets). La structure des trames ZigBee à été conçue afin de limiter au maximum la complexité et pour les rendre suffisamment robustes pour une transmission en environnement fortement bruité. Chaque couche protocole ajoute à la trame sa structure en incluant des en-têtes et des pieds de trames spécifiques.

La couche MAC IEEE 802.15.4 définit 4 types de trames :

- ✓ Trame de données (data), utilisée pour tous les transferts de données,
- ✓ Trame d'acquittement (ack), utilisée pour confirmer qu'une trame de données a été reçue avec succès,
- ✓ Trame de balise (beacon), utilisée par le coordinateur pour transmettre les beacons,
- ✓ Trame de commande MAC service (MAC command) , utilisée pour gérer tous les transferts de contrôle MAC

Les trames MAC (MAC frames ou MAC packet data unit:PMDU) sont encapsulées dans un paquet appelé Packet *Service Data Unit*(PSDU). Chaque trame MAC est composée de :

- ✓MAC header (en-tête), qui contient le champ contrôle de trame (frame control :FC),le numéro de sequence(Sequence Number),information d'adressage et information de sécurité. Ces deux derniers ne sont pas inclus dans toutes les trames, par exemple, la trame d'acquittement ne les inclut pas.

- ✓MAC payload (charge utile) contient les données utiles de la trame. La trame d'acquittement ne l'inclut pas.

- ✓ MAC footer (MFR) (les pieds), qui contient la trame FCS pour la détection des bits d'erreurs.

La figure suivante illustre le format général de la trame MAC.

<table>
<tr><td>Octets: 2</td><td>1</td><td>0/2</td><td>0/2/8</td><td>0/2</td><td>0/2/8</td><td>0/5/6/10/ 14</td><td>variable</td><td>2</td></tr>
<tr><td rowspan="2">Frame Control</td><td rowspan="2">Sequence Number</td><td>Destination PAN Identifier</td><td>Destination Address</td><td>Source PAN Identifier</td><td>Source Address</td><td rowspan="2">Auxiliary Security Header</td><td rowspan="2">Frame Payload</td><td rowspan="2">FCS</td></tr>
<tr><td colspan="4">Addressing fields</td></tr>
<tr><td colspan="7">MHR</td><td>MAC Payload</td><td>MFR</td></tr>
</table>

Figure I. 6. Le format général de la trame MAC

2.3 La couche réseau

a. Les éléments de la topologie du réseau

Le protocole ZigBee possède trois types de périphériques : le coordinateur (ZC :ZigBee Coordinator),les routeurs (ZR :ZigBee Router) et les dispositifs terminaux(ZED :ZigBee End-Device).

i) Le coordinateur

Ce module va gérer les fonctions de haut niveau du réseau comme la sécurité. Un seul coordinateur doit être présent pour le même réseau (même identifiant de réseau PAN ou PANID).Il est obligatoire pour la mise en place du réseau.
Le coordinateur doit d'être présent tout au long de la vie du réseau. Il est donc fortement conseillé d'alimenter en permanence ce module (pas de pile). Il est aussi possible de le mettre sur onduleur (batterie de grande capacité avec recharge automatique).

ii) Les routeurs

Ces modules disposent de toutes les fonctions d'un module End-Device (dispositif terminal du réseau) avec en plus des fonctions de haut niveau utiles pour étendre le réseau. On peut en avoir plusieurs disposés généralement à des points clef de maillage.

Ils permettent :

- ✓ d'étendre la taille du réseau en permettant aux autres modules de s'enregistrer auprès d'eux et non exclusivement auprès du coordinateur. Cela évite de saturer le coordinateur en nombre de modules inscrits. La taille maximum d'un réseau peut ainsi atteindre 65536 modules.
- ✓ d'étendre la portée du réseau. Chaque module routeur répète les signaux reçu aux autres modules qui lui sont enregistrés. Le signal se répercute ainsi de module en module pour atteindre le End-device concerné.

L'utilisation d'un routeur présente quelques inconvénients : ses fonctions augmentent fortement la consommation au niveau des modules terminaux. De plus, en cas d'indisponibilité d'un routeur tous les modules enregistrés auprès de lui deviennent invisibles. Il est donc fortement conseillé de choisir une alimentation conséquente et continue.

iii) Les End-Devices

Ce sont tous les modules terminaux comme les capteurs, actionneurs... Ils ne sont actifs que sur changement de leurs états ou sur réponse à une trame, leurs consommation est donc très faible et ils peuvent tout à fait être alimenté par des piles ou des batteries.

La couche réseau de ZigBee supporte les topologies en étoile(star), arbres(tree) et maillées(mesh) visibles sur la figure suivante.

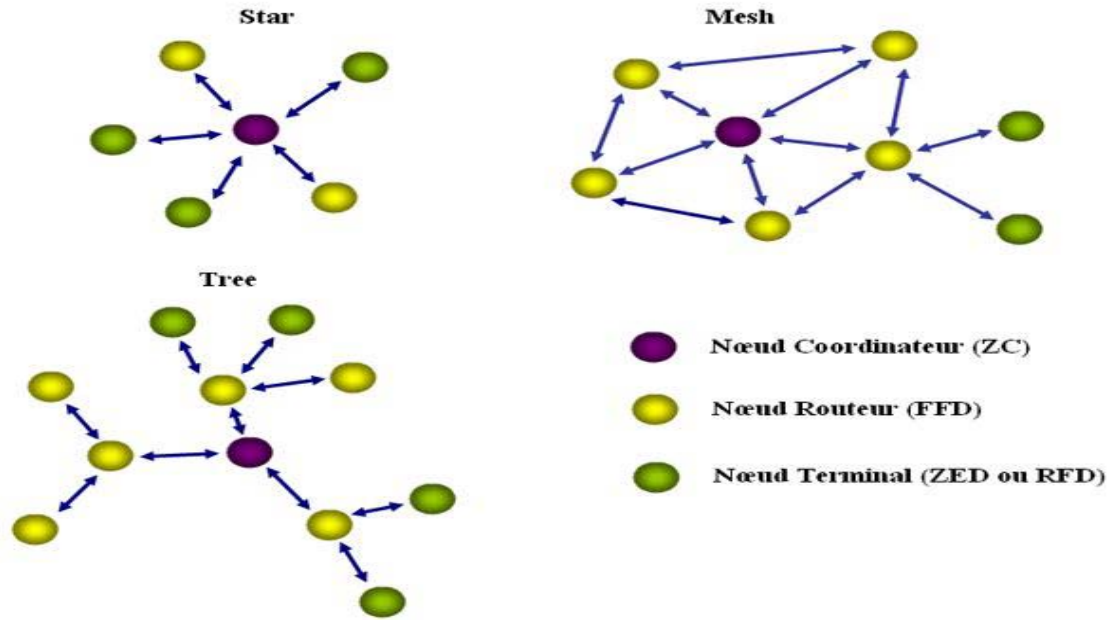

Figure I. 7. Les topologies réseaux sous Zigbee .

Dans la topologie en étoile, le réseau est contrôlé par le coordinateur(ZC) qui, en plus des fonctions de création du réseau et de sa maintenance, assure la communication de tous les ZED entre eux. En effet, Tous les autres nœuds ne communiquent qu'avec le coordinateur ZigBee.

Dans les deux autres topologies (en arbre et maillée), le coordinateur ZigBee est responsable de la création du réseau et de la définition de certains paramètres réseau. Le réseau peut, ensuite, être étendu en utilisant des routeurs ZigBee. Les réseaux en arbre emploient généralement la structure de supertrame pour la communication (mode *Beacom*) et le routage se fait d'une manière hiérarchique(voir chapitre 2). Le réseau maillé, en revanche, permet une communication paire-à-paire entre les nœuds et utilise donc le mode sans *beacon*.

b. Adressage

ZigBee utilise 2 modes d'adressage :

- ✓ Une adresse 64 bits IEEE qui peut être comparée à une adresse IP sur l'Internet, adresse source + adresse destination (topologie pairs à pairs)
- ✓ Une adresse courte sur 16 bits, Id du réseau + Id du nœud (topologie en étoile), utilisée seulement une fois que le réseau est en place ce qui donne un total de 2^16 = 65536

nœuds disponibles dans un réseau, ce qui semble bien assez. En cas de besoin supplémentaire, il est tout a fait envisageable de créer un système de passerelle sur un nœud.

L'attribution d'adresses dans ZigBee se fait de deux manières selon une certaine configuration.

I.10 Conclusion

Dans ce chapitre nous avons vu les concepts généraux liés aux réseaux de capteurs sans fil.

Parmi ces concepts nous avons identifié les contraintes de conception des réseaux de capteurs sans fil. Parmi ces contraintes nous avons vu les contraintes matérielles et conceptuelles.

Un point sur les protocoles de communication sans fil dont Zigbee a été aussi abordé.

La compréhension de ces concepts est nécessaire pour les concepteurs des protocoles et applications destinés aux réseaux de capteurs sans fil.

Chapitre II

II.1 INTRODUCTION

Un réseau de capteurs sans fil consiste en un ensemble de nœuds capteurs déployés à grande échelle dans un champs 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 serrait, 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 leur 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 protocole de routage n'est pas une tâche facile.

Plusieurs travaux de recherche dans le domaine des RCSFs ont été effectués récemment et ont abouti à une multitude protocoles de routage destinés à ces réseaux.

Dans ce chapitre, nous faisons une étude bibliographique sur les protocoles de routage proposés pour les RCSFs. Vu leur multiplicité, il nous semble important de commencer par leur classification qui les groupe suivant un certain nombre de critères.

II.2 Classification des protocoles de routage pour les RCSFs

Récemment, les protocoles de routage pour les RCSFs ont été largement étudiés. Les protocoles proposés présentent les points communs et donc peuvent être classifiés suivant un certain nombre de critères .La figure II.1 ci-dessous résume une classification qui se base sur quatre critères : la topologie (structure) du réseau, mode d'établissement établissement des chemins, les paradigmes de communication et selon le mode de fonctionnement du protocole .

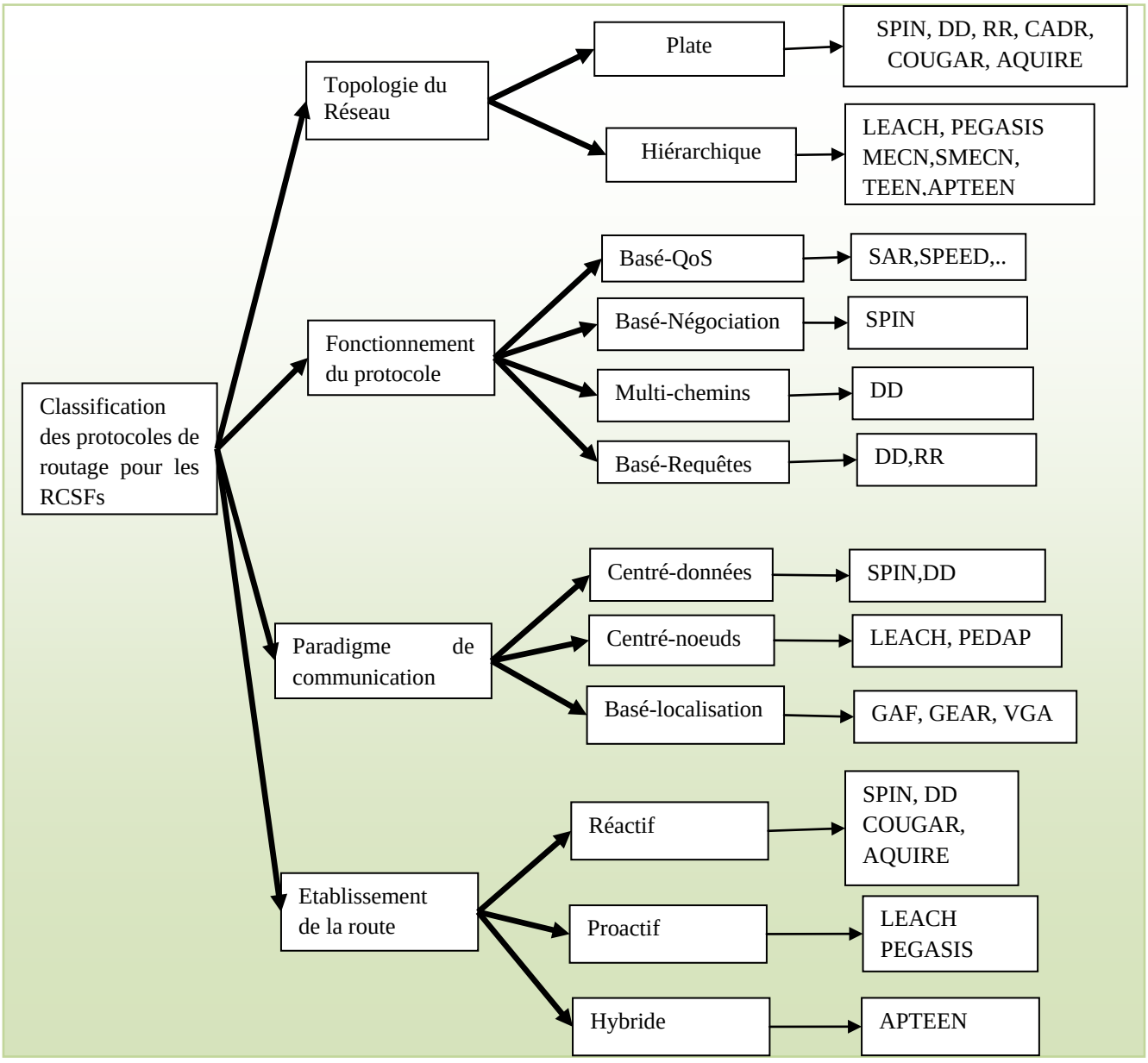

Figure II. 1 . Classification des protocoles de routage pour les Réseaux de capteurs sans fil

II.2.1 Selon la topologie (structure) du réseau

La topologie détermine l'organisation des capteurs dans le réseau. Globalement, il existe deux topologies dans les RCSFs **[8]** :la topologie plate et la topologie hiérarchique.

1. Topologie plate

Dans une topologie plate, tous les nœuds capteurs possèdent le même rôle et collaborent entre eux pour accomplir la tâche de routage. Les réseaux plats sont caractérisés par : la simplicité des protocoles de routage, un coût de maintien réduit, une grande tolérance aux pannes ainsi qu'une habilité à construire de nouveaux chemins suite aux changements de topologie. Cependant, Les nœuds proches du puits participent plus que les autres aux tâches de routage. De plus, ces réseaux présentent une faible scalabilité dû au fonctionnement identique des nœuds et d'une manière distribuée nécessitant ainsi un grand nombre de messages de contrôle. La figure suivante illustre l'organisation des capteurs dans une topologie plate.

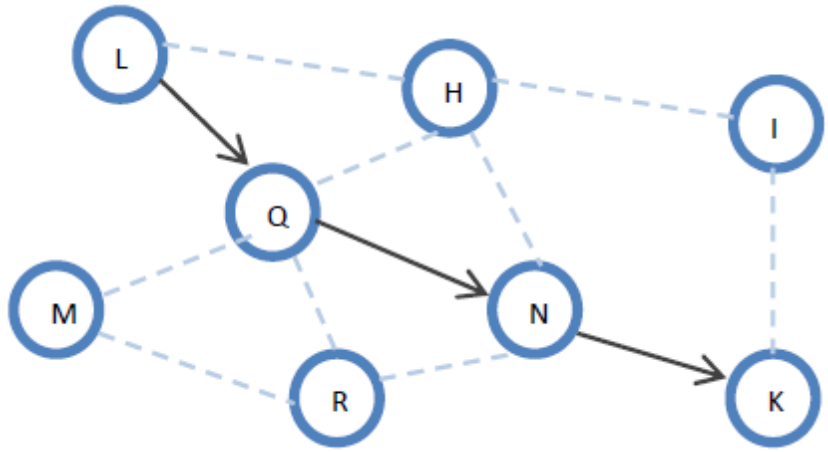

Figure II. 2 Topologie plate

2. Topologie hiérarchique

Dans une topologie hiérarchique, les nœuds ont des différents rôles. En effet, certains nœuds sont sélectionnés pour exécuter des fonctions particulières. Une des méthodes les plus utilisées dans cette topologie est le clustering. Il consiste en un partitionnement du réseau en groupes appelés clusters. Un cluster est constitué d'un chef *(clusterhead)*

et de ses membres. Suivant l'application, les membres peuvent être des voisins directs du chef ou pas.

Cette topologie présente beaucoup d'avantages tels que l'agrégation des données collectées ainsi qu'une grande scalabilité. Son inconvénient majeur est la surcharge des clusterheads qui induit un déséquilibre de la consommation d'énergie dans le réseau. Pour remédier à ce problème, clusterheads peuvent être des capteurs spécifiques avec plus de ressources énergétiques et plus de capacités de traitement ou bien ils peuvent être élus dynamiquement et ainsi garantir un équilibre de la consommation d'énergie et augmenter la tolérance aux pannes. Un exemple de cette topologie est donné dans la figure II.3 ci-dessous. Pour que les paquets générés par le nœud F atteignent le nœud L, ils doivent passer par les passerelles P, S et R.

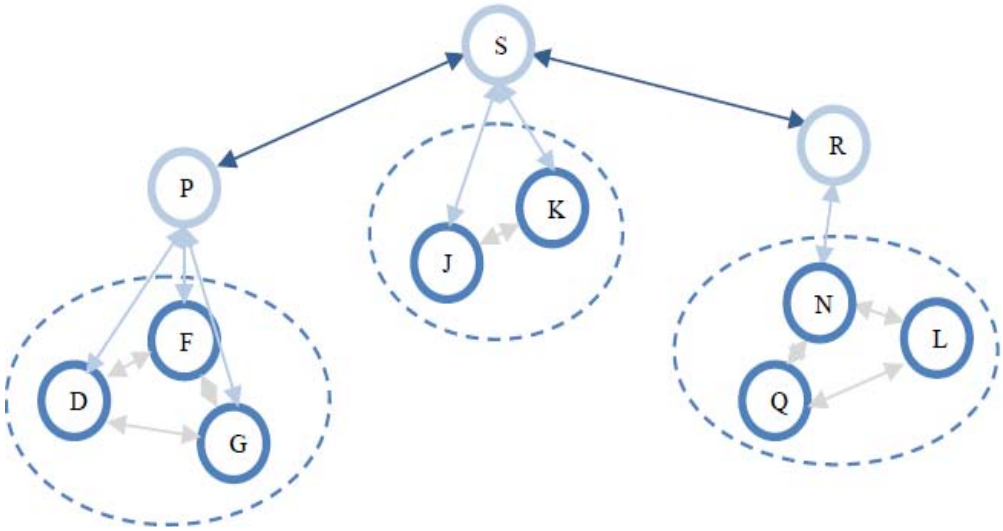

Figure II. 3. Routage hiérarchique

II.2.3 Selon les paradigmes de communication

Le paradigme de communication détermine la manière dont les nœuds sont interrogés. Dans les RCSFs, il existe trois paradigmes de communication [16**]** :centré-nœuds,centré-données et basé sur la localisation.

3. Centré-nœuds (Node-centric)

Ce paradigme est celui employé dans les réseaux conventionnels, où il est nécessaire de connaître et d'identifier les nœuds communicants (comme l'adresse IP). Les réseaux ad hoc

utilisent ce genre de paradigme, qui s'intègre bien avec l'utilisation de ce type d'environnement. Cependant, pour les réseaux de capteurs, un routage basé sur une identification individuelle des nœuds ne reflète pas l'usage réel du réseau. Pour cela, un autre paradigme a été introduit : *data centric*. Néanmoins, le paradigme *node centric* n'est pas à écarter totalement, car certaines applications nécessitent une interrogation individuelle des capteurs.

4. Centré-données (Data-centric)

Ce paradigme suppose qu'il est difficile d'avoir des identifiants comme les adresses MAC ou IP pour pouvoir communiquer entre les nœuds capteurs [29].Ainsi, le routage ne se fait pas en fonction d'une adresse de destination, mais suivant les données disponibles au niveau des capteurs. Ces données seront propagées de proche en proche pour arriver au nœud puits.

5. Basé-localisation (location-based)

Dans cette approche, les décisions de routage sont établies selon la position des nœuds. La distance entre les nœuds voisins peut être estimée sur la base de la puissance du signal arrivé. Un tel type de routage nécessite que les nœuds aient connaissance de leurs positions géographiques. Par conséquent, ce type de mécanismes nécessite un déploiement d'une solution de positionnement, dont le degré de précision requis dépend de l'application ciblée. L'utilisation du GPS reste trop coûteuse pour un RCSF. Néanmoins, d'autres méthodes de localisation et de positionnement des capteurs ont été développées comme par exemple la triangulation [32].

II.3.3 Selon le mode de fonctionnement du protocole

Le mode de fonctionnement définit la manière avec laquelle les données sont propagées dans le réseau. Selon ce critère, les protocoles de routage peuvent être classifiés en quatre catégories : routage basé sur la Qualité de Service "QdS" *(Quality of Service "QoS" based routing)*, routage basé sur les requêtes (query-based routing), routage multi-chemins (Multi-path routing), et routage basé sur la négociation (Negociation based routing) [33].

1. Routage basé sur les multi-chemins

Dans cette catégorie, les protocoles de routage utilisent des chemins multiples plutôt qu'un chemin simple afin d'augmenter la performance du réseau. La fiabilité d'un protocole peut être mesurée par sa capacité à trouver des chemins alternatifs entre la source et la destination en cas de défaillance du chemin primaire. Pour cette raison, certains protocoles construisent plusieurs chemins indépendants, c.-à-d. : ils ne partagent qu'un nombre réduit (voir nul) de nœuds. Malgré leur grande tolérance aux pannes, ces protocoles requièrent plus de ressources énergétiques et plus de messages de contrôle.

2. Routage basé sur les requêtes

Dans ce type de routage, le puits génère des requêtes afin d'interroger les capteurs. Ces requêtes sont exprimées soit par un schéma valeur-attribut ou bien en utilisant un langage spécifique (par exemple SQL : Structured Query Language). Les nœuds qui détiennent les données requises doivent les envoyer au nœud demandeur à travers le chemin inverse de la requête. Les requêtes émises par le puits peuvent aussi être ciblées sur des régions spécifiques du réseau.

3. Routage basé sur la négociation

En détectant le même phénomène, les nœuds capteurs inondent le réseau par les mêmes paquets de données. Ce problème de redondance peut être résolu en employant des protocoles de routage basés sur la négociation. En effet, avant de transmettre, les nœuds capteurs négocient entre eux leurs données en échangeant des paquets de signalisation spéciales, appelés métadonnées. Ces paquets permettent de vérifier si les nœuds voisins disposent déjà de la donnée à transmettre. Cette procédure garantit que seules les informations utiles seront transmises et élimine la redondance des données.

4. Routage basé sur la qualité de service

Dans les protocoles de routage basés sur QoS, le réseau doit équilibrer entre la consommation d'énergie et la qualité de données. En particulier, le réseau doit satisfaire certaines métriques de QoS, par exemple, retard, énergie, largeur de bande passante, etc. Les protocoles de cette approche sont très recommandés pour les applications de surveillance (centrales nucléaires, applications militaires, etc).

II.3.4 Selon le mode l'établissement des chemins

Suivant la manière de création et de maintien des chemins pendant le routage,nous distinguons trois catégories de protocoles de routage : les protocoles proactifs, les protocoles réactifs et les protocoles hybrides[36].

1. Les protocoles proactifs

Utilisent l'échange régulier de messages de contrôle pour maintenir au niveau de chaque nœud des tables de routage (qui associent à chaque destination ou groupe de destinations un voisin direct par lequel les paquets doivent être relayés) vers toute destination atteignable depuis celui-ci. Ces tables sont maintenues même quand les routes ne sont pas utilisées. Cette approche permet de disposer d'une route vers chaque destination immédiatement au moment où un paquet doit être envoyé. Les protocoles proactifs sont adaptés aux applications qui nécessitent un prélèvement périodique des données. Et par conséquent, les capteurs peuvent se mettre en veille pendant les périodes d'inactivité, et n'enclencher leur dispositif de capture qu'à des instants particuliers.

2. Les protocoles réactifs

Dits aussi les protocoles de routage à la demande, créent et maintiennent les routes selon les besoins. Lorsqu'un nœud à besoin d'une route, une procédure de découverte globale est lancée. Cette procédure s'achève par la découverte de la route ou lorsque toutes les permutations de routes possibles ont été examinées. La route trouvée est maintenue par une procédure de maintenance de routes jusqu'à ce que la destination soit inaccessible à partir du nœud source ou que le nœud source n'aura plus besoin de cette route.

3. Protocoles hybrides

Ces protocoles combinent les deux idées des protocoles proactifs et réactifs. Ils utilisent un protocole proactif pour apprendre le proche voisinage (par exemple le voisinage à deux ou à trois sauts), ainsi, ils disposent de routes immédiatement dans le voisinage. Au-delà de la zone du voisinage, le protocole hybride fait appel à un protocole réactif pour chercher des routes.

Dans le reste de cette section, nous présentons une vue d'ensemble détaillée des principaux protocoles de routage dans les RCSFs appartenant à ces classes.

II.3 Exemples des protocoles de routage dans les RCSFs

II.3.1.Propagation et discussion (flooding and gossiping)

Pour le *flooding*, chaque capteur recevant un paquet de données le renvoie à tous ses voisins et ce processus continue jusqu'à ce que le paquet arrive à la destination ou le nombre maximum de sauts pour le paquet(TTL) est atteint. Ce protocole présente un ensemble d'inconvénients dont:

- ✓ *Implosion* : le même message est dupliqué plusieurs fois ; chaque nœud reçoit autant de fois la même donnée que le nombre de ses voisins ;
- ✓ *Chevauchement* : si deux nœuds observent le phénomène dans la même région, la même information sera envoyée deux fois(redondance des données) ;
- ✓ Il utilise aveuglement les ressources disponibles sans tenir en compte de leur quantité.

D'autre part, le *gossiping* est une version légèrement améliorée du *flooding* où le nœud récepteur envoie un paquet à un sous ensemble de ses voisins choisis aléatoirement. L'implosion n'est plus un inconvénient pour ce protocole. Cependant, l'envoi d'un message à tous les nœuds (un à un) demande plus de temps ;

II.3.2. Directed Diffusion(DD)

Directed Diffusion [37][38] est un protocole de routage de catégorie *data-centric*, permettant d'utiliser plusieurs chemins pour le routage d'information.Le principe de fonctionnement du protocole DD est le suivant :

Le noeud « *Sink* » commence à envoyer, vers tous les noeuds, un message *Interest* pour démarrer une application bien déterminée. Ce paquet sera acquitté par un autre appelé *gradient*. Un gradient est un lien de réponse de la part du voisin recevant l'intérêt. En utilisant les intérêts et les gradients, plusieurs chemins peuvent être établis entre le « *Sink* » et la source. L'un de ces chemins est sélectionné par renforcement. Si ce chemin échoue un nouveau ou un alternatif doit être identifié La figure II.4 suivante illustre les phases de fonctionnement de ce protocole

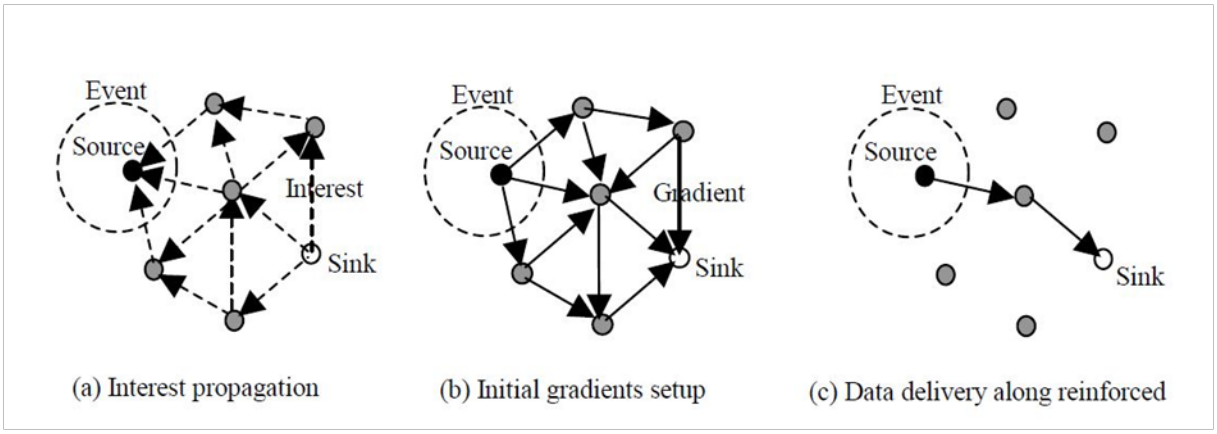

Figure II. 4. Les phases de communication du protocole Directed Diffusion

II.3.3 Rumor Routing(RR)

Rumor routing [39] est une variante du protocole Directed Diffusion. Généralement Directed Diffusion utilise une forme d'inondation pour la propagation des requêtes (intérêts). Cependant, dans certains cas il y a seulement une petite quantité de données demandée des nœuds et ainsi l'utilisation d'inondation est inutile. Une approche alternative est d'inonder les événements si le nombre d'événements est petit et le nombre des requêtes est grand. Le protocole Rumor Routing essaie de trouver un compromis entre l'inondation des intérêts et la propagation des données.

L'idée est de router des requêtes aux nœuds qui ont observé un événement particulier plutôt qu'inonder le réseau entier pour récupérer des renseignements sur les événements se produisant. Pour inonder l'événement à travers le réseau, l'algorithme Rumor Routing utilise le concept d'*agent*.

Un agent est un paquet avec une grande portée (TTL) qui traverse le réseau et informe de nœud en nœud en informant ces nœuds des événements qu'il a rencontrés durant toute cette traversé.

Quand un nœud détecte un événement, il l'ajoute à sa table locale et génère un paquet *agent*. Ce paquet parcourt le réseau afin de propager des informations sur les événements locaux aux nœuds distants. Quand un nœud génère une requête pour un événement, les nœuds qui connaissent l'itinéraire, peuvent répondre en se référant à la table d'événement.

II.3.4. Sensor Protocols for Information via Negotiation: (SPIN)

SPIN [40] est parmi les premiers protocoles de routage *data-centric* basé sur la *négociation*. SPIN est basé sur l'idée que les nœuds capteurs opèrent plus efficacement et conservent l'énergie en envoyant des données qui décrivent les données des capteurs au lieu d'envoyer les données entières, à moins que les données entières ne soient explicitement demandées. Cela permet de pallier au problème d'inondation. Pour cela, SPIN utilise trois types de messages : ADV(ADVertise), REQ (REQuest)et DATA.

Avant d'envoyer une donnée entière (message DATA),un nœud diffuse un message ADV qui contient la description ,c-à-d méta-données, de la donnée en question. Un nœud recevant un message ADV, consulte sa base d'intérêt. S'il est intéressé par cette information, il émet un message REQ vers son voisin (émetteur de ADV) .En recevant un message REQ, l'émetteur transmet à l'intéressé la donnée sous forme d'un message DATA. Les nœuds voisins répètent ainsi cette opération, comme le montre la figure II.5. Comme résultat, les nœuds qui sont intéressés par la donnée en auront une copie.

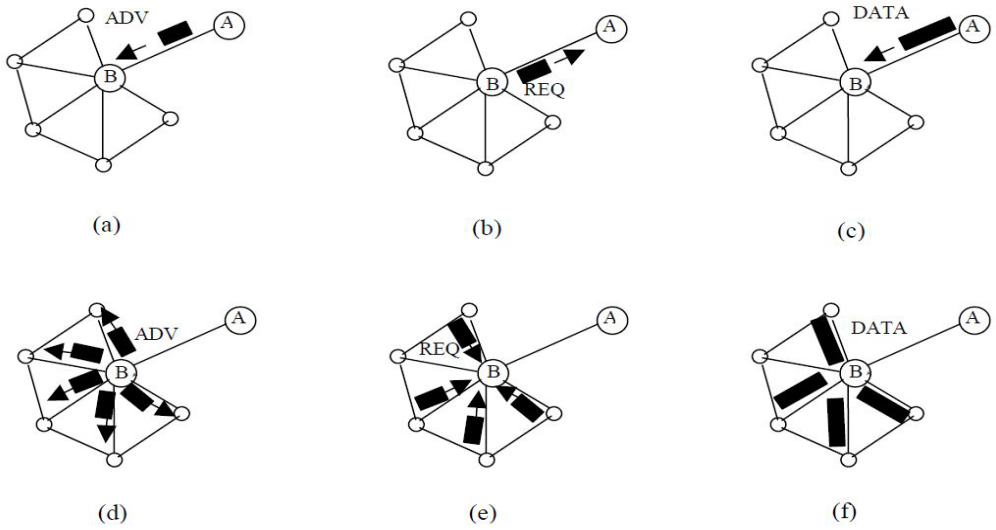

Figure II. 5 Le protocole SPIN

Sur cette la figure,le noeud A commence par aviser ses données au noeud B (a). Le nœud B répond en envoyant la requête au nœud B (b) .Après voir reçu la donnée demandé (c), le nœud B avise cette donnée à ses voisins (d), qui rendent à tour de rôle des requêtes à B (e-f).

Le protocole SPIN accommode son exécution suivant l'énergie restante du capteur, et modifie en conséquence le comportement du nœud. En effet, les nœuds contrôlent leur niveau d'énergie d'une manière continue. Lorsque le nœud s'aperçoit que son niveau d'énergie a atteint un certain seuil, il change son mode de fonctionnement, et ne répond à aucun message ADV.

Notons que le format des méta-données utilisé dans SPIN n'est standardisé (dépend de l'application)[41].

II.3.5.Low-Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [42] [43] est le premier et le plus populaire des algorithmes de routage hiérarchiques avec une efficacité énergétique, qui a été proposé pour les RCSFs [44]. LEACH emploie la technique de clustering qui divise le réseau en deux niveaux : les clusterheads(CHs) et les nœuds membres. Le protocole se déroule en rounds. Chaque round se compose de deux phases : construction et communication. La durée de la phase de communication est plus longue que celle de la phase de construction afin de minimiser l'overhead.

1. Phase de construction

Le but de cette phase est la construction des clusters en choisissant les chefs (CHs) et en établissant la politique d'accès au média au sein de chaque groupe. Durant cette phase, chaque nœud n choisit un nombre aléatoire compris entre 0 et 1.Si ce nombre est inférieur à un seuil T(n), le nœud devient clusterhead. T (n) est défini comme suit **[42]** :

$$T(n) = \begin{cases} \dfrac{p}{1 - p * \left(r \bmod \dfrac{1}{p}\right)} & si\ n \in G \\ 0 & Sinon \end{cases}$$

Avec:

p : pourcentage des nœuds désirant devenir clusterhead.

r: numéro du round courant.

G: ensemble de nœuds n'ayant pas été élus clusterheads durant les 1/P dernières périodes précédents. Par la suite, chaque nœud qui s'est élu clusterhead émet un message de notification à tous ces voisins. Les nœuds qui récoltent les messages de notification, décident

leur appartenance à un cluster. La décision est basée sur l'amplitude du signal reçu : le clusterhead ayant le signal le plus fort est choisi (i.e. le plus proche). En cas d'égalité, un chef aléatoire est choisi. Chaque membre informe son chef de sa décision.

***2.* Phase de communication**

Pendant cette phase, les nœuds capteurs peuvent commencer à envoyer leurs données captées au clusterhead pendant leurs propres slots. Cela leur permet d'éteindre leur interface de communication en dehors de leurs slots réservés, afin d'économiser leur énergie. Le clusterhead agrège les données reçues avant de les transmettre au collecteur (puits).Cette communication, entre un clusterhead et le collecteur, se fait d'une manière directe[1]. Après la phase de communication, qui dure un certain temps, la phase de construction recommence.

II.3.6 Cougar

Dans Cougar[45], le réseau est vu comme une base de données distribuée où quelques nœuds contenant les renseignements sont temporairement inaccessibles. Les données produites par le réseau de capteurs sont modélisées comme une table relationnelle. Dans cette table, chacun des attributs représente soit des informations sur le nœud capteur ou bien des données produites par ce nœud. Pour pouvoir interroger les nœuds du réseau, Cougar fournit une interface semblable à SQL(Structured Query Language) étendue, au niveau du puits. Cette interface permet au puits de générer des requêtes pour interroger un nœud spécial du réseau, appelé leader (chef). L'approche Cougar fournit une agrégation partielle au niveau des nœuds. Chaque nœud maintient une liste d'attente contenant les nœuds fils qui doivent lui envoyer les paquets. Le nœud n'émet le paquet agrégé au prochain saut que s'il a reçu les paquets de tous les nœuds de la liste d'attente. Cependant, un nœud peut devenir inaccessible à cause du mouvement ou d'un problème de batterie. Pour cela, Cougar utilise un timer afin d'éviter une attente indéfinie.

II.3.7 ACQUIRE (Active Query Forwarding in Sensor Networks)

Cet algorithme [46] considère aussi un RCSF comme une base de données distribuée. Dans ce plan, un nœud injecte un paquet de requête actif dans le réseau. Le voisin qui découvre que le paquet contient des informations obsolètes, émet un message de mise à jour au nœud source du paquet. Alors, le nœud choisit au hasard un voisin pour propager la

[1] LEACH suppose que les chefs de clusters sont capables d'atteindre la station de base à un seul saut.

requête. Comme la requête active progresse à travers le réseau, elle est progressivement résolue en plus petits composants de jusqu'à ce qu'elle soit complètement résolue. Alors, la requête est retournée au nœud source comme une réponse complète.

II.3.8 Geographic and Energy Aware Routing (GEAR)

Le protocole GEAR [47] découpe le réseau en régions. Chaque nœud connaît le coût pour atteindre chaque région. L'acheminement des paquets suit les étapes suivantes :

- ✓ Acheminer le paquet jusqu'à la région, en envoyant le paquet au nœud le plus proche de la région parmi ses voisins et ayant le niveau d'énergie résiduelle le plus élevé (fonction de distance et d'énergie),
- ✓ Acheminer le paquet dans la région de destination par une sorte de diffusion si le nombre de nœud n'est pas élevé, sinon la région est découpée en sous-région et le paquet est transmis individuellement à chaque sous-région.

Chaque paquet contient la région destination. Chaque nœud connaît sa position, son énergie résiduelle, la position et l'énergie résiduelle de ses voisins (à la demande). Un lien existe entre 2 nœuds quand ils sont à portée et leur niveau d'énergie leur permet d'effectuer l'envoi.

II.3.9 Sequential Assignment Routing (SAR)

SAR [48] est l'un des premiers protocoles de routage pour les RCSFs qui introduit la notion de qualité de service(QoS) dans les décisions de routage [49]. SAR est une approche multi-chemins qui s'efforce à réaliser l'efficacité énergétique et la tolérance aux fautes. Pour cela, SAR crée des arbres en prenant en compte les trois facteurs : métriques QoS, la ressource énergétique sur chaque chemin et le niveau de priorité de chaque paquet. En utilisant ces arbres, des routes multiples du *sink* aux capteurs sont formées. Une ou plusieurs routes peuvent alors être empruntées. Pour assurer la tolérance aux panes, SAR recalcule périodiquement les routes à choisir en cas de défaillance d'un nœud ou de changement de topologie [50] [51].

II.3.10 SPEED

SPEED [52], est un autre protocole de routage basé sur la QoS qui garantit le temps réel et le routage bout en bout. Une métrique supplémentaire par rapport à GEAR : le délai. En se

basant sur une table de positions, SPEED estime le délai sur chaque saut en calculant le délai d'aller-retour (en retranchant le temps de traitement côté récepteur).Autrement dit,SPEED assure une vitesse de livraison des paquets constante, qu'on note Setspeed. Ceci permet de garantir des délais de livraison de bout en bout acceptables. Après avoir déterminé le délai, le prochain saut est choisi parmi les voisins qui sont plus proches de la destination.

II.3.11 TEEN et APTEEN:

TEEN (Threshold sensitive Energy Efficient sensor Network protocol) **[54]** est un protocole hiérarchique conçu pour être sensible aux changements improviste des attributs détectés tels que la température. L'architecture du réseau est basée sur un groupement hiérarchique où les nœuds les plus proches forment des clusters. Après la construction des clusters, le clusterhead diffuse deux seuils aux nœuds. Qui sont la valeur minimale d'un attribut pour pouvoir être transmis et le degré minimale du changement de cet attribut.

APTEEN (*Adaptive Periodic TEEN*) **[53]** est une extension du TEEN basée sur la capture périodique des données et la réaction aux événements temps-réel. Quand la station de base forme les clusters, les cluster-heads diffusent les attributs, les seuils et le plan de transmission à tous les nœuds et effectuent également l'agrégation des données afin d'économiser l'énergie.

II.4 Conclusion

Le routage dans les RCSFs forme un axe de recherche intéressant, avec un ensemble limité, mais en pleine croissance de résultats des recherches. Cette croissance fortement liée à deux facteurs:

- ✓ Les capteurs sont des véritables systèmes embarqués et donc sont conçus pour les applications spécifiques,
- ✓ L'expansion du champ de domaine d'application des RCSFs (voir chapitre I).

Les protocoles de routage proposés pour les RCSFs sont donc nombreux, mais ils ont tous un objectif commun : Assurer l'acheminement des données collectées par les nœuds capteurs tout en essayant d'étendre la durée du réseau. Cela nécessite la prise en compte des caractéristiques des RCSFs et des exigences des applications pour lesquelles ces réseaux sont destinés.

Dans ce chapitre, nous avons identifié et classifié quelques protocoles de routage pour les RCSFs proposés dans la littérature. De façon générale, ces protocoles sont classifiés selon la structure du réseau, le fonctionnement des protocoles, l'établissement de la route, ou les paradigmes de communication.

Chapitre III

III.1 Introduction

Comme nous l'avons souligné dans le premier chapitre de ce travail, le modèle de communication en amont[1]dans les RCSFs est de type *many-to-one* (nœuds sources vers le puits) tandis que celui en aval est de type *one-to-many multicast*. En raison de la nature convergente d'un trafic en amont, ce modèle de communication crée un trafic intense de flux de données dans la région proche du puits que dans le reste du zone de captage (à la périphérie).Ainsi, les nœuds proches du puits devront effectuer plus de tâches que les autres. Etant limités en ressources, il n'est pas rare que ces nœuds (proches du puits) se trouvent dans l'état où ils ne peuvent plus effectuer toutes ces tâches. Le terme utilisé pour désigner cette situation est la *congestion.*

Dans les réseaux sans fil, particulièrement dans les RCSFs, il existe plusieurs sources de congestion [55] ; comme le débordement des mémoires tampon, les transmissions concurrentes, collision des paquets. Parmi les problèmes causés par la congestion, on peut citer : le retarder l'information, perte de paquets qui contiennent parfois les informations critiques, gaspillage de la bande passante. Il est facile de constater que la congestion dégrade les performances du réseau en l'empêchant de garantir certaines exigences de Qos comme le temps réel, le routage de bout en bout, la maximisation de la durée de vie du réseau. Ce problème motive le besoin de mise en place des mécanismes de contrôle de congestion pour améliorer la performance et prolonger la durée de vie du système [56].

Notre objectif dans ce chapitre est de faire une étude bibliographique sur les protocoles ou techniques de contrôle de congestion proposées dans la littérature, afin de voir leur comportement dans certaines applications.

Pour cela, le reste de ce chapitre est organisé comme suit. Premièrement nous donnons quelques notions sur la détection et le contrôle de congestion dans les RCSFs. Ensuite nous présenterons quelques protocoles et techniques de contrôle de congestion avant la conclusion de ce chapitre.

[1] La communication en amont (upstream communication) désigne une communication qui s'effectue dans la direction des nœuds sources d'information vers le nœud puits.Par contre la communication en aval (downstream communication) s'effectue dans le sens inverse.

III.2. Définition et typologies de congestion

Dans un réseau informatique, la congestion est provoquée par les sources excédant le lien de communication ou la capacité (de stockage ou de traitement) des éléments du réseau. Pour cela, deux types de congestion pourraient se produire dans un RCSF [56] : la congestion au niveau du nœud (node-level congestion) et la congestion au niveau du lien (link-level congestion).

La congestion au niveau du nœud, qui est répandue dans les réseaux conventionnels, est provoquée par le débordement des tampons dans le nœud et peut causer la perte et le retard des paquets.

Dans les RCSFs, le canal de communication sans fil est partagé par plusieurs nœuds en utilisant les protocoles de contrôle d'accès au média de la famille CSMA. Les collisions pourraient se produire quand les nœuds capteurs essaient de transmettre en même temps suite à la détection d'un un événement critique (incendie dans la forêt, séisme, etc.).

Dans cette situation, on parle de congestion au niveau du lien. Cette congestion conduit à un gaspillage de la bande passante et à un taux d'erreur élevé des paquets lors de leur réception.

La limitation de congestion suit généralement deux étapes [55]: détection de congestion et contrôle de congestion. La détection exacte et efficace de congestion joue un rôle essentiel dans le contrôle de congestion dans les RCSFs [55].

Généralement il existe deux approches de contrôle de congestion [56] : La gestion des ressources du réseau et la régulation du trafic. La gestion des ressources du réseau essaie d'augmenter ses ressources (mémoire tampon, puissance d'émission de l'interface de communication, vitesse de traitement, etc) pour atténuer la congestion quand elle se produit.

La régulation du trafic implique le contrôle de congestion par ajustement du taux de circulation des paquets afin qu'il s'adapte aux nœuds sources ou intermédiaires. La plupart des protocoles de contrôle de congestion existants appartiennent à ce type. La régulation du trafic peut être « bout en bout » ou « saut par saut ».Dans le bout en bout, la régulation s'effectue au niveau du nœud source pour simplifier la tâche aux nœuds intermédiaires ; il en résulte une réponse lente et dépend fortement du temps d'aller-retour. En « saut par saut » la

réponse est très rapide. Il est généralement difficile d'ajuster le taux de transmission des nœuds intermédiaires car ce taux dépend du protocole MAC et peut être variable.

III.3 Classification des approches de contrôle de congestion

Nous pouvons différencier les protocoles de contrôle de congestion à travers plusieurs axes [56] qu'on va décrire dans cette section.

III.3.1 Mécanisme de détection de congestion

Le mécanisme de détection de congestion peut être local ou global. La détection de congestion locale est réalisée aux nœuds intermédiaires en contrôlant des indicateurs locaux de congestion tels que l'occupation de la file d'attente ou l'état de canal. D'autre part, la détection de congestion globale est réalisée au niveau du puits où les attributs de bout en bout tels que les retards inter-paquets (inter-packet delays) et la fréquence de pertes peuvent être utilisés pour déduire la congestion.

III.3.2 L'objectif du contrôle de congestion

Dans leur nature, les RCSFs sont orientés application (application specific).Donc, les protocoles de congestion seront différents selon l'application visée par un RCSFs où ils sont appliqués. Pour cette raison, les protocoles de contrôle de congestion sont aussi orientés application.

III.3.3 Les mécanismes de contrôle de taux de transfert

Les mécanismes de contrôle de taux dans les RCSFs peuvent être centralisé (centralised), contrôle de la source (source-control) et *hop-by-hop backpressure*.Le mecanisme de contrôle de la source est réalisé par le puits). Essentiellement, quand la congestion (ou le premier signe de congestion) est découvert, le puits donne l'ordre aux nœuds sources de régler leurs taux. Alors que, dans *hop-by-hop backpressure* le mécanisme est réalisé aux nœuds intermédiaires, dans lesquels le nœud intermédiaire donne l'ordre aux nœuds qui sont en son amont de régler leurs taux en se basant sur son état de congestion local.

III.3.4 Equité et/ou QoS

Classiquement, les protocoles de contrôle de congestion sont chargés de réduire le taux de transmission afin d'éviter ou de réduire la congestion. En plus de cette tâche, d'autres exigences peuvent être envisagées. Cela inclut par exemple, les approches qui essaient de maintenir l'équité des flux opposés quant la congestion survient. De même, les approches QoS essaient d'allouer les ressources selon l'importance du flux ou les niveaux de réservation du canal.

Les différentes notions d'équité peuvent être utiles, selon l'application. Celles-ci peuvent être celles qui garantissent que tous les nœuds dans le réseau fournissent la même quantité de données (exemple, dans une simple application de collecte de données : simple data-gathering application), de l'équité maximum-minimum, de l'équité proportionnelle [57].

III.3.5 Modèle de l'application cible

La plupart des protocoles se basent sur le modèle de communication many-to-one. Cependant, quelques protocoles diffèrent sur leurs hypothèses dans ce modèle, telles que la supposition du flux à haut débit (high-rate flows), ou multiples requêtes à multiples puits (multiple queries to multiple sinks),etc.

III.3.6 D'autres métriques

D'autres métriques peuvent différencier les protocoles de contrôle de congestion. Par exemple, quelques protocoles nécessitent un support additionnel (MAC spécialisé, ou capacité réseau supplémentaire). De plus, quelques protocoles font l'attention spéciale à l'efficacité énergétique.

III.4 Quelques protocoles de contrôle de congestion dans les RCSFs

Dans le passé récent, de nouveaux protocoles de contrôle de congestion ont été proposés [58].Dans le paragraphe suivant nous allons faire une étude un peu détaillée sur certains d'entre eux.

III.4.1 Congestion detection and avoidance: CODA

CODA [59] est une technique de contrôle de congestion pour les RCSFs qui comprend trois mécanismes :

1. Détection de congestion basée sur le récepteur (Receiver-based congestion detection)

L'occupation du tampon a été abondamment utilisée dans les algorithmes de détection de congestion traditionnels comme une mesure de niveau de congestion. Dans leur algorithme, les auteurs démontrent que l'occupation du tampon seule n'est pas une bonne mesure de congestion dans les réseaux sans fil à cause de la nature partagée du canal. La file d'attente peut se décongestionner potentiellement même si les paquets sont perdus en raison de la collision. Il est possible aussi pour les nœuds de déterminer la congestion en écoutant le canal et déterminer comment il est occupé/chargé .Cependant, cela peut avoir un coût énergétique significatif. Cependant, l'écoute continue encourt le haut prix d'énergie. Donc, la CODA utilise un plan d'échantillonnage qui active la surveillance du canal local surveillant seulement sous de certaines conditions, par exemple seulement quand le tampon d'envoi n'est pas vide, pour économiser l'énergie.

2. Open-loop hop-by-hop back-pressure

Quand le récepteur détecte une congestion, il envoie un message de suppression (une notification de congestion explicite),appelé « backpressure signal » ,en anglais, vers la source. Le message de suppression est envoyé à plusieurs reprises tant que l'état de congestion persiste. Les nœuds peuvent répondre à ce message en supprimant des paquets ou en réduisant leur taux. Le message de suppression peut se propager entièrement jusqu'à la source, ou atteindre seulement les nœuds intermédiaires selon leur état de congestion local.

3. Closed-loop multi-source regulation

Ce mécanisme de contrôle de congestion est utilisé par le puits pour intervenir dans la régulation des sources multiples, dans le cas où la congestion est persistante. Essentiellement, quand le taux de transmission d'une source excède le débit théorique maximum, S_{max}, la source informe le puits par un bit qu'elle met dans chaque paquet qu'elle transmet au puits tant que le taux de transmission reste supérieur à S_{max}.En réponse, le puits commence à envoyer les ACKs à la source jusqu'à ce qu'il détecte la congestion. Quand le puits détecte la congestion, il arrête d'envoyer les ACKs jusqu'à l'atténuation de la congestion, pour

implicitement informer l'expéditeur de baisser son taux de transmission. En général, les sources maintiennent, diminuent, ou augmentent leurs taux selon la fréquence de réception des ACKs.

III.4 .2 Fair rate allocation(FRA)

Fair rate allocation, ou allocation équitable de taux de transmission Est une approche explicite au contrôle de congestion avec garantie d'équité proposée par Ee et Bajcsy [60]. Le mécanisme de FRA comprend les trois démarches suivantes :

Déterminez le taux moyen, r, de transmission d'un paquet : En supposant que les paquets ont la même taille, le taux de transmission du paquet peut être estimé comme l'inverse de l'intervalle de temps de transmission d'un seul paquet. L'intervalle est mesuré à partir du moment où la couche transport envoie le paquet à la couche réseau jusqu'à au moment où la couche réseau signale que le paquet a été transmis.

Assignez le taux r aux nœuds en amont (c-à-d les nœuds fils dans l'arbre de collecte de données) : Le taux moyen de transmission de paquet est divisé par le nombre, n , de nœuds fils pour assigner le taux de génération de paquet de données comme $r_{data} = \frac{r}{n}$

Pour calculer n, chaque nœud inclut la taille de son sous-arbre(le nombre de ses nœuds fils) dans un paquet et l'envoie au parent. Le parent décompte les nombre de ces descendants, y ajoute un (si le parent lui-même produit des données) et inclut le total dans le paquet avant de l'envoyer vers le puits.

Quand les files d'attente débordent ou sont au point de déborder, le nœud assigne un taux de génération de paquet inférieur aux nœuds qui sont en son amont.

Obtenir le taux du nœud parent r_{data_parent} par l'écoute du canal ou via un message de contrôle. Comparer r_{data} avec r_{data_parent} et propager le plus petit taux aux nœuds du sous-arbre.

L'équité proportionnelle est obtenue en mesurant et en divisant le taux par le nombre de nœuds en aval. Il s'agit donc de l'équité proportionnelle Pour réaliser cela, chaque nœud maintient une file d'attente de type FIFO pour chaque nœud fils comme le montre la figure III.1. Alors, un mécanisme de sélection probabiliste est employé pour mesurer le poids du

choix des paquets. Le choix de la file d'attente à partir de laquelle le paquet sera transmis est proportionnel au nombre de nœuds entretenus par cette file d'attente.

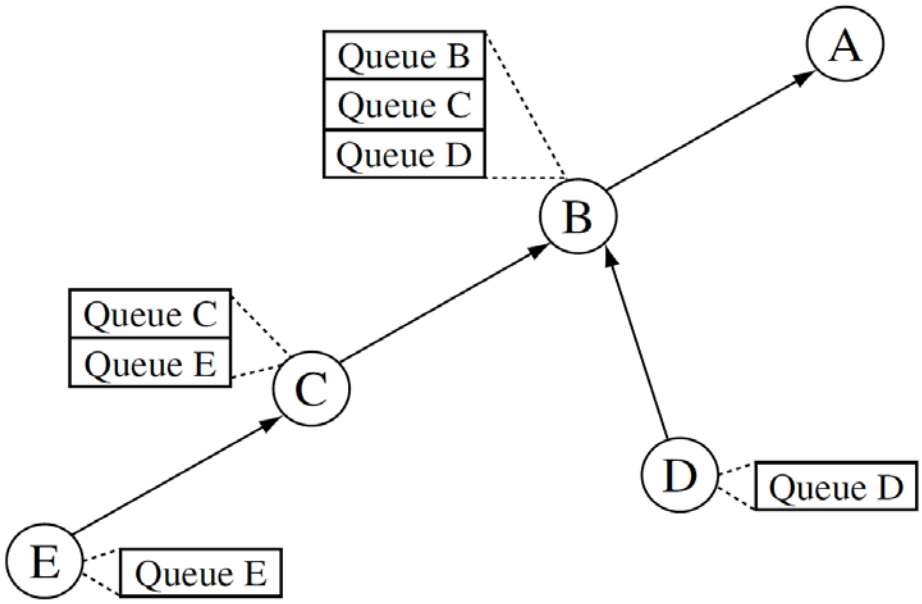

Figure III. 1. FIFO multiples pour assurer la délivrance équitable de données dans FRA

III.4 .3 Event-to-Sink Reliable Transport (ERST)

Le protocole ESRT [61] se focalise sur l'ajustement du taux d'activité des nœuds sources afin d'assurer la fiabilité souhaitée par le puits, avec l'utilisation minimale de ressources. ESRT suppose que le puits est assez puissant pour atteindre tous les nœuds sources par diffusion. L'idée clé dans ESRT est que le puits ordonne les nœuds sources d'ajuster leur fréquence d'activité selon la fiabilité mesurée au niveau du puits et de l'état de congestion dans le réseau. ESRT piste deux paramètres : (1) l'indicateur d'intégrité, calculé par le puits; et (2) l'état actuel de congestion. Le puits calcule $\boldsymbol{\eta}$ pour la période i comme suit

$$\eta_i = \frac{r_i}{R_i}$$

Où r_i est la fiabilité d'événement observée et Ri est la fiabilité d'événement par le puits.

Pour informer le puits de l'état actuel de congestion, chaque nœud capteur contrôle la taille de sa file d'attente et met le bit de congestion dans le paquet à envoyé s'il constate que prochain paquet de données risque de causer un débordement de sa file d'attente.

En se basant sur ces paramètres, L'algorithme ESRT établit un diagramme de transition à cinq états comme le montre la figure III.2 .Les états ont les significations suivantes :

- ✓ *No Congestion, Low Reliability (NC, LR):*

Le réseau n'est pas congestionné, mais la fiabilité observée est inférieure à la fiabilité souhaitée. Dans ce cas, les sources doivent augmenter leurs taux d'activité pour augmenter la fiabilité.

- ✓ *No Congestion, High Reliability (NC, HR):* Le réseau n'est pas congestionné, mais la fiabilité observée est supérieure à la fiabilité souhaitée. Ainsi, le puits ordonne aux nœuds sources de réduire leurs taux d'activité prudemment, pour maintenir la fiabilité exigée, mais avec moins d'overheards.

- ✓ *Congestion, High Reliability (C,HR):*Le réseau est congestionné et la fiabilité est supérieure à celle souhaitée. Dans ce cas, les nœuds doivent réduire leurs taux jusqu'à ce que la congestion soit résolue ou la fiabilité tombe en dessous du niveau souhaité.

- ✓ *Congestion, Low Reliability (C, LR):* C'est le pire état possible, dans lequel ESRT réduit exponentiellement la fréquence d'activité pour alléger la congestion et potentiellement améliorer la fiabilité.

- ✓ *Optimal Operating Region (OOR):* C'est la région d'exploitation optimale où le taux d'activité est suffisant juste pour atteindre la fiabilité souhaitée. Plus précisément,

$$1 - \varepsilon \leq \boldsymbol{\eta_i} \leq \mathbf{1} + \varepsilon$$

Où Ɛ est une petite marge d'erreur utilisée pour assurer la stabilité. Le but d'ESRT est de maintenir toujours l'état du réseau dans OOR.

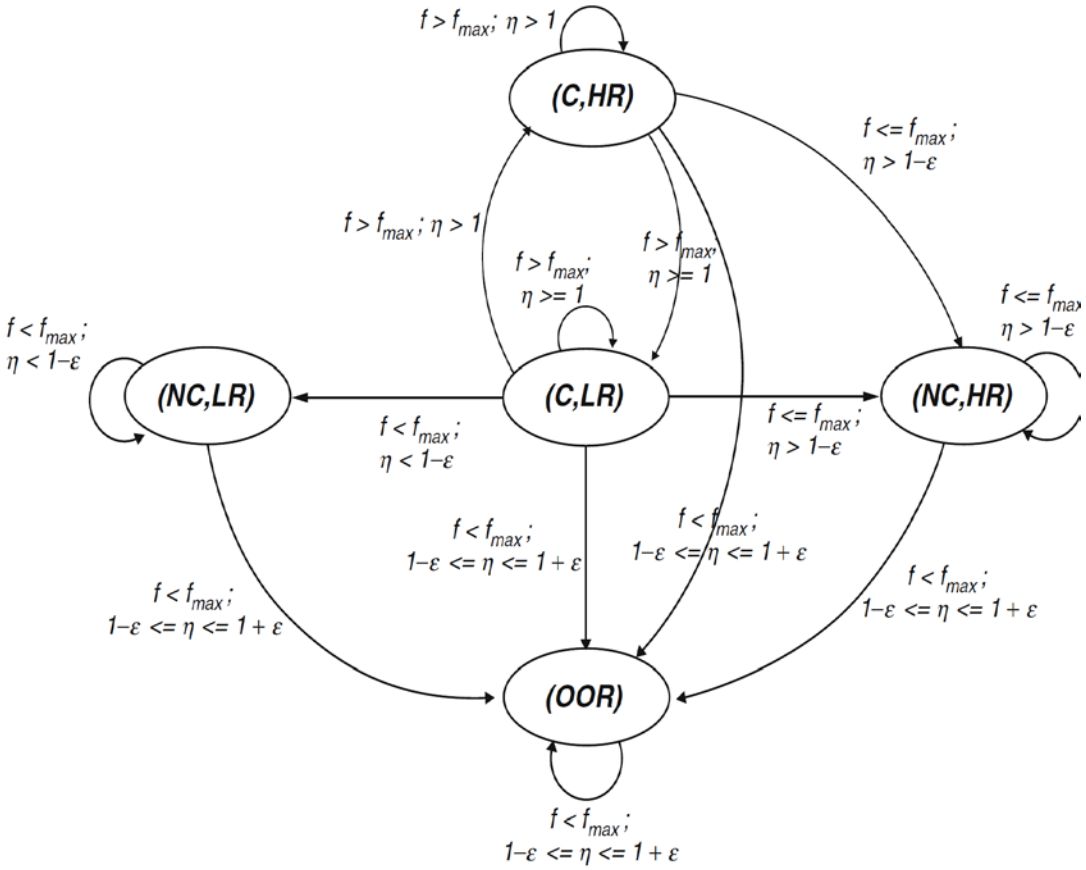

Figure III. 2. Diagramme de transition du protocole ESRT

III.4.4 Autres protocoles

Le nombre élevé des protocoles de contrôle de congestion ne nous donne pas la possibilité de les explorer tous. Néanmoins nous avons pu détailler quelques uns dans les paragraphes précédents. Le choix des algorithmes détaillés dépend de leurs divergences et surtout de leur importance, selon notre point de vue, à résoudre ou à alléger le problème de congestion. Il existe d'autres algorithmes qui, la plupart, se basent sur ceux que nous avons a vus précédemment. Nous les résumons dans les paragraphes suivants.

Dans Fusion [62], basé sur CODA, les auteurs proposent un nouveau protocole MAC avec priorité (prioritezed MAC).L'idée principale de Fusion est que à la présence de congestion, le protocole MAC de la famille CSMAC classique qui alloue de façon équitable l'accès au canal de transmission à tous les nœuds n'est pas adapté.

Dans [63], les auteurs proposent le protocole IFRC (Interferance-Aware Fair Rate Control) qui prend en considération les interférences pour contrôler la congestion.

Dans [64], les auteurs partent de l'idée que dans un RCSF les événements captés peuvent être d'importances différentes pour proposer le protocole COMUT (Congestion Control Based on Importance of Data).

III.5 Conclusion

Le modèle de communication many-to-one dans les RCSF, le partage du canal de transmission ainsi que la limite des ressources dont sont équipés les nœuds capteurs sont les principaux éléments qui peuvent conduire à un réseau de capteur congesté. La congestion, lorsqu'elle existe dans un réseau, dégrade les performances du système et peut causer beaucoup de problèmes.

Dans ce chapitre nous avons identifié quelques de ces problèmes parmi lesquels une perte des paquets, parfois contenant l'information critique (par exemple l'état de santé d'un patient), le gaspillage de la bande passante ainsi que d'énergie utilisés pour transmettre ces paquets perdus. Nous avons aussi étudié quelques les protocoles proposés pour détecter et/ou contrôler la congestion dans les RCSF.

Le point commun de ces protocoles est que, lorsqu'une congestion ou le premier signe de congestion apparait au niveau d'un nœud, ce dernier procède à une régulation du taux d'activité des nœuds sources des messages dont il est le destinataire.

Malgré que ces protocoles ont proposé une amélioration significative aux techniques classiques de contrôle de congestion qui se basaient presque uniquement sur le taux d'occupation du tampon de données au niveau du destinataire des messages pour détecter la congestion ,ces protocoles peuvent ne pas garantir la non perte des paquets dans le cas où une congestion au niveau d'un nœud persiste.

En effet, dans ce genre de protocoles, lorsqu'une congestion au niveau d'un nœud persiste, quelques nœuds amont (fils dans l'arbre de collecte de données) peuvent réduire leur taux de transfert jusqu'à ne pas transmettre pendant un certain temps et si un capteur a généré plus d'un message pendant ce temps, l'information contenue dans les premiers messages ne serait plus valide.

Dans le premier chapitre de ce mémoire, nous avons vu que l'une des propriétés d'un RCSF est qu'il est constitué par un grand nombre e nœuds déployés, parfois aléatoirement, dans une zone de captage et que selon la densité des nœuds du réseau et la nature du phénomène mesuré, ces nœuds sont susceptibles de capter et de transmettre la même donnée ou les données présentant un écart de valeur non significatif.

La transmission de la même donnée par les capteurs crée une redondance de données au niveau des nœuds intermédiaires utilisés pour transmettre ces données au nœud puits. Cette redondance, selon son intensité, peut être une source sérieuse de congestion.

Le problème qui se pose dans les protocoles étudiés dans ce chapitre, est que, à notre connaissance, aucun d'entre eux ne prend en considération cette redondance. Ce qui nous motive à proposer un nouveau protocole exploitant cette propriété des RCSF pour contrôler la congestion.

La technique utilisée pour réduire ou éviter cette redondance est connue sous l'appellation d'agrégation des données et va être décrite en détails dans le chapitre suivant avant de proposer notre protocole.

Chapitre IV

IV.1 Introduction

Selon leur mode de déploiement, les capteurs peuvent être disséminés dans une portion de leur zone d'intérêt où ils recueilleront les données fortement corrélées ou les mêmes données. La transmission des toutes ces données créerait une redondance de données au niveau des nœuds utilisés comme leurs points de relais vers le puits. Avec les moyens de traitement dont ils sont équipés, les capteurs peuvent réduire ces redondances, en combinant les paquets contenant les données redondantes. Cette technique s'appelle l'agrégation de données ou fusion de données.

Il ya deux raisons principales pour lesquelles un capteur ne devrait pas envoyer directement la donnée brute captée au nœud puits [62]:La première raison est que le taux de transfert de données dans un RCSF est limitée. La deuxième est que la communication est plus consommatrice en énergie que le traitement. En effet, il a été montré dans plusieurs publications scientifiques que la transmission d'un seul bit est équivalente, en termes d'énergie, à l'exécution d'environ 1000 instructions [66] et que cette valeur augmente avec la portée du module radio. Il convient donc de réduire cette énergie en agrégeant les données dans leur routage.

Le problème des données redondantes est que non seulement elles sont la source de gaspillage de l'énergie mais aussi dans la plupart des cas elles sont source de congestion dans le réseau. Un bon algorithme d'agrégation des données pourrait donc réduire ou éviter considérablement cette congestion.

L'objectif de ce chapitre est de passer en revue de certains algorithmes d'agrégation de données proposés dans le passé est de voir s'ils peuvent être utilisés pur réduire la congestion dans quelques genres d'applications.

Le reste de ce chapitre est organisé comme suit :

En premier lieu nous verrons quelques notions générales sur les techniques d'agrégation de données. Ensuite nous ferrons une étude sur les techniques d'agrégation existant à travers une certaine classification et nous finirons par une conclusion.

IV.2 Définition

L'agrégation données est un processus global de collecte et de routage d'information par un réseau multi-sauts, en traitant des données aux nœuds intermédiaires avec l'objectif de réduire la consommation de ressources et ainsi augmenter la durée de vie du réseau [67].

IV.3 Performances et limites d'une technique d'agrégation

Plusieurs études théoriques fournissent des limites sur la performance de techniques d'agrégation de données dans le réseau et aident ainsi la conception d'algorithmes convenables [67]. L'efficacité de ces algorithmes dépend de la corrélation entre les données produites par de différentes sources d'information (les unités de captage). Une telle corrélation peut être spatiale, quand les valeurs produites par les capteurs voisins sont rattachées, temporelles, quand les lectures de capteurs changent lentement au fil du temps, ou sémantique, quand les contenus de différents paquets de données peuvent être classifiés sous le même groupe sémantique (par ex, les données produites par les capteurs placés dans la même pièce). Les augmentations d'agrégation de données dans le réseau peuvent être le mieux démontrées dans le cas extrême quand les données produites par de différentes sources peuvent être combinées dans un paquet simple (par ex, quand les sources produisent des données identiques

IV.4 Approches d'agrégation de données

Dans un RCSF on distingue deux approches d'agrégation de données [67][68] :agrégation sans et avec réduction de la taille des données .

- ✓ *Agrégation avec réduction de la taille (In-network aggregation with size reduction) :* fait allusion au processus de combinaison des données venant de différentes sources pour réduire la quantité de données à envoyer dans le réseau. Comme exemple, supposez qu'un nœud reçoit deux paquets de deux différentes sources contenant les températures localement mesurées. Au lieu d'envoyer les deux paquets, le capteur peut calculer la moyenne des deux lectures et l'envoyer dans un seul paquet.
- ✓ *Agrégation sans réduction de la taille (In-network aggregation without size reduction) :* fait allusion au processus de fusion des paquets venant de différentes sources dans le même paquet sans aucun traitement sur les données. Supposez par exemple qu'un noud reçoit deux paquets contenant les données qui décrivent les

phénomènes physiques différents, par exemple, l'humidité et l'acidité du sol. Ces deux valeurs ne peuvent pas être traitées ensemble mais elles peuvent être transmises dans un seul paquet simple et réduire ainsi l'overhead.

L'avantage de la première approche est qu'elle permet de réduire au maximum la quantité de données échangées. Son principal inconvénient est que après l'opération d'agrégation, il n'est pas toujours possible de reconstituer tous les paquets originaux[1].La seconde approche, par contre, préserve les paquets originaux (c.à.d.,au niveau du puits, les paquets originaux peuvent être reconstruits). Le choix de l'approche à utiliser dépend de plusieurs facteurs dont : type d'application, taux de transfert de données, les caractéristiques du réseau.

IV.5 Les éléments de base de l'agrégation de données

Les techniques d'agrégation dans le réseau exigent trois éléments fondamentaux [67] : un protocole de routage convenable, les fonctions d'agrégation efficaces et une façon efficace de représenter les données, voir figure IV.1 ci-dessous. Dans le reste de cette section nous décrivons chacun de ces aspects.

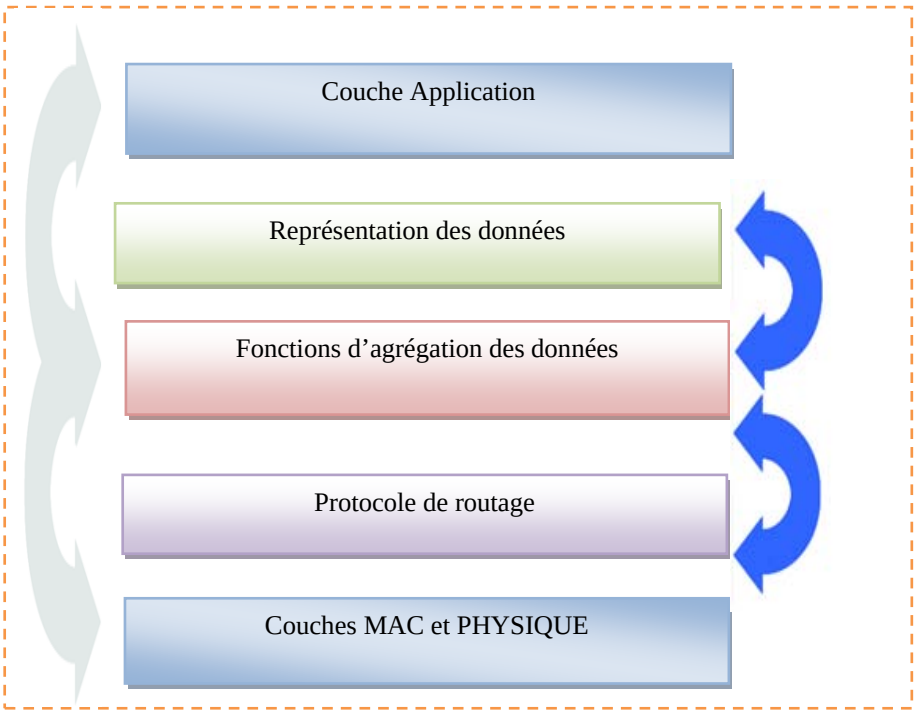

Figure IV.1 Relation entre la technique d'agrégation et d'autres couches de la pile

[1] Tous dépend de la fonction d'agrégation utilisée ,c.à.d, avec ou sans perte.

Notons que dans la plupart des cas, la technique d'agrégation interagit avec les couches application, Physique et MAC, comme le montre la figure ci-dessus.

IV.5.1 Fonctions d'agrégation de données.

L'une des fonctionnalités les plus importantes que les techniques d'agrégation dans le réseau devraient fournir est la capacité de combiner des données venant de différents nœuds. Il y a plusieurs types de fonctions d'agrégation et la plupart d'entre elles sont fortement dépendantes de l'application cible. Néanmoins, nous pouvons identifier quelques paradigmes communs pour leur classification [67][68] :

- ✓ *Insensibles à la duplication Vs sensible à la duplication*

Dans un réseau de capteur sans fil, un nœud intermédiaire peut recevoir des copies multiples d'une même information. Dans ce cas, il arrive que cette même information soit considérée plusieurs fois lors de l'agrégation la fonction d'agrégation utilisée est sensible à la duplication (duplicate sensitive), le résultat final dépendra du nombre de fois que la même valeur a été considérée. Dans le cas contraire, la fonction d'agrégation est dite insensible à la duplication (duplicate insensitive).Par exemple, les fonctions telles que MIN et MAX qui calculent respectivement la valeur minimale et maximale des données reçues au niveau d'un nœud sont insensibles à la duplication tandis que les fonctions telles que SUM et AVG, qui calculent respectivement la somme et la moyenne de ces mêmes données sont sensibles à la duplication.

- ✓ *Sans perte Vs Avec perte*

L'agrégation de données peut se faire avec ou sans perte d'information. L'agrégation avec perte ne permet pas de reconstruction parfaite mais l'agrégation sans pertes garantit une récupération complète de toutes les données de capteur individuelles à la station de base (puits) [69].

De bonnes fonctions d'agrégation pour les réseaux de capteur sans fil ont besoin de satisfaire des besoins supplémentaires. En particulier, elles devraient tenir compte du traitement très limité et des capacités d'énergie des nœuds intermédiaires et devraient être donc implémentées au moyen des opérations élémentaires.

IV.5.2 Protocoles de routage avec agrégation des données

L'ingrédient le plus important pour l'agrégation dans le réseau est un protocole de routage bien conçu [67]. Comparativement au routage classique, la prise en compte d'agrégation des données exige de nouveaux paradigmes pour acheminer les données .Par exemple, les techniques d'agrégation dans le réseau peuvent exiger une forme de synchronisation parmi les nœuds. En particulier, la meilleure stratégie d'un nœud donné n'est pas toujours d'envoyer des données aussi tôt que c'est disponible. L'attente d'information venant des nœuds voisins peut mener à de meilleures techniques d'agrégation des données et, ainsi, améliorer la performance. Selon la stratégie utilisée pour choisir le moment où un nœud doit retransmettre la donnée agrégée, nous pouvons distinguer trois types d'agrégation : [70] : Agrégation périodique simple, Agrégation périodique par saut et Agrégation périodique par saut ajusté.

- ✓ *Agrégation périodique simple (Periodic simple aggregation) :*

chaque nœud doit attendre une période de temps prédéterminé, pour agréger tous les paquets reçus avant de transmettre le résultat de l'agrégation.

- ✓ *Agrégation périodique par saut (Periodic per-hop aggregation) :*

est similaire à l'agrégation périodique simple, la seule différence est que la donnée agrégée est transmise aussitôt que le nœud entend de tous ses enfants(les nœuds fils dans l'arbre de collecte de données) .Cela nécessite que chaque nœud connaisse le nombre de ces enfants. En plus, un temps mort (timeout) est utilisé dans le cas où il y a une perte de paquets de quelques nœuds enfants.

- ✓ *Agrégation périodique par saut ajustée (Periodic per-hop adjusted aggregation)*

Ajuste le temps mort d'un nœud, après lequel il envoie les données agrégées, selon la position du nœud dans l'arbre se de collecte de données.

Notons que le choix de la stratégie à utiliser affecte fortement la conception d'un protocole de routage [70][71].

Même si nous avons présenté les protocoles de routages et les techniques d'agrégation dans deux chapitres différents, notons que ces deux notions sont fortement liées. Dans le deuxième chapitre, nous avons vu les protocoles de routage dans le cas général. Un certain nombre d'entre eux réalisent l'agrégation des données. Nous pouvons citer comme exemples

les protocoles: LEACH, Directed Diffusion, PEGASIS et COUGAR (voir Chapitre II) .Etant données que ces protocoles ne sont pas conçus pour assurer l'agrégation des données comme un service privilégié, ces protocoles utilisent des fonctions simples comme Moyenne, ou Maximum. Dans ce qui suit, nous essayons de donner un aperçu sur quelques protocoles de routages avec agrégation des données en les classifiant quatre catégories : hiérarchique, basés sur les clusters, multi-chemins et hybrides.

IV.5.2.1 Approche hiérarchique

La plupart des travaux faits sur l'agrégation de données dans les RCSFs ont proposé des solutions exploitant une structure hiérarchique (arborescente) [72].En effet, la façon la plus simple d'agréger des données venant des sources vers le puits est d'élire quelques nœuds spéciaux qui travaillent comme les points d'agrégation et définissent une direction préférée à être suivie en envoyant des données.

Selon l'approche hiérarchique [73] [74] [75], un arbre s'étendant à partir du puits est construit d'abord. Par la suite, une telle structure est exploitée en répondant aux requêtes produites par le puits. Ceci est fait en performant l'agrégation le long de l'arbre d'agrégation en procédant niveau par niveau de ses feuilles à sa racine. Ainsi, puisque au moins deux messages arrivent à un nœud donné, le résultat de leur agrégation peut être calculé exactement.

Dans les paragraphes suivants, nous considérons les principaux algorithmes de routage basées sur les arbres d'agrégation de données.

1. Tiny AGregation (TAG)

TAG [76] est un protocole data centric basé sur les arbres d'agrégation et est spécifiquement conçu pour l'application de surveillance .Cela signifie chaque nœud devrait produire l'information périodiquement. Donc, il est possible de classifier TAG comme un protocole à Agrégation périodique par saut ajustée.TAG se compose de deux phases : une phase de *distribution*(distribution phase), dans laquelle les requêtes sont disséminées et une phase de *collecte* (collection phase), où les valeurs agrégées sont continuellement routées vers le haut.

Pendant la phase de distribution, le puits diffuse un message en demandant aux nœuds de s'organiser dans un arbre de routage et envoie ensuite ses requêtes. Dans chaque message il

y a un champ spécifiant le niveau (level), ou la distance depuis racine, qui est incrémenté à chaque fois qu'un nœud reçoit un message et le rediffuse aux autres nœuds voisins. Le niveau du puits est égal à zéro. Ce processus continue jusqu'à ce que tous les nœuds aient été assignés identificateur et un parent.

TAG adopte la sélection et l'agrégation offertes par les langages d'interrogation de bases de données (SQL). Par conséquent, les requêtes de TAG ont la forme suivante :

SELECT{agg(expr), attrs} from SENSOR
WHERE{selPreds}
GROUP BY{attrs}
HAVING{havingPreds}
EPOCH DURATION i

En pratique, le puits envoie une requête, où il spécifie les quantités qu'il veut recueillir (le champ *attrs*), comment celles-ci doivent être agrégées (*agg (expr))* et les capteurs qui devraient être impliqués dans l'extraction de données. Cette dernière demande est spécifiée par les clauses WHRERE, GROUP BY et HAVING [76]. Finalement, un champ de durée *EPOCH* spécifie le temps (en secondes) que chaque capteur devrait attendre avant d'envoyer de nouvelles lectures de captage. Cela signifie que les utilisées pour calculer le résultat de l'agrégation appartiennent toutes au même intervalle de temps, ou epoch.

Pendant la phase de collecte de données, en raison de la structure arborescente, chaque parent doit attendre des données de tous ses enfants, pendant un temps égal à epoch, avant qu'il puisse envoyer le résultat de l'agrégation à son parent. Les *epochs* sont divisés en intervalles plus courts appelés slots de communication. Le nombre de ces slots est égal à la profondeur maximum de l'arbre de routage et l'agrégation de données est exécutée par tous les nœuds intermédiaires.

Exemple de requête utilisée dans TAG :

SELECT AVG(temperature),chambre FROM capteurs
WHERE etage = 10
GROUP BY chambre
HAVING AVG(temperature) > 25
EPOCH DURATION 60s

Cette requête partitionne les capteurs se trouvant au dixième étage d'un immeuble suivant les chambres dans lesquelles ils se trouvent. La requête renvoie toutes les chambres dans lesquelles la température moyenne est supérieure à 25 unités de température. Les résultats mis à jour sont envoyés après chaque minute (60secondes).

2. Directed Diffusion

Dans Directed Diffusion, l'agrégation de données est exécutée, quand les données sont envoyées au puits, au moyen des méthodes convenables, qui peuvent être choisies selon les exigences de l'application. L'arbre de collecte de données (renforcement des chemins) doit être périodiquement rafraîchi par le puits. Cela peut être cher en cas de topologies dynamiques.

3. PEGASIS

Rappelons que dans PEGASIS les nœuds sont organisés en chaînes avec un chef (leader) par chacune et qu'il est le seul nœud autorisé, parmi tous les nœuds qui forment la chaîne dont il est leader, à transmettre directement au puits. L'agrégation dans PEGASIS se fait selon la procédure suivant :

Chaque nœud reçoit la donnée de son voisin (son prédécesseur dans la chaîne), l'agrège avec sa propre donnée en générant un seul paquet de même taille et envoie le résultat à son successeur. Ce processus continue jusqu'à ce que le paquet atteigne le leader de la chaine courante .A ce stade, le leader inclut sa propre donnée dans le paquet et l'envoie au puits.

IV.5.2.2 Approche basée sur les clusters

Dans les protocoles bases sur les cluster tels que LEACH et COUGAR, vus dans le deuxième chapitre, l'agrégation est exécutée par les clusterheads. Le cluster-head collecte les données des membres de son cluster, les agrèges avec sa propre donnée et envoie le résultat au puits (si il se trouve dans la portée du puits) ou transmet le résultat à un autre cluster. Le processus est donc semblable à celui vu dans l'algorithme PEGASIS, en comparant le cluster-head au chef de la chaîne et les membres du cluster aux nœuds appartenant à la même chaîne.

IV.5.2.3 Approche multi-chemins

L'idée principale de cette approche est que chaque nœud peut envoyer à ses (si possible) multiples voisins en exploitant la nature broadcast du média sans fil. Une structure qui convient bien à cette approche s'appelle topologie en anneaux où les nœuds capteurs sont divisés en plusieurs niveaux selon le nombre de sauts les séparant du puits. L'agrégation de données est exécutée sur les chemins multiples puisque les paquets se déplacent niveau par le niveau vers le puits (voir figure IV.2).

Dans ce qui suit, nous allons voir Synopsis Diffusion qui appartient à cette classe de protocoles.

Dans **Synopsis Diffusion** [77] la topologie de diffusion de données est organisée en anneaux concentriques autour du puits. Synopsis Diffusion comprend deux phases : la phase de distribution des requêtes(distribution of the queries) et la phase d'extraction de données(data retrieval). La topologie en anneau est formée quand un nœud envoie une requête dans le réseau. En particulier, deux différentes structures, énumérées ci-dessous, peuvent être prises en compte. Le premier type de topologie consiste en structure d'anneau simple.
Pendant la phase de distribution des requêtes, les nœuds du réseau forment un ensemble d'anneaux autour du nœud émetteur de la requête *q*, qui est le seul capteur appartenant à l'anneau R_0. Un nœud est dans l'anneau R_i s'il est à i sauts du nœud q. Le deuxième type de topologie a quelques améliorations qui le rendent plus robuste que le précédent et capable de faire face aux changements dans le réseau. Cette topologie s'appelle *anneaux adaptatifs* (adaptative rings). La phase de distribution ne change pas mais cette fois ci un nœud *n* dans l'anneau *i* garde la trace du nombre de fois, n_{ov}, que les transmissions de n'importe quel nœud n_{i-1} dans l'anneau *i-1* ont inclus ses propres données pendant quelques derniers epochs. Si le novembre est petit, *n* essaie de trouver un meilleur anneau pour avoir plus de ses propres données incluses dans les transmissions ultérieures.

Dans l'exemple de la figure IV.2, les données générées au nœud A peuvent atteindre le puits par sept chemins :{A, B, F, I, S}, {A, B, F, H, S}, {A, B, F, G, H, S}, {A, C, D, E, I, S}, {A, C, F, H, S}, {A, C, F, I, S} et {A, C, G, H, S}.

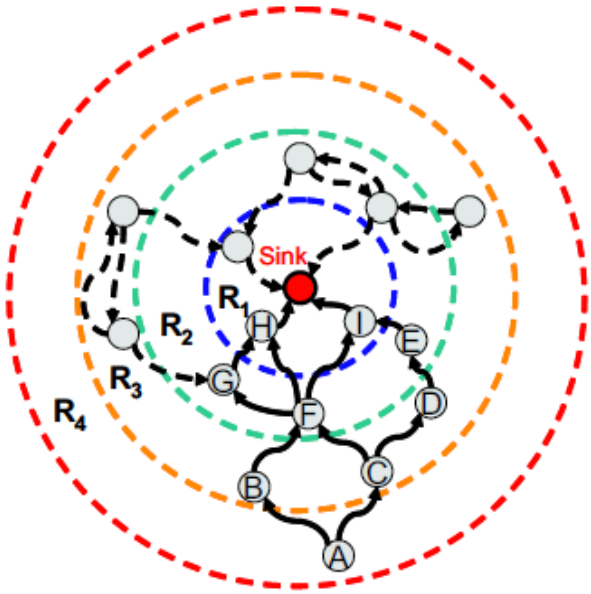

Figure IV. 2 Exemple d'une technique d'agrégation utilisant une structure en anneau

Notons que comme la caractéristique principale de Synopsis Diffusion est que les données peuvent être transmises via les multiples chemins, un nœud peut recevoir des copies de la même information. Cela peut affecter le résultat d'agrégation, surtout quand les fonctions d'agrégation sont sensibles à la duplication. Ce problème est adressé par les auteurs dans [77] en proposant des fonctions et des structures de données convenables, qu'on verra dans la suite de ce chapitre.

IV.5.2.4 Approche hybride

Pour profiter des avantages de deux approches (hiérarchique et multi-chemins), il est possible de définir des approches hybrides. L'exemple typique est présenté dans [78] et est expliqué ci-après.

Dans le protocole **Tributaries and Deltas [78]** les structures d'agrégation des données peuvent circuler simultanément dans de différentes régions du réseau. L'idée consiste en ce que sous les taux faibles de perte de paquets, un arbre d'agrégation de données est la structure la plus convenable. D'autre part, quand ces taux deviennent, une approche multi-chemins peut être la meilleure. Pour cela, les nœuds sont divisés en deux catégories : les nœuds utilisant l'approche hiérarchique pour envoyer les paquets (appelés aussi les nœuds T) et les ceux utilisant l'approche multi-chemins (Les nœuds M). Cela signifie que le réseau est organisé dans les régions en exécutant l'une des deux approches. La difficulté principale est

de relier les régions exécutant de différentes structures d'agrégation de données. De cette manière, les règles suivantes doivent être satisfaites [78] :

- ✓ *Exactitude d'extremité(Edge Correctness) :* Une extremité du nœud M ne peut jamais être l'incident à un noeud T. Cela signifie que le résultat d'agrégation dans une région multi-chemins peut seulement être reçu par un nœud M.
- ✓ *Exactitude de chemin (Path Correctness*): les nœuds M forment un sous-graphe incluant le puits qui est formé par les arbres composés des nœuds T.

Selon ces règles, le puits est entouré seulement par les nœuds M. Ceux-ci forment une région appelée delta qui peut être étendue ou rétrécie en échangeant les nœuds T en nœuds M et vice-versa, respectivement. Dans la pratique, seuls les nœuds qui se trouvent le long de la limite entre les deux régions changent leur mode d'opération comme le montre la figure IV.3 ci-dessous.

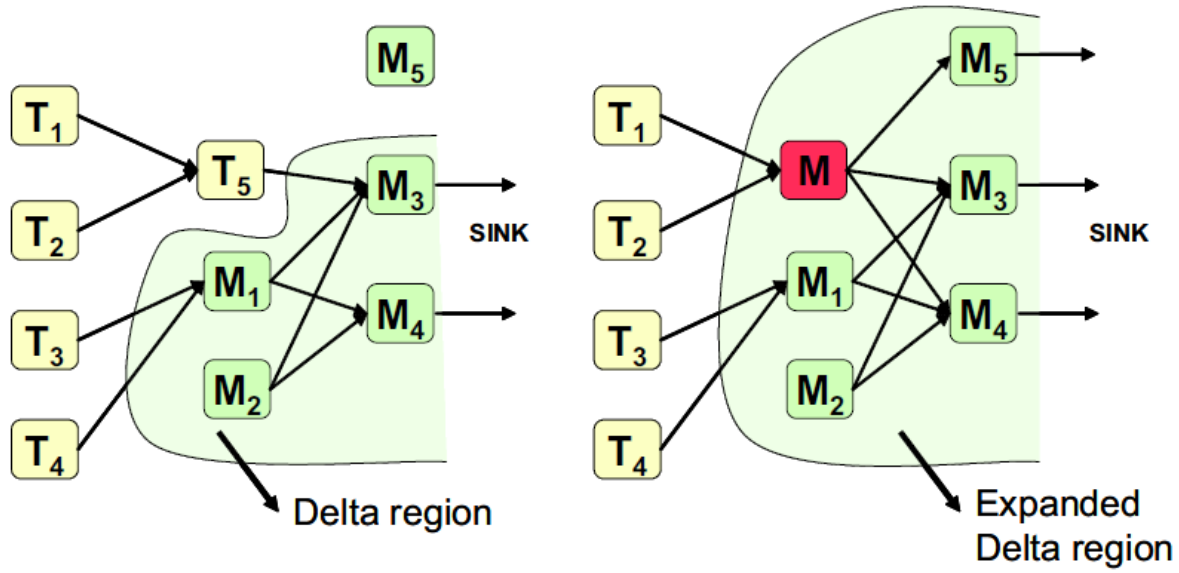

Figure IV. 3 Exemple des régions de collecte des données dans Tributaries and Deltas.

IV.5.3 La représentation des données

En raison de ses capacités de stockage limitées, un nœud peut ne pas être capable de conserver tous les paquets reçus et générés dans sa mémoire tampon. Il a besoin de décider donc s'il faut les conserver, les supprimer, les compresser, ou les transmettre. Toutes ces opérations exigent une façon convenable de représenter les données. La structure de données correspondante peut varier selon les exigences de l'application, du nœud ou même de la position du nœud. Ces structures exigent aussi qu'il y ait des fonctions d'agrégation capables de les exploiter. En effet, les fonctions d'agrégation présentées dans les paragraphes précédents comme MAX,MIN,AVG ont un avantage d'être simples et de réduire considérablement la quantité de données à transmette mais ne donnent aucune précision sur les données agrégées(ex :fonctions avec perte).Actuellement, beaucoup de travaux de recherche ont proposé des modèles de représentation de données dans les réseaux de capteurs *[77] [79] [80] [81]* .Nous avons choisi de détailler *Synopsis Diffusion Framework* proposé dans [77].

Comme nous l'avons vu dans la section IV.5.2.3 de ce chapitre, le protocole Synopsis Diffusion [77] se base sur une approche multi-chemins. La présence des données dupliquées au niveau d'un nœud du réseau faussera donc le résultat de l'agrégation si la fonction d'agrégation utilisée est sensible à la duplication. Les auteurs proposent un ensemble de structure de données et des fonctions d'agrégations pour résoudre ce problème. L'ensemble ainsi formé s'appelle *Synopsis Diffusion Framework* [77].

Un Synopsis est défini comme un résumé du résultat partiel du processus d'agrégation global reçu à un nœud donné. Trois fonctions sur les synopsis sont possibles pour exécuter l'agrégation de données :

- ✓ *Génération d'un synopsis(Synopsis Generation):*Etant donné une lecture d'un capteur, une fonction de génération d'un synopsis SG (.) produit le synopsis correspondant à cette lecture.
- ✓ *Fusion des synopsis (Synopsis Fusion) :* Etant donné deux synopsis, une fonction de fusion des synopsis SF (.; .) produit un nouveau synopsis qui résume tous les deux.
- ✓ *Evaluation d'un synopsis (Synopsis Evaluation) :* Étant donné synopsis, une fonction d'évaluation d'un synopsis SE (.) donne le résultat final.

Les implémentations des fonctions exactes et des définitions des synopsis dépendent de la requête d'agrégation considérée. Une façon de vérifier si l'algorithme Synopsis Diffusion est ODI-correct[1] est basée sur les quatre propriétés suivantes :

- ✓ Préserve les doublons : si deux lectures contiennent les mêmes valeurs de données, l'algorithme génère le même synopsis.
- ✓ La fonction de synopsis SF (.) est commutative : pour tout synopsis s1 et s2 nous avons SF (s1; s2) = SF (s2; s1).
- ✓ La fonction de synopsis SF (.) est associative : pour tout triplet (s1; s2; s3) nous avons
- ✓ SF (s1; SF (s2; s3)) = SF (SF (s1; s2); s3).
- ✓ La fonction de synopsis SF (.) est same-synopsis idempotent : pour tout synopsis s nous avons SF (s; s) = s.

Les quatre propriétés ci-dessus sont nécessaires et suffisantes pour l'exactitude d'ODI. Plus de propriétés et exemples peuvent être trouvés dans [77].

IV.6 Conclusion

Dans ce chapitre, nous avons étudié les protocoles d'agrégation des données utilisés dans les réseaux de capteurs sans fil. Nous avons constaté que ces protocoles utilisent les fonctions variées allant des fonctions de calcul de la moyenne des données collectées à des fonctions utilisant les requêtes semblables à celles des langages d'interrogation des bases de données.

Dans les applications où l'utilisateur a besoin d'avoir l'information captée par chaque capteur, afin par exemple d'intervenir, les fonctions que nous avons étudiées dans ce chapitre s'avèrent inadaptées. En effet, les fonctions telles que MIN et MAX ne permettent d'avoir qu'une seule donnée à partir des données captées par l'ensemble des nœuds capteurs. L'utilisation des fonctions d'agrégation basées sur les langages d'interrogation des bases de données comme le SQL, peut être couteuse en terme d'exécution et d'espace mémoire utilisé, car dans ce cas la communication doit se faire dans les deux sens (requete-réponse).Il convient donc dans ce genre de situation d'avoir recours à des fonctions utilisant les la corrélation entre les données.

[1] Une propriété ODI (Order and Duplicate Insensitive) est une propriété permet d'assurer que le résultat de l'agrégation est indépendant de la topologie ».Voir [67] pour plus d'information sur cette propriété

Les fonctions d'agrégation de données ont pour but de réduire le nombre de messages transmis dans les RCSFs. Ces messages peuvent causer la congestion et donc la technique d'agrégation des données peut être utilisée pour contrôler la congestion.

Dans la suite de ce travail nous proposons un protocole de routage utilisant la technique d'agrégation des données pour contrôler congestion dans les RCSFs en se basant sur la corrélation entre les données collectées.

Chapitre V

V.1 Introduction

Dans le troisième chapitre de ce travail, nous avons présenté et analysé les protocoles de contrôle de congestion proposés dans la littérature. Nous avons remarqué que ces protocoles peuvent ne pas être efficaces dans les applications où il y a beaucoup de redondances des données. Pour remédier à ce problème nous avons vu qu'il est possible d'utiliser les techniques d'agrégation des données. Cela nous a amené encore à étudier les techniques d'agrégation des données dans le quatrième chapitre.

A travers ces deux études nous avons remarqué que, à notre connaissance, il n'existe pas de protocoles de contrôle de congestion utilisant la technique d'agrégation des données ; ce qui nous a conduit à proposer un nouveau protocole de contrôle de congestion utilisant la technique d'agrégation des données.

Dans ce chapitre nous allons présenter et évaluer le protocole proposé. Pour ce faire, nous organisons ce chapitre selon un plan méthodologique suivant :

Premièrement nous fixons le cadre dans lequel nous allons évoluer à travers un certain nombre d'hypothèses. Ensuite nous décrivons notre protocole à travers ses phases de communications. Pour évaluer le protocole proposé il nous faut un outil de simulation des réseaux de capteurs, nous aurons donc à choisir et décrire l'outil de simulation utilisé. Après cela nous évaluerons notre protocole et nous finirons ce chapitre par une conclusion.

V.2 Hypothèses de travail

Afin de déterminer le travail que nous avons à faire, il est important de fixer le cadre dans lequel nous allons évoluer. Nous avons donc posé certaines hypothèses de travail suivantes :

H1. Nous traitons des réseaux de capteurs sans fil susceptibles d'être congéstés et dans lesquels l'utilisateur final a besoin de connaitre la donnée captée par chaque capteur individuellement.

H2. Un nœud n'envoie la donnée captée que si cette dernière diffère d'un certain seuil de la donnée précédemment envoyée.

H3. Les capteurs du réseau sont déployés sur une zone d'intérêt de forme carrée ou rectangulaire (de dimension L x l). Les capteurs peuvent être déployés aléatoirement ou placés manuellement dans des endroits stratégiques.

H4. Une fois déployés, les capteurs peuvent être mobiles ou stationnaires et sont capables de déterminer leurs positions géographiques grâce à l'utilisation d'un GPS ou en utilisant un système de localisation virtuel comme par exemple la triangulation [32].Dans le cas ou les capteurs ne sont pas capables de connaître leurs positions par l'une de ces méthodes, ils seront positionnées manuellement et l'utilisateur est capable de les. La connaissance des positions nous permettra de répondre à l'hypothèse H1.

H5. Nous supposons que l'ensemble des nœuds du réseau sont adressables. C'est-à-dire qu'on pourra établir une communication directe entre deux nœuds. Cela évitera d'avoir beaucoup de redondances de paquets à agréger et donc évitera d'avantage la congestion à certains niveaux du chemin de routage.

En nous basant sur ces hypothèses, nous proposons donc un nouveau protocole de routage utilisant la technique d'agrégation des données pour contrôler la congestion dans un réseau de capteurs sans fil. Nous commençons par sa description.

V.3 Description générale du protocole proposé.

Le but d'un protocole de communication est de permettre l'échange d'information entre les entités communicantes en assurant certaines métriques de qualité de service. Le protocole que nous proposons est un protocole ayant comme métrique de QoS principale de contrôler la congestion dans le réseau de capteurs dans lesquels il est implémenté.

Comme dans tout autre protocole de communication, les entités du réseau communiquent par l'échange de messages appelés les paquets. Le nombre de paquets varie selon le protocole et ont, chacun, un format bien déterminé.

Le protocole proposé utilise une technique d'agrégation de données pour réduire le nombre de paquets transmis au nœud puits dans le but d'éviter la congestion qui peut être causée par la présence des données redondantes dans le réseau. Cela veut dire quand un capteur recueille une donnée à partir de son unité de captage, il ne l'envoie pas directement mais il attend l'arrivée des données venant d'autres capteurs dont il assure le rôle de routeur, les agrègent avec sa propre donnée et envoie le résultat d'agrégation au nœud suivant dans le chemin de routage. Ce processus continue jusqu'à ce que les données atteignent le nœud puits. Au moment de cette attente, les données arrivées précédemment doivent être stockées dans une unité de stockage du capteur.

Les entités du réseau sont organisées dans une topologie arborescente dont le puits est la racine (voir figure V.1). Dans cette architecture, l'arbre de routage est construit à partir du nœud puits jusqu'aux nœuds extrêmes considérés comme les feuilles de cet l'arbre .Les nœuds intermédiaires entre le puits et les nœuds extrêmes jouent le rôle d'agrégateurs des paquets de données venant des nœuds dont ils sont parents.

La communication entre les entités du réseau s'effectue en différentes phases synchronisées au niveau de chaque nœud et la durée de chaque phase dépend de son importance et exigences du protocole.

Dans les paragraphes suivants, nous détaillons les types des paquets définis par notre protocole, ainsi que ses différentes phases de communication.

V.4 Les phases de communication

V.4.1 Phase d'initialisation

La phase d'initialisation est la première phase de notre protocole. Elle consiste à construire le chemin de routage. Pour minimiser le nombre de sauts qu'un paquet doit effectuer, et donc le nombre d'agrégation que peut subir ce paquet, nous adoptons un chemin de routage arborescent dont le puits est la racine. Pour cela, chaque nœud doit garder deux paramètres suivant :

a) *a)Sa profondeur (scrDepth)*:le nombre de saut qu'un paquet généré par ce nœud devra effectuer avant d'atteindre le nœud puits.

b) *b) L'identifiant de son nœud parent(SrcNodeId)* : pour éviter le routage de données en mode diffusion, chaque nœud possède un identifiant (son adresse)(Hypothèse H5).Cet adresse peut être l'adresse MAC du capteur ou tout autre type d'adresse reconnu par les capteurs du réseau pourvu que chaque nœud possède son propre adresse.

Après le déploiement des capteurs, le puits initie cette phase en émettant un paquet d'initialisation (voir tableau V.1) avec le champ *profondeur* à zéro. A la réception du premier paquet d'initialisation, un nœud initialise un compteur(Timer) ,*initTimer*, de durée aléatoire mais comprise entre deux valeurs(*initMinDuration* et *initMaxDuration*).Avant l'expiration de ce timer, tous les paquets reçus seront stockés dans la mémoire du nœud récepteur. Il est à noter que, pour ne pas dépasser la capacité mémoire d'un capteur, le nombre de messages d'initialisation est limité.

Quand un nœud reçoit un message d'initialisation, il vérifie si le nombre maximum n'est pas atteint. Si ce nombre est atteint, il supprimer le timer et choisit le nœud parent. Si non le paquet est stocké. Le choix du nœud pare s'effectue comme suit :

Un nœud choisit l'identifiant du nœud ayant une plus petite profondeur comme l'identifiant du nœud parent et met sa profondeur à la profondeur du parent plus un. Ensuite il diffuse un paquet d'initialisation contenant sa profondeur et son identifiant. Ce processus

continue jusqu'à ce que tous les nœuds qui peuvent être atteints à partir du puits aient leurs parents.

A partir du moment où un nœud possède l'identifiant de son nœud parent, il peut commencer à envoyer les données car il existe un chemin le reliant au nœud puits. La topologie définie établie dans cette phase d'initialisation n'est pas statique. Elle pourra changer tout au long de la durée de vie du réseau pendant la phase de réaffiliation qu'on va décrire dans la suite de cette section.

Tableau V. 1 Les Champs du paquet d'initialisation

Champs	Description
srcNodeId	Identifiant du nœud émetteur du paquet
scrDepth	La profondeur du nœud émetteur du paquet

La figure suivante est un diagramme de transition d'états résumant la phase d'initialisation au niveau d'un nœud capteur.

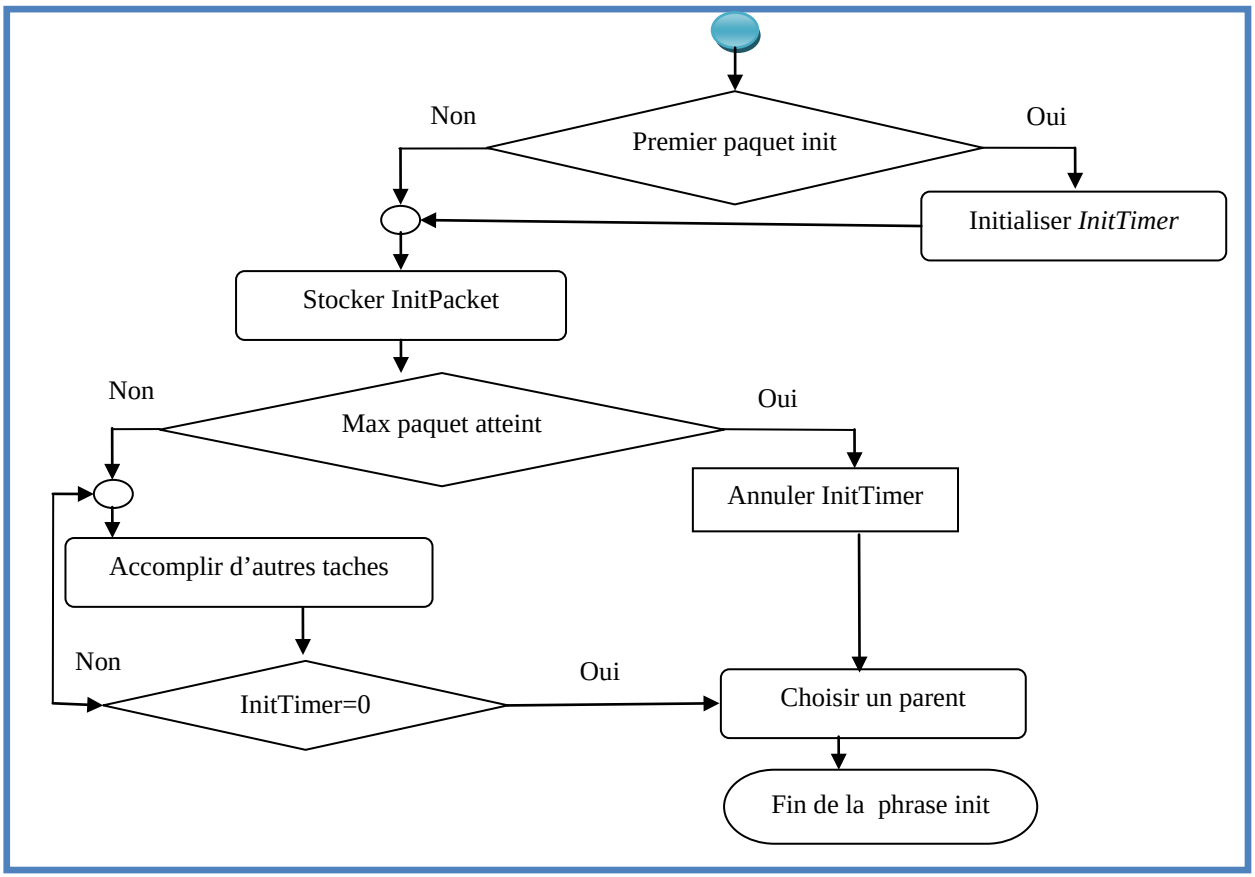

Figure V. 1. Diagramme de transition d'états de la phase d'initialisation

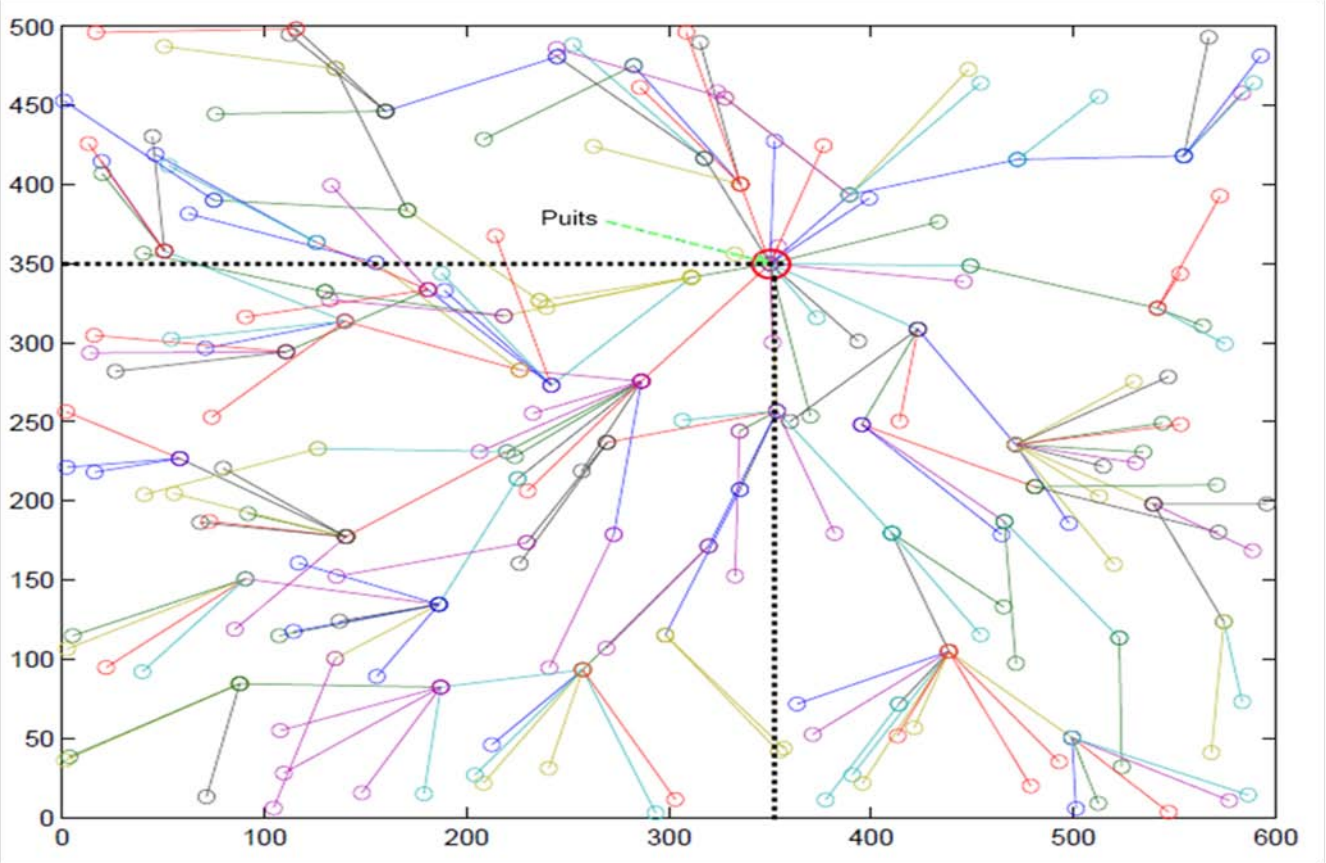

Figure V. 2. Exemple de la topologie hiérarchique du protocole proposé

V.4.2 Phase de collecte des données

La collecte des données s e fait au niveau du nœud puis et au niveau des capteurs

1. au niveau d'un nœud capteur

Nous distinguons deux types de collecte des données au niveau de chaque nœud capteur : la collecte des données venant de son unité de captage et celles en provenance des nœuds capteurs qui sont ses fils dans l'arbre de routage.

Pour accomplir la phase de collecte des données, chaque nœud capteur maintient une structure de données(buffer) de taille fixe[1] permettant de stocker les paquets de données (voir tableau)envoyés par ses nœuds fils pendant cette phase et une variable permettant de stocker la dernière donnée reçue de son unité de captage. Lorsque la phase de collecte commence, un

[1] La taille de ce buffer dépend de l'espace mémoire dont dispose un capteur et de la taille des paquets échangés.

capteur initialise un timer, a*ggregWaitingTimer*. A chaque réception d'une donnée, le capteur calcule le taux d'occupation du buffer. Si ce taux dépasse un certain seuil ou bien atteint son niveau maximal, un capteur arrête le timer et commence la phase d'agrégation. Si non il stocke la donnée reçue et la phase continue. A l'expiration du timer, le capteur commence la phase d'agrégation. Le format d'un paquet de données est décrit dans le tableau ci-dessous.

Tableau V. 2. Les Champs du paquet de données

Champs	Description
DestNid	L'identifiant du nœud destinateur (parent du nœud émetteur)
Data	Donnée du phénomène physique étudié
SrcIds	Une structure de données permettant d'indentifier les nœuds sources
appSpecific	Une structure de données relatives à l'application cible

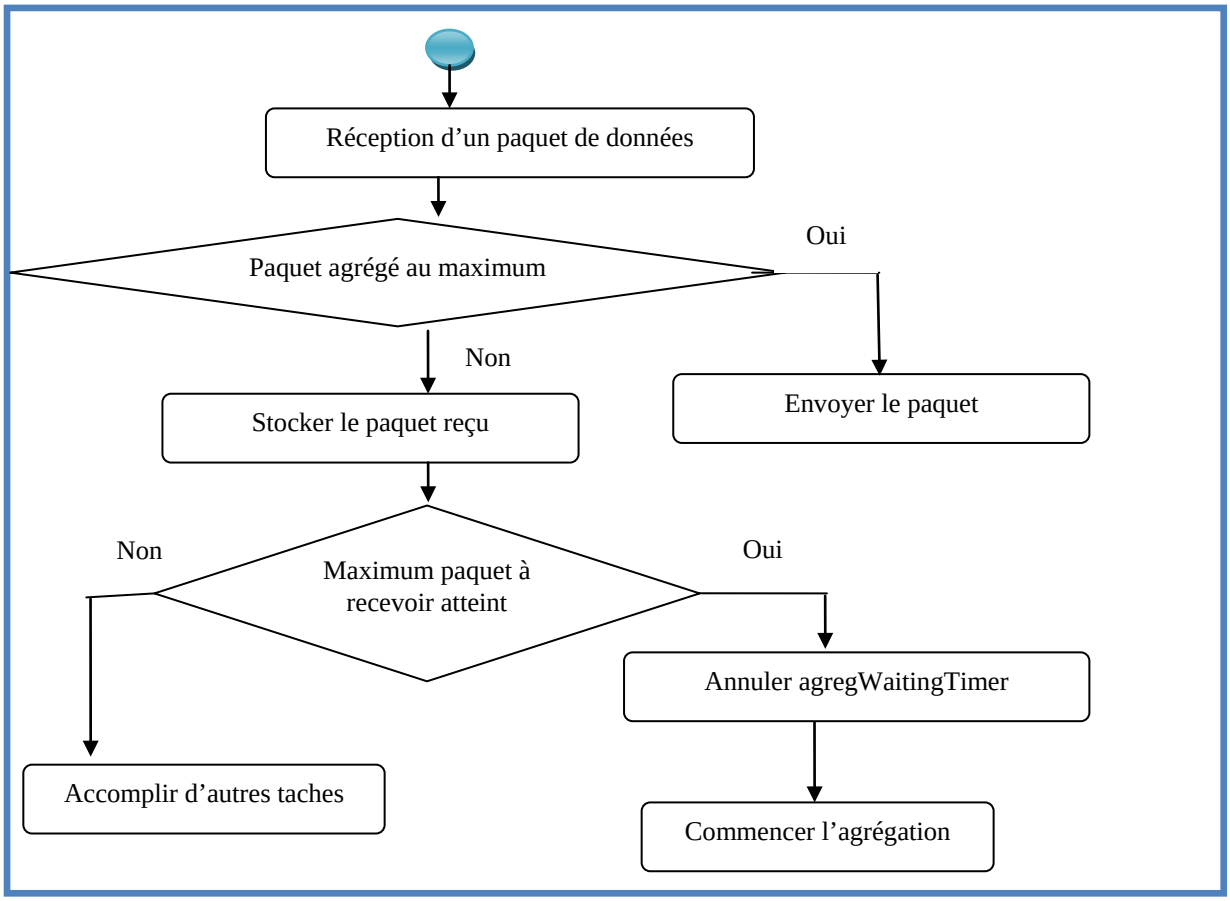

Figure V. 3. Diagramme de transition d'états de la phase de collecte des données.

2. au niveau du nœud puits

La phase de collecte des données au niveau du nœud puits est très différente de celle qui s'effectue au niveau des nœuds capteurs. Rappelons que suivant l'architecture d'un réseau de capteur (voir Chapitre I), le nœud puits est connecté à l'ordinateur central de l'utilisateur final. Le puits est un capteur et donc même si nous supposons qu'il possède des ressources suffisantes, ses capacités d'affichage ou d'interprétation des données sont à elles limitées.

En effet, il serait important, pour un utilisateur final, de garder une trace de cette collecte de telle sorte qu'il puisse avoir l'information sur la donnée collectée à chaque moment et par quels nœuds sources ces données ont été collectées. Pour cela nous proposons deux solutions :

a) Après avoir reçu un paquet, le puits le transmet à un ordinateur central et ce dernier stocke l'information utile à l'utilisateur dans un fichier de trace qui est analysé selon les besoins de l'utilisateur est le résultat sera affiché dans une interface graphique ou sur un invité de commande.
b) Le fichier de trace peut être remplacé par une base de données qui sera interrogée selon les besoins de l'utilisateur.

Les fichiers de trace ou la base de données permettront à l'utilisateur de prendre quelques mesures d'intervention. Par exemple soit à une défaillance importante d'un certain nombre de nœuds capteurs, il pourra procéder à un redéploiement d'autres capteurs.

V.4.3 Phase d'agrégation des données

Pendant la phase d'agrégation des données, la fonction d'agrégation des données est appelée .A partir des données stockées dans le buffer et la donnée issue de l'unité de captage, cette fonction se base sur la corrélation entre ces données pour former les paquets à envoyer au prochain saut. Pour cela, nous définissons une variable *seuil de sensibilité* (dataSensibility). Tous les paquets de données dont le champ *data* diffèrent d'une valeur inférieure ou égale à *dataSensibility* seront agrégés en un seul paquet dont le champ data est la moyenne arithmétique des données agrégées. Le nombre de paquets à envoyer après l'agrégation dépendra de la corrélation entre les données reçues.

V. 4.4 Phase de contrôle de congestion

Dans le chapitre 3 de ce travail, nous avons distingué deux types de congestions à savoir la congestion au niveau du nœud et la congestion au niveau du canal sans fil. Pour contrôler la congestion globalement, nous effectuons le contrôle sur ces deux types de congestion.

a) *Contrôle de congestion au niveau du nœud* **:** Ce type de contrôle est effectué chaque fois qu'un nœud reçoit un paquet de données. Rappelons que le buffer de réception des données a une taille limitée, *bufferSize*, égale au nombre maximum des nœuds dont les paquets de données peuvent être stockés avant qu'un nœud déclare un état de congestion. *BufferSize* est configurable et dépend de l'espace mémoire dont dispose un capteur. Pour vérifier le niveau d'occupation du buffer de données, un nœud capteur utilise un compteur, *bufferOccupied*.A la réception des paquets de données, un nœud capteur vérifie le niveau de congestion comme suit :

Il ajoute à *bufferOccupied* la taille du paquet reçu. Si le pourcentage d'occupation du buffer dépasse un certain seuil, le nœud déclare une congestion à ces nœuds fils et commence l'agrégation. Si non il sauvegarde le paquet dans le buffer.

b) *Congestion au niveau du canal de transmission :* Pour contrôler la congestion au niveau du canal de transmission, la couche application interroge la couche MAC ainsi que la couche Physique pour connaitre l'état du canal sans fil et celui de la couche MAC avant d'envoyer les données à la couche basse. Cela est dû au fait que dans un protocole de contrôle de congestion, les couches hautes (application, réseau) peuvent envoyer les paquets à la couche MAC alors que le canal sans fil n'est pas libre. Dans ce cas il y'aura les collisions (congestion du lien) qui provoqueront la perte de ces paquets. Nous voulons donc minimiser ou éviter cette perte.

Lorsqu'un nœud est congesté, il émet un paquet de notification de congestion (*congNotifPkt*) à ces nœuds fils. Le tableau suivant décrit le format de ce paquet.

Tableau V. 3. Les champs du paquet de notification de congestion

Champs	Description
srcNid	**L'identifiant du nœud émetteur (nœud congesté)**

V.4.5 Phase de réafiliation

Durant cette phase, il y a un changement de la topologie du réseau. Ce changement survient suite à l'apparition d'un des événements suivants :

a) *Détection de la congestion au niveau d'un nœud* **:** Contrairement aux protocoles qui ordonnent aux nœuds sources de messages de réduire leurs taux de transmissions quand un une congestion est détectée, nous voulons que lorsqu'une congestion est détectée, un nœud congesté déclare son état de congestion à ses nœuds amont et que dans le cas où cette congestion est persistante, les nœuds amont puissent changer de nœud parent. Cela évitera le retard et la perte des paquets dus à cette baisse de taux de transmission.

b) un certain niveau d'énergie au niveau du nœud parent : L'énergie est une ressource critique dans un réseau de capteurs. Pour équilibrer l'usage d'énergie dans un réseau nous définissons trois niveaux d'énergie :

i) *Niveau normal* : lorsqu' un nœud capteur possède un niveau d'énergie supérieur au niveau normal, il peut effectuer toutes les opérations de réception et transmission sans se soucier de son niveau d'énergie.

ii) *Niveau moyen* : C'est le deuxième niveau d'énergie. Un nœud qui atteint ce niveau d'énergie, le déclare à ces nœuds amonts en leur transmettant un paquet d'alerte du niveau d'énergie (Tableau V.4).Ces derniers peuvent changer leurs parents après la réception de ce paquet.

iii) *niveau bas* : Quand un nœud atteint ce niveau d'énergie, il ne peut plus recevoir les paquets. L'énergie restante lui permettra de transmettre les paquets déjà reçus et les données qu'il va capter jusqu'à l'épuisement de son énergie.

c) *Déploiement d'un nouveau nœud* : Suite à la panne d'un nœud capteur ou de l'existence des zones non couvertes, le déploiement d'autres nœuds peut être envisageable. Ce déploiement doit se faire sans réinitialiser le réseau. Chaque nouveau capteur envoie un paquet de recherche d'un nœud parent (*parentSearchPkt*) (Tableau V.5) et les autres nœuds peuvent le choisir comme parent.

d) *panne physique d'un nœud capteur :* En plus de la panne causée par l'épuisement de l'énergie d'un capteur, un nœud peut être défaillant pour d'autres raisons (voir chapitre I).Quand un nœud tombe en panne, les nœuds capteurs qui étaient ses fils dans l'arbre de routage doivent trouver un autre chemin. Pour assurer cela, les nœuds transmettent périodiquement les paquets *hello* (Tableau V.5) à leurs nœuds fils. Après une certaine période qu'un nœud n'a pas reçu un paquet hello de son parent, il le considère comme défaillant et cherche un autre chemin de routage.

Avant la phase de réaffiliation, un nœud capteur émet un paquet de recherche d'un parent*(parentSearchPkt)* ses nœuds voisins. Quand un noud reçoit ce paquet, il peut répondre par un paquet d'acceptation (*parentSearchReplayPkt*).A la réception de ces derniers paquets, un nœud émetteur du *parentSearchPkt* choisit le meilleur parent selon la même procédure que celle décrite dans la phase d'initialisation.

Tableau V. 4. Les champs du paquet d'alerte du niveau d'énergie

Champs	Description
srcNodeId	L'identifiant du nœud émetteur du paquet
nrgValue	Niveau d'énergie restant

Tableau V. 5. Les champs du paquet de recherche d'un nouveau parent

Champs	Description
searcherNid	L'identifiant du nœud émetteur du paquet
reason	La raison de recherche du nœud parent

Le champ *reason* peut prendre l'une des trois valeurs suivantes :

1: Si le parent du nœud émetteur est contesté

2 : Si le parent du nœud émetteur est en manque d'énergie

3 : Si l'émetteur du paquet est un nœud qui vient d'être déployé (phase de redéploiement) ou qui vient d'apparaitre dans une zone (mobilité).

Tableau V. 6. Les champs du paquet de réponse à la recherche d'un nouveau parent

Champs	Description
replayerDepth	La profondeur du nœud répondeur
replayerNid	L'identifiant du nœud répondeur
reason	La raison de recherche d'un nœud parent
searcherNid	L'identifiant du nœud émetteur de la requête de recherche d'un parent

Tableau V. 7. Les champs du paquet *Hello*

Champs	Description
srcNid	L'identifiant du nœud émetteur du paquet

V.5 Implémentation et simulation du protocole proposé

Après l'étape de description de notre protocole, on passe à l'étape d'implémentation. Cette étape consiste à traduire les phases de communication précédemment définies en code source d'un langage de programmation cible. Pour cela, nous avons besoin d'un environnement de développement dont le choix dépendra du langage de programmation choisi et de la finalité de ce code source. La finalité du code source peut être son chargement sur le matériel réel(les capteurs) ou pour des faits de simulation du comportement du protocole et de mesure de sa performance. Pour ce qui est de notre cas, nous envisageons tout d'abord la simulation.

Avant sa mise en place, le déploiement d'un réseau de capteurs nécessite une phase de test afin de s'assurer le bon fonctionnement de tous ses composants; matériel ou logiciel. Pour se faire, il existe à ce jour trois grandes solutions [82]: le test en environnement réel, la simulation et l'émulation.

1 .Test en environnement réel

La première solution consiste à utiliser un réseau de capteurs réel pour bien mener son évaluation. Bien qu'elle permette d'avoir des conditions parfaitement réalistes, cette solution n'est pas complètement satisfaisante car elle ne permet pas de contrôler l'ensemble des paramètres de l'environnement. De plus, cette solution ne permet pas de reproduire plusieurs fois de suite les mêmes conditions avec précision. En effet, les conditions de propagation du signal à l'intérieur du réseau ne sont pas contrôlables, si bien que d'une expérience à l'autre les conditions observées peuvent évoluer et parfois modifier considérablement le résultat des tests.

2. Simulation

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

3. L'émulation

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

Bien que l'émulation paraisse plus intéressante et porte plus d'avantages, elle impose cependant certaines contraintes. Comme l'émulation travaille en temps réel, les modèles utilisés pour simuler les couches basses ne peuvent pas être trop complexes ce qui implique un impact négatif sur le réalisme de l'émulation rendue. A cet effet, l'approche de simulation détient toujours sa place comme solution pour le test et la validation de protocoles et d'applications [83].

V.5.1 Caractéristiques des environnements de simulation des RCSFs

Un simulateur réseau ou à événements discrets se caractérise par le fait que les changements d'états dans le réseau simulé (événements) se produisent à des instants répartis de manière discrète sur l'axe de temps. Classiquement, une simulation consiste à reproduire le comportement du système complet dans un environnement synthétique où le temps est simulé par une horloge à événements discrets. La simulation demande une modélisation complète, depuis les applications jusqu'au réseau physique, ce qui peut se révéler à la fois complexe et coûteux en terme de temps d'exécution et d'utilisation mémoire dans le cas de réseaux de taille importante. En général, un simulateur réseau doit avoir les caractéristiques suivantes [83]:

1. Génération de la topologie du réseau

Chaque simulateur a son mécanisme pour générer la topologie du réseau. Plusieurs possibilités existent: un script ou un langage de configuration spécifique au simulateur, des fichiers textes ou via des interfaces graphiques. Le simulateur doit aussi permettre la création de topologies hiérarchiques et la génération automatique de topologies aléatoires.

2. Génération et la gestion du trafic réseau

Afin de générer le trafic de données circulant sur l'environnement de simulation, il est nécessaire d'utiliser des générateurs de données suivant une distribution qui doit être la plus représentative du trafic réel et qui permettra d'établir les statistiques nécessaires pour tirer des conclusions.

3. Contrôle du réseau

Pendant l'exécution, il est très utile de contrôler dynamiquement le réseau soit par flux de données ou par nœuds via des interfaces graphiques. Les traces et les résultats du contrôle peuvent êtres sauvegardés dans des fichiers afin de faire des comparaisons ou ré exécuter la simulation avec un paramétrage plus affiné.

4. Modèle protocolaire, de mobilité et de propagation radio

Un bon simulateur dans son modèle protocolaire doit disposer de plusieurs protocoles dans les différentes couches de communication. Dans certains cas pour les RCSFs, nous avons besoin de gérer la mobilité des nœuds. Cela se fait généralement par l'intégration d'un modèle de mobilité dans l'environnement de simulation. De plus, le modèle de la propagation radio pour les réseaux sans fil a un grand impact sur les résultats de la simulation, donc le simulateur doit fournir plusieurs modèles de propagation radio qui doivent être les plus proches possible du cas réel.

5. Flexibilité et extensibilité

Le simulateur doit donner la possibilité de contrôler facilement les composants dans différents niveaux (changer leurs états par exemples). De plus, les modèles disponibles peuvent êtres remplacés, modifiés ou enrichis par l'ajout de nouveaux modèles de nouveaux modèles sans perturber le bon fonctionnement du simulateur. D'une manière générale, l'architecture du simulateur doit être modulaire et ouverte.

6. Temps d'exécution

Le temps d'exécution est un facteur qui doit être pris en considération dans le choix du simulateur et surtout s'il s'agit de simuler des réseaux de grandes tailles comme c'est le cas pour les réseaux de capteurs.

7. Ergonomie, visualisation des résultats et statistiques

L'environnement de simulation doit être le plus ergonomique que possible et doit faciliter les tâches de configuration et de suivie des différentes étapes de la simulation via des interfaces graphique. De plus, une représentation graphique des résultats et statistiques à travers des diagrammes, graphes ou sur des cartes géographiques est très importante afin de permettre une meilleure interprétation des résultats.

V.5.2 Le simulateur J-Sim

Plusieurs environnements de simulation sont utilisés pour évaluer les protocoles et des architectures proposés pour les WSNs. Certains sont libres et parfois Open source par contre il y en a d'autres qui sont commercialisés. Aucun de ces simulateurs n'est parfait ni ne répond à tous les besoins. Chacun d'eux présente des avantages et des inconvénients. Dans cette section, nous n'abordons le sujet d'étude comparative de ces outils car cela a fait l'objet de plusieurs travaux de recherche dont [84] et [85]. Nous allons décrire le simulateur *J-Sim* que nous avons choisi pour simuler notre protocole.

J-SIM J-Sim [86] (appelé autrefois JavaSim, ancien nom qui était en conflit avec la marque de Java Sun) a été développé par une équipe du laboratoire Distributed Realtime Computing Laboratory (DRCL) de l'université d'État d'Ohio. Ce simulateur,développé entièrement en langage Java, repose sur une architecture logicielle basée sur les composants autonomes, appelée « Autonomous Component Architecture » (ACA).

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

La simulation du fonctionnement d'un réseau de capteurs, qui exige la définition des composants et leur mise en relation, est réalisée grâce à un langage spécifique, TCL(Tool Command Line)[87]. Il s'agit d'un langage de script dans lequel on spécifie l'architecture du réseau ainsi que les paramètres de simulation et d'analyse. Les commandes de script peuvent également être fournies en ligne de commande, instruction par instruction.

A l'aide de TCL, on définit les composants puis on les connecte. Tous les composants sont hébergés dans un conteneur, qui est à son tour un composant. La définition des composants est en fait la création des objets. Cette création est réalisée par la commande TCL *mkdir*. Chaque composant est d'une entité indépendante (la couche MAC par exemple) qui fonctionne indépendamment des autres entités. Les composants possèdent des ports par défaut pour qu'ils puissent communiquer entre eux. D'autres ports peuvent être créés pour un composant. Une connexion entre deux composants est réalisée par l'intermédiaire de deux ports dédiés, un dans chaque composant (voir la Figure V.4). Cette connexion est matérialisée par la commande TCL *connect*. Il suffit de suivre un schéma qui indique l'interconnexion

entre les différents types de composants que l'on veut utiliser. On obtient ainsi l'architecture du nœud. Toujours dans ce script TCL, on y définit les paramètres globaux (par exemple la taille du champ de simulation), les outils de visualisation des résultats et l'ordonnancement de la simulation.

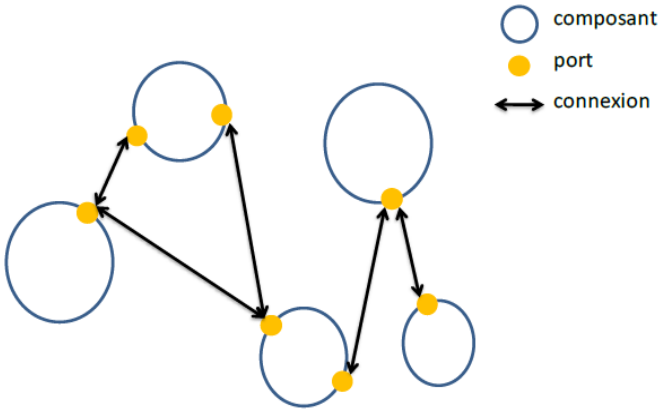

Figure V. 4. Connexions entre les composants dans J-Sim

On distingue trois types de nœuds dans J-Sim (voir figure V.5): *target*, *sink* et *sensor*. Les nœuds target ont pour fonction de générer des signaux (stimuli) qui peuvent être, par exemple, une secousse sismique, un bruit, etc. Le nœud sink est quant à lui le nœud cible, celui où les informations doivent arriver. Le reste du réseau est formé par les nœuds sensor qui captent le signal généré par target, forment les paquets contenant l'information de mesure et acheminent ces paquets jusqu'au nœud sink.

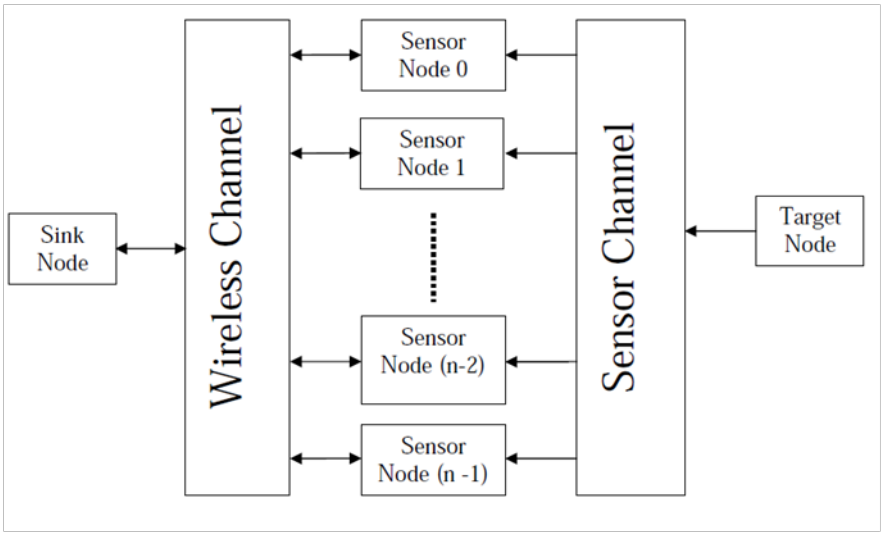

Figure V. 5 . Vue générale sur trois types de nœuds dans J-Sim[88]

La Figure V.6 (partie en pointillé) montre l'architecture interne d'un nœud sensor dans J-Sim. On y distingue notamment les composants CPU et Radio qui définissent les modèles de la consommation énergétique du microcontrôleur et de l'antenne radio. Le composant CPU décrit en principe le coût de traitement des instructions. Le composant Radio décrit le coût de l'envoi et la réception des signaux et la mise en veille. Le composant Battery définit le modèle de la batterie d'un nœud. Ce composant calcule la consommation énergétique due au traitement des instructions, envois et réceptions des signaux en se basant sur les modèles du CPU et de Radio. Un programmeur peut facilement créer ses propres modèles (CPU, Radio, Battery) et les insérer dans un scénario de simulation afin de les tester. A noter que seuls les nœuds sensor sont dotés d'une batterie.

La Figure V.6 montre aussi que les composants des différentes couches d'un nœud sont séparés (Network layer, Mac Layer, Physical Layer). Ils communiquent entre eux via des ports. Les connexions (désignées par des flèches) entre les composants peuvent être unidirectionnelles ou bidirectionnelles. Une connexion unidirectionnelle d'un composant C1 vers un composant C2 signifie que les informations peuvent être envoyées dans une seule direction (de C1 à C2). D'autre part, une connexion bidirectionnelle signifie un échange d'information dans les deux directions. Le composant Wireless Channel est le composant commun pour tous les nœuds sensor du réseau. Il fait communiquer tous les nœuds sensor entre eux. Chaque nœud sensor du réseau doit avoir une connexion, réalisée par le programmeur, avec ce composant. Le cas échéant, le nœud ne peut pas communiquer avec les autres nœuds. Le composant Sensor Channel est responsable de la communication entre les nœuds target et les nœuds sensor. De même, chaque nœud (sensor ou target) doit avoir une connexion avec ce composant. Si, par exemple, un tel événement est créé par un nœud target, ce composant envoie l'information à tous les nœuds sensor qui entourent le target (la distribution des nœuds et leurs positions dans le champ de simulation sont définies dans le fichier TCL).

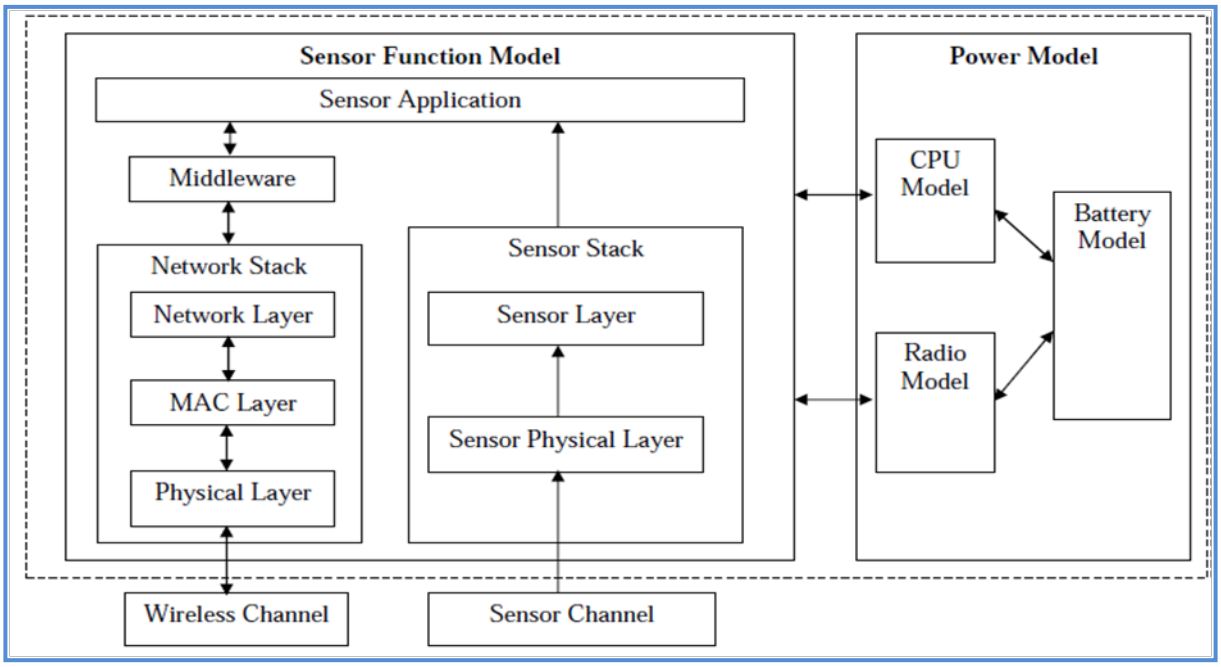

Figure V. 6. Architecture interne d'un nœud sensor dans J-Sim [88]

V.5.2.1 L'architecture détaillée de J-Sim

1. Le composant

L'entité de base dans l'architecture logicielle de J-Sim est le composant. Nous définissons une application en réalisant une composition de composants. Les composants dans J-Sim sont faiblement connectés (loosely coupled). Ils communiquent entre eux en connectant leurs ports ensemble, et sont liés par des contrats.

2. Le port

Un composant communique avec les autres composants par l'intermédiaire de ses ports. Il peut posséder plus d'un port. L'interface de programmation entre un composant et son port est bien définie. Un composant peut être développé sans la présence d'autres composants (il se doit d'être autonome). En outre, le mécanisme réel de communication qu'un composant emploie pour communiquer avec le reste du monde est complètement caché par la notion de port.

3. Le contrat

La communication entre composants est décrite par les contrats. Le contrat impose les conditions sur les ports d'entrée/sortie des composants afin de les faire communiquer. Un contrat indique comment un initiateur (visiteur) et un réacteur (appelé) accomplissent une certaine communication. Il décrit comment un composant répond aux données qui arrivent à chacun de ses ports (par exemple, comment le composant traite les données, certaines structures de données de mises à jour, et produit des sorties à certains ports). Les contrats indiquent la causalité des informations envoyées/reçues entre les composants, mais n'indiquent pas les composants qui participent à la communication.

4. Le langage utilisé pour définir un scénario de simulation

Bien que J-Sim soit écrit en Java, les scénarios de simulation ne sont pas décrits en Java (car ce processus serait trop complexe). La combinaison des langages Java et TCL ne facilite pas cette tâche. Pour développer un grand projet, il peut devenir encombrant d'employer des commandes TCL/Java parce que les références des objets Java doivent être stockées dans des variables TCL afin d'y accéder. Pour simplifier la syntaxe des simulations il existe un système appelé RUntime Virtual (RUV). Ce système s'appuie sur la similitude entre les systèmes composants et les systèmes de fichiers d'UNIX. L'analogie entre un composant/port et un chemin de fichier permet d'accéder au composant de la même manière que l'on accède à un dossier dans un système de fichiers. D'ailleurs des commandes systèmes UNIX, telles que ls, cd, pwd, mkdir, cp, mv, et rm peuvent être utilisées pour manoeuvrer des composants et des ports dans le système. Les ports ont une utilisation particulière pour le nommage : a@b (où a est le nom du port et b le groupe du port). En résumant, des composants peuvent être créés et ils peuvent être liés grâce à une interface de commande qui utilise la syntaxe des invités de commandes (shells).

5. Modèle de transmission dans J-Sim

Comme nous pouvons le voir sur la figure figure V. 7, le nœud target et les nœuds sensors communiquent vivant l'interface appelée sensor channel tandis que la communication entre les nœuds sensors et le sink est réalisée via l'interface Wireless channel. Ces deux interfaces possèdent chacune son propre modèle de transmission.

a) Modèle de transmission dans Sensor channel

La version de J-Sim que nous utilisons (V 1.3), la plus utilisée, implémente deux modèle de propagation du signal généré par le nœud target. Il s'agit du modèle sismique (seismic propagation model) et le modèle acoustique (acoustic propagation model).

Le modèle sismique calcule la puissance du signal reçu(Pr) par un nœud sensor suivant l'équation

$$P_r = \frac{P_t}{\max(d, d_0)^{fa}}$$

Avec P_t la puissance de transmission du signal, d la distance entre le target et le sensor,d_0 et fa(atenuation factor: facteurs d'atténuation du signal) sont des paramètres configurables du modèle de propagation sismique.

Dans le modèle acoustique, cette formule devient :

$$P_r = N(p * \mu_{g,\sigma_g^2})$$

$$p = \frac{P_t}{max\,(d,d_0)^{fa}} \quad , \mu = U(min_g, max_g)$$

min_g, max_g, μ_g et σ_g^2 sont respectivement le minimum, maximum, moyen et la variance du gain de microphone et sont configurables.

Notons que les concepteurs de nouveaux protocoles/applications peuvent à tout implémenter leurs propres modèles de transmission ou modifier ces deux modèles.

b) Modèle de transmission dans Wireless channel

Ce modèle de propagation divise le champ de simulation en plusieurs sous-champs à deux dimensions. Chaque sous-champ est un rectangle de taille dx*dy (voir la Figure V.7 côté gauche). Un nœud définit sa portée de transmission comme suit : il peut communiquer seulement avec les nœuds appartenant à son sous-champ et avec ceux dans les sous-champs voisins. La Figure V.7 (côté gauche) montre un nœud A et ses nœuds voisins qu'il peut

atteindre en un seul saut. Il peut communiquer avec les nœuds appartenant aux neuf sous-champs grisés : le sous-champ où il appartient et les huit sous-champs voisins au sien.

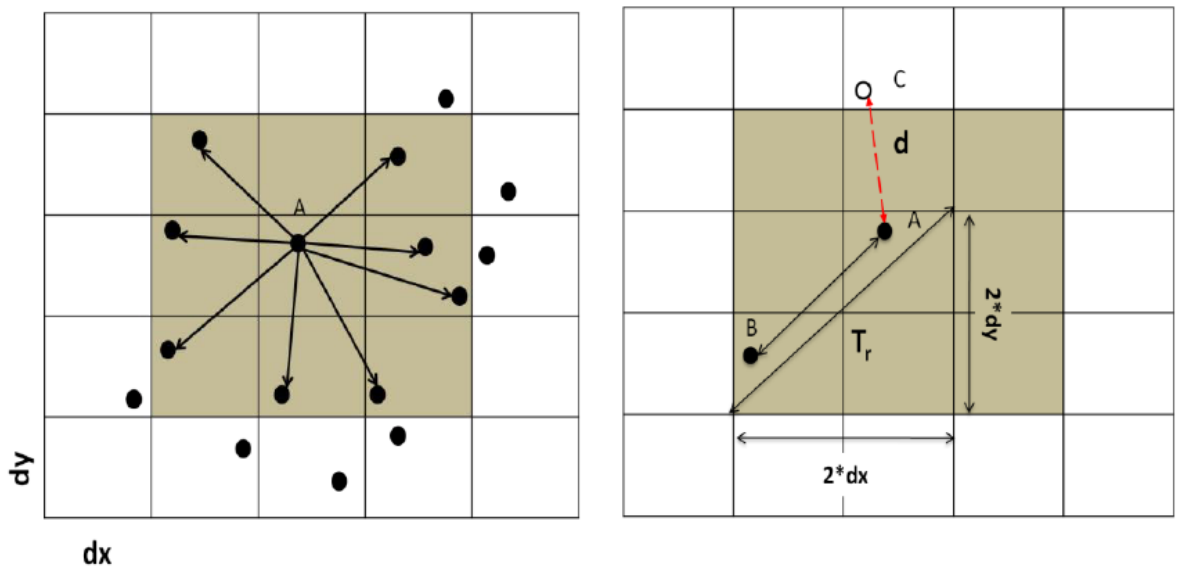

Figure V. 8. Champs de voisinage dans J-Sim

Le rayon maximal de transmission, nommé Tr, est la distance maximale entre deux noeuds communicants. Selon ce modèle de transmission, ce rayon est la diagonale du carré formé par quatre sous-champs (voir la Figure V.7, côté droit). Ainsi, ce rayon peut être calculé en utilisant la formule $Tr = 2\sqrt{(dx^2+dy^2)}$.

Le côté droit de la Figure V.7 décrit le comportement de trois nœuds communicants dans J-Sim en se basant sur le modèle de transmission décrit précédemment. Le nœud A ne peut pas communiquer avec le nœud C qui n'appartient à aucun sous-champ des neuf en gris (portée de transmission de A) même si la distance d entre eux est inférieure à Tr. Cependant, le nœud A peut communiquer avec le nœud B qui appartient à un des neuf sous-champs en gris. Ce modèle de propagation est spécifique à J-Sim, qui n'utilise donc ni le modèle théorique sphérique ni celui de voisinage qui en découle et est indépendant d'une distance constante entre l'émetteur et le récepteur. Ceci peut traduire une simulation de l'atténuation du signal à cause des obstacles ou d'autres facteurs (par exemple antenne unidirectionnelle).

Dans nos simulations, nous avons ajouté un code dans le code d'implémentation de J-Sim de façon à ce que notre protocole utilise un modèle de transmission sphérique pour les nœuds

du réseau car la gestion des obstacles dans le simulateur J-Sim n'est pas simple et les simulations se font généralement dans les conditions idéales.

V.5.2.2 Installation de J-Sim

Le fait que J-Sim soit développé en java lui offre l'avantage d'être un simulateur multiplateforme. Cela veut dire qu'il s'installe sur toute plateforme (Système d'exploitation) disposant d'une machine virtuelle Java (JVM : Java Virtual Machine).Pour cela avant d'installer J-Sim, on doit disposer de cette machine virtuelle. Le guide d'installation de J-Sim sous Windows et UNIX/Linux,est disponible sur le site officielle de J-Sim [86].

V.5.2.3 Intégration du code source du protocole dans J-Sim

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

Pour faciliter le développement du code source, il est nécessaire de choisir l'un des environnements de développements pour le langage Java. Il en existe plusieurs mais les plus populaires sont Eclipse IDE [89] et NetBean IDE [90] .Nous avons choisi NetBeans IDE pour sa facilité à créer les projets qui nous a permis d'ailleurs d'installer J-Sim dans le répertoire */project* de NetBeans IDE.

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

Nous créons un projet sous NetBeans dans lequel nous installons J-Sim.On crée ensuite les packages (au sens Java) de notre projet sont dans le répertoire */src/drcl/inet/sensosim* . Une fois le code source est sauvegardé, sous netBeans, nous le compilons par la commande « *ant compile* » (voir guide d'installation de J-Sim cité précédemment).

V.5.2.3 Définition de quelques méthodes du protocole

Parmi les méthodes de traitement de notre protocole nous pouvons donner :

- ✓ *protected void _start ()* : exécutée au démarrage d'un nœud.
- ✓ *protected void _stop()*: exécutée à l'arrêt d'un nœud

- ✓ *public synchronized void initialize()*:exécutée par le nœud puits pour initialiser le réseau.
- ✓ *protected synchronized void recvSensorEvent(Object data_)* :exécutée à la réception d'un stimulis généré par un nœud target.
- ✓ *protected synchronized void recvSensorPacket(Object data_)*:exécutée quand un noeud reçoit un paquet.
- ✓ *protected synchronized void hdlInitPkt(InitPkt inp_)*:exécutée quand un nœud reçoit un paquet d'initialisation.
- ✓ *public synchronized void hdlParentSearchReplayPkt (ParentSearchReplayPkt pr_):*execute par un noeud lorsqu'il reçoit un paquet de type *ParentSearchReplayPkt*
- ✓ *public synchronized void hdlIncEnrgLevelAlertPkt(EnrgLevelAlertPkt eLPkt_):*exécutée à la réception d'un paquet de type *EnrgLevelAlertPkt.*
- ✓ *public synchronized void sendPkt(Object data_,int type_):* exécutée par un nœud pour envoyer un paquet.
- ✓ *public synchronized void sendCongNotifPkt()*: exécutée par un nœud pour envoyer un paquet de notification de congestion.
- ✓ *public synchronized void hdlIncCongNotifPkt(CongNotifPkt congPkt_):*exécutée par un nœud lorsqu'il reçoit un paquet de notification de congestion.
- ✓ *public synchronized void replyParentSearchPkt(SensorPacket snpk_):*exécutée par un noeud pour répondre à un paquet *ParentSearchPkt*
- ✓ *public synchronized void processAggreg()*: pour effectuer l'agrégation
- ✓ *public synchronized void processOther(Object data_, Port inPort_)*: exécutée quand un composant reçoit un message via un port *inPort_*
- ✓ *protected synchronized void timeout(Object data_):*exécutée par un composant à l'expiration d'un timer relatif à un événement.
- ✓ *public synchronized boolean readyToSend()*: exécutée par un nœud pour tester l'état du canal sans fil et de la couche MAC avant d'envoyer les paquets à la couche inférieure.

V.5.2.4 Exécution et simulation

Pour lancer la simulation, nous avons deux possibilités :

Charger le fichier de configuration de notre protocole via l'interface RUV (voir paragraphe précedent),ou utiliser la commande *« java drcl.ruv.System –u nom_fichier_de_configuration ».*

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

V.5.2.5 Affichage des résultats de la simulation

J-Sim intègre un outil d'affichage des résultats de la simulation, appelé « plot » (voir les exemples sur les figures des résultats de simulation).Les utilisateurs de J-Sim peuvent choisir d'autres outils d'affichage ou d'interprétation des résultats non fournis par J-Sim, comme par exemple le logiciel que nous avons d'ailleurs utilisé pour générer la topologie(voir figure V.2).Il suffit de générer un fichier de configuration décrivant ce qu'on veut afficher dans le format compatible avec l'outil choisi.

V.6 Evaluation du protocole proposé.

Dans cette phase, nous allons évaluer le comportement de notre protocole en fonction des différents paramètres de configuration pour voir s'il répond bien à notre problématique.

V.6.1 Paramètres de simulation

Nous simulons un réseau de capteurs sans fil à topologie hiérarchique illustrée dans la figure V.2. Les paramètres pouvant être utilisés pour simuler notre protocole sont nombreux. La plupart d'entre eux sont les paramètres de configuration du protocole J-Sim.Le tableau ci-dessous donne un certain nombre de paramètres que nous avons utilisés pour évaluer notre protocole.

Tableau V. 8. Les paramètres de simulation.

Zone de déploiement	600 x 500
Nombre de nœuds capteurs	50, 100 ,150 et200 (stationnaires, déployés aléatoirement)
Nombre de nœuds puits	1 stationnaire à la position (350,350)
Nombre de nœuds target	2 (mobiles)
Données du domaine étudiées	Varient entre 0 et 100
Sensibilité	1
Durée du agregWaitingTimer	3s
Rayon de transmission sensor channel	200m
Rayon de transmission wireless channel	100m
Modèle de propagation	acoustique
Temps de simulation	1000s

V.6.2 Métriques d'évaluation

Le choix des métriques de performance pour évaluer un protocole dépend de l'objectif de l'évaluation et des relations entre ces métriques. Dans notre cas, nous utilisons une métrique d'évaluation qui englobe plusieurs métriques (consommation énergétique, réduction de la congestion au niveau du nœud et au niveau du canal sans fil, etc.).Dans ce qui suit, nous commençons par définir cette métrique et montrerons ensuite la relation entre cette métrique et d'autres métriques qu'on vient de citer.

Dans un protocole de contrôle de congestion par la technique d'agrégation des données, nous avons jugé important de définir une métrique appelée « taux *d'agrégation* » pour mesurer la performance de notre protocole.

Nous définissons le taux d'agrégation au niveau d'un nœud capteur i (T_i)comme étant le rapport entre le nombre de paquets collectés sur le nombre de paquets envoyés. Nous appelons ce taux le taux d'agrégation locale.

$$T_{i=\frac{Nombre\ de\ paquets\ collectés}{Nombre\ de\ paques\ envoyés}}$$

Au niveau du puits, ce taux est appelé « taux d'agrégation globale »(T_g).Puisque le nœud puits reçoit mais n'envoie pas de paquets de données, le taux d'agrégation globale est défini comme étant le rapport entre le nombre de paquets de données générés par l'ensemble des nœuds sources du réseau sur le nombre de paquets de données reçus par le nœud puits.

$$T_g = \frac{Nombre\ de\ paquets\ de\ données\ générés\ par\ les\ noeuds\ sources}{Nombres\ des\ paquets\ de\ données\ reçus\ par\ le\ noeud\ puits}$$

Ces taux sont toujours supérieurs à un .Plus le taux d'agrégation globale augmente, plus l'algorithme d'agrégation des données est performant et donc plus on évite la congestion.

V.6.3 Interprétation des résultats de la simulation

Avec les paramètres de simulation donnés dans le tableau précédent, et après un certain nombre de fois de simulations, nous obtenons les résultats suivants :

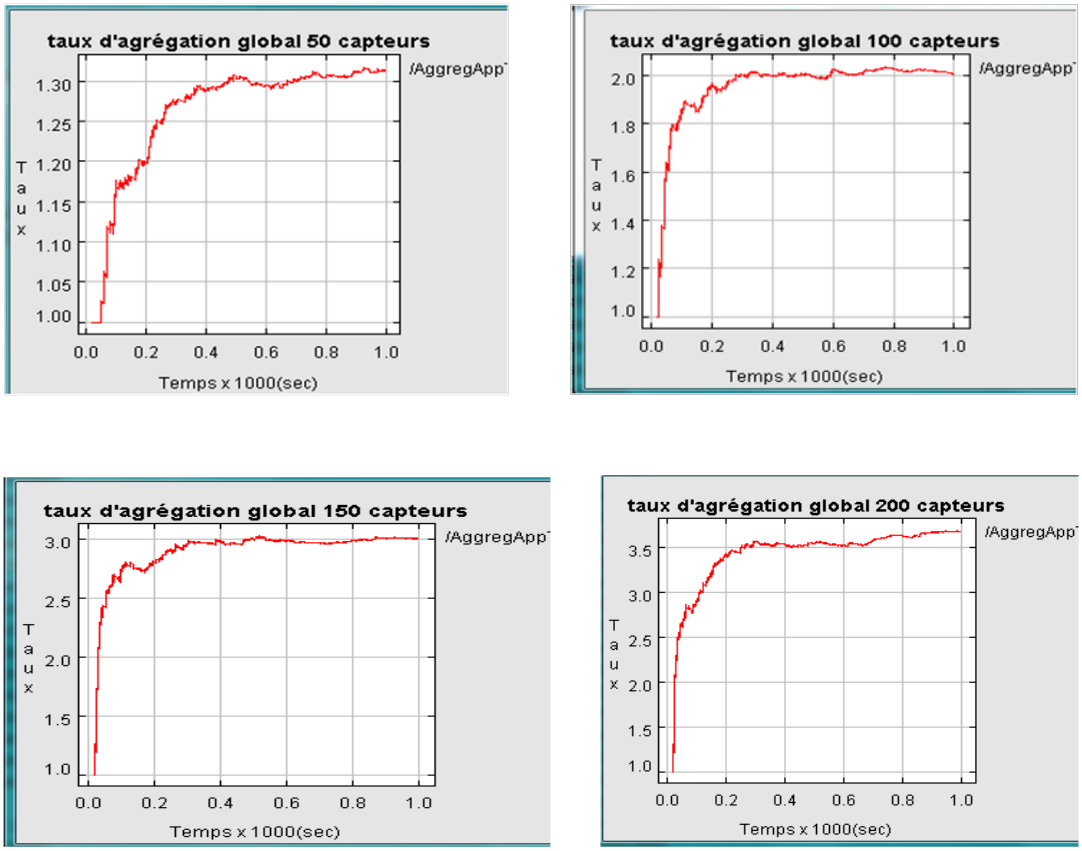

Figure V. 9. Résultats de la simulation : taux d'agrégation global.

D'après les résultats obtenus nous observons que le taux d'agrégation global est proportionnel au nombre de nœuds du réseau. Le protocole proposé est donc plus performant avec un certain grand nombre de nœuds capteurs.

Avec un taux d'agrégation global égal à x, cela signifie qu'en moyenne x paquets produits par l'ensemble des nœuds capteurs du réseau ont été combinés en un seul paquet. Ce taux varie en fonction de quelques autres paramètres comme la corrélation entre les données du domaine étudié, la taille du buffer de données, etc. En réduisant le nombre de transmission(x augmente), on réduit en même temps le taux d'occupation du média sans fil qui peut être causé par la sur-écoute (overhead) et donc on réduit la congestion au niveau du canal sans fil.

Du point de vue consommation énergétique, l'énergie consommée par les nœuds capteurs est inversement proportionnelle au taux x, puisque les modules radio des nœuds capteur vont passer la plus part de temps dans l'état d'attente (idle state), qui consomme moins d'énergie par rapport à l'état de réception (recepting state) et encore moins par rapport à l'état de transmission (transmitting state).

La figure suivante montre la variation de l'énergie restante en fonction du temps au niveau des nœuds capteurs.

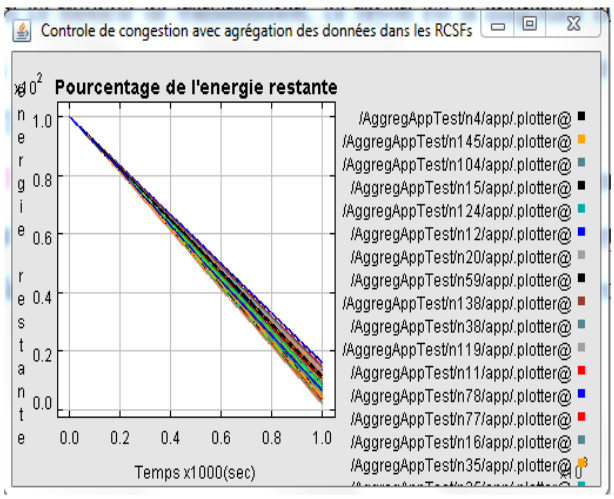

Figure V. 10. Consommation énergétique au niveau des nœuds capteurs.

Sur cette figure, on remarque que l'énergie est consommée d'une manière presque équitable. Cela peut être justifié par la réduction des sur-écoutes et la topologie hiérarchique et dynamique choisie.

V.7 Conclusion

Après avoir étudié les protocoles de contrôle de congestion destinés aux RCSFs, nous avons constaté que parmi les protocoles que nous avons pu rencontrer, il n'existe pas de protocoles utilisant la technique d'agrégation des données pour contrôler cette congestion.

Vu le degré de congestion pouvant être causée par les données redondantes, nous avons proposé un nouveau protocole de routage utilisant la technique d'agrégation des données pour contrôler cette congestion. Nous avons décrit ce protocole à travers un certain nombre de phases de communication et les résultats de la simulation sous le simulateur J-Sim nous montre que ce protocole se permet d'éviter la congestion au niveau du canal de transmission et au niveau des nœuds capteurs.

Conclusion Générale et Perspectives

Les réseaux de capteurs constituent un axe de recherche très fertile et peuvent être appliqués dans plusieurs domaines différents. Cependant, il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans les conditions réelles. L'un des problèmes qu'on peut rencontrer dans ce genre de réseau est la congestion causée par les données redondantes. Cette congestion est liée à la nature des réseaux de capteurs, principalement leur mode de déploiement. Quand un réseau est congesté, il y a un gaspillage des ressources dont la bande passante et l'énergie qui est une ressource critique dans les réseaux de capteurs. Il est donc important de mettre en place des protocoles de contrôle de congestion pour assurer le bon fonctionnement de ces réseaux.

Dans ce projet, nous avons mis en place un protocole de contrôle de congestion utilisant la technique d'agrégation des données. Au cours de ce travail qui a commencé par une étude bibliographique des protocoles existant, nous avons découvert un champ de recherche très intéressant qui touche presque tous les domaines d'application. Cela nous a donné d'ailleurs un grand plaisir d'intégrer la recherche.

Travailler sur un domaine comme les réseaux de capteurs, demande la prise en compte des contraintes que nous avons vues dans le premier chapitre de ce projet. Ces contraintes, nous les avons rencontrées réellement au moment de l'implémentation du code source, ce qui a été en quelque sorte un obstacle pour nous pendant un certain moment. Avec une motivation comme celle que nous avions pour mener ce travail, nous avons finalement réussi à nous en sortir et nous espérons que ce travail sera pour nous une grande motivation pour poursuivre la recherche.

Avec un peu de temps que nous avions pour mener ce travail, nous pouvons admettre qu'il y a beaucoup de perspectives pour notre travail. En effet nous aimerions comparer les performances de notre protocole avec celles de quelques protocoles de contrôle de congestion existant .Cela n'as pas été fait puisque non seulement il nous fallait avoir le code source de ces protocoles, mais aussi il fallait trouver un protocole qui utilise la technique d'agrégation des données.

Comme perspectives de notre travail, nous allons continuer à l'améliorer et surtout en prenant en compte d'autres métriques comme le taux de perte des paquets, le niveau de congestion au niveau de chaque nœud du réseau, sa scalabilité, l'intégration d'une base de données au niveau de l'ordinateur central pour collecter les données et le comparer avec

d’autres protocoles. Nous envisageons tous cela dans nos futurs travaux de recherche qu’ils soient d’ordre individuels, professionnels ou académiques.

Références Bibliographiques

Références Bibliographiques

[1] **S. Tilak, N. B. Abu-Ghazaleh** et **W. B. Heinzelman**, « A taxonomy of wireless micro-sensor network models », ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6,no. 2, pp. 28-36, April 2002.

[2] **H. Karl** et **A. Willig**, " Protocols and Architectures for Wireless Sensor Networks", Wiley, April 2005 .

[3] Internet Engineering Task Force (IETF), Groupe de travail MANET (mobile ad hoc network). http://www.ietf.org .

[4**] K. Romer et F. Mattern**, « The design space of wireless sensor networks», in IEEE Wireless Communications, 2006.

[5] **Y. Wang, G. Attebury** et **B. Ramamurthy**, « A Survey of Security Issues in Wireless Sensor Networks » , 2005 .

[6] Ibid

[7] **DARPA IPTO**, "SensIT : Sensor Information Technology program", 1999

[8] **J.N. AL-KARAKI** et **A. E. KAMAL**, « Routing Techniques in Wireless Sensor Networks: A Survey », IEEE wireless communication, 2004

[9] **E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis** et **V.Z. Groza** , " Sensor-based information appliances", IEEE Instrumentation MeasurementMagazine, vol.3, no.4, p.31-35, December 2000.

[10] Bluetooth SIG, " Specification of the Bluetooth system, Core, Version 1.1", Bluetooth SIG, www.Bluetooth.com .

[11] ZigBee Standards Organization .ZigBee Specification, Document 053474r13, December 2006.

[12] IEEE Computer Society. IEEE Standard for Information technology, "Telecommunications and information exchange between systems" , Local and metropolitan area networks ,Specific requirements.Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), October 2003.

[13] www.zigbee.org .

[14] http ://www.maxstream.net.

[15] LAN-MAN Standards Committee of the IEEE Computer Society, "802.15.4 IEEE Standard for Information technology", Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), IEEEStd 802.15.4-2003 (2003) .

[16] **Dragos NICULESCU**, « Topics In Ad-Hoc Networks: Communication Paradigms for Sensor Networks », NEC Laboratories America, IEEE Communications Magazine, Mars 2005.

[17] **E. Callaway, P. Gorday, L. Hester, J.A. Gutierrez, M. Naeve, B.Heile** et **V. Bahl** "Home Networking with IEEE 802.15.4 : a developing standard for low-rate wireless personal area Networks » ,IEEE Communications Magazine vol. 40, no. 8, p 70-77, Août. 2002

[18] **F. Brissaud ,D. Charpentier, A. Barros** et **C. Bérenguer**, « nouvelles technologies et nouvelles problématiques pour la sureté de fonctionnement », Maîtrise des Risques et de Sûreté de Fonctionnement, Lambda-Mu 16, Avignon : France (2008)

[19] **Vernon S. Somerset**, "Intelligent and Biosensors, Edited by Vernon S. Somerset", Intech, January 2010

[20] **Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam** et **Erdal Cayirci** "A Survey on Sensor Networks",IEEE Communications Magazine,P.102-114,August 2002

[21] **Alcides Montoya, Diana Carolina Restrepo** et **Demetrio Arturo Ovalle**, "Artificial Intelligence for Wireless Sensor Networks Enhancement",InTech,2010

[22] **QinghuaWang** et **Ilangko Balasingham**,"Wireless Sensor Networks – An Introduction"

[23] **Ben L. Titzer** et **Jens Palsberg** , "Nonintrusive Precision Instrumentation of Microcontroller Software"

[24] **Holger Karl** et **Andreas Willig**, "Protocols and Architectures for Wireless Sensor Networks", John Wiley & Sons, 2005

[25] **Bhaskar Krishnamachari** , "Networking Wireless Sensors", Cambridge university press(NEWYORK), 2005

[26] **Gaurav Jolly, Mustafa C. Kusçu, Pallavi Kokate** et **Mohamed Younis**, "A Low-Energy Key Management Protocol for Wireless Sensor Networks", ,Proceedings of the Eighth IEEE International Symposium on Computers and Communication ,(ISCC'03), IEEE COMPUTER SOCIETY,2003

[27] **Shio Kumar Singh** et **al**, "Applications, Classifications, and Selections of Energy-Efficient Routing Protocols for Wireless Sensor Networks", INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES (IJAEST), Vol No. 1, Issue No. 2, p85 – 95

[28] **VIOLET R. SYROTIUK , BING LI** et **ANGELA M. MIELKE** , "Heterogeneous Wireless Sensor Networks"

[29] **Gérard CHALHOUB**, "Les réseaux de capteurs sans fil "

[30] **Moshaddique Al Ameen, S.M. Riazul Islam** et **Kyungsup Kwak**, "Energy Saving Mechanisms forMAC Protocols in Wireless Sensor Networks" ,International Journal of Distributed Sensor Networks, Volume 2010, Article ID 163413, 16 pages

[31] **Cédric Lamorinière** et **Damien Ligot**, « Réseaux de capteurs »

[32] **Chris Townsend** et **Steven Arms**, "Wireless Sensor Networks: Principles and Applications",MicroStrain, Inc.

[33] **H.ALATRISTA, S.ALIAGA, K GOUAÏCH** et **J.MATHIEU**, « Implementation d'une plateforme de réseaux de capteurs sans-fil»,TER Master1 informatique,Université de Montpellier 2, 29 Avril 2008.

[34] http://www.ieee802.org/11/

[35] http://www.ieee802.org/16/

[36] **Raja Jurdak**, « Wireless Ad Hoc and Sensor Networks:A Cross-Layer Design perspective»,University College Dublin,2007.

[37] **C. INTANAGONWIWAT, R. GOVINDAN** et **D. ESTRIN**, « Directed diffusion: A scalable and robust communication paradigm for sensor networks», Proceedings ACM MobiCom'00, pp. 56-67, Boston, MA, Août 2000.

[38] **C. INTANAGONWIWAT, R. GOVINDAN, D. ESTRIN, J. HEIDEMANN** et **F. SILVA,** « Directed diffusion for wireless sensor networking», IEEE/ACM Transactions on Networking, vol. 11, N°1, pp.2-16,Fevrier 2003,.

[39] **D. BRAGINSKY** et **D. ESTRIN**, « Rumor Routing Algorithm for Sensor Networks » dans les démarches du premier atelier sur les réseaux de capteurs et applications (WSNA), Atlanta, GA, Octobre 2002.

[40] **J. KULIK, W. HEINZELMAN** et **H. BALAKRISHNAN,** « Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks», Wirel. Netw, Vol. 8, pp.169-185,2002.

[41] **Kemal AKKAYA** et **Mohamed YOUNIS**, « A Survey on Routing Protocols for Wireless Sensor Networks».

[42] **W.R. HEINZELMAN, A. CHANDRAKASAN** et **H. BALAKRISHNAN**, «Energy-efficient Communication Protocol for Wireless Microsensor Networks», dans les démarches de la société d'ordinateur d'IEEE à la trente troisième conférence internationale d'Hawaï sur les sciences de systèmes (HICSS '00),Vol. 8, pp. 8020, Washington DC, USA, Janvier 2000.

[43] **W.R. Heinzelman, A. Chandrakasan** et **H. Balakrishnan**, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks" in IEEE Tmnsactions on Wireless Communications (October 2002), vol. 1(4), pp. 660-670.

[44] **Jamal AL-KARAKI** et **Ahmed E. KAMAL**, «Routing Techniques in Wireless Sensor Networks: A Survey», Magazine: IEEE Communications, vol. 11, N° 6, pp. 6-28, Dec. 2004.

[45] **Y. YAO** et **J. GEHRKE**, « The cougar approach to in-network query processing in sensor networks », dans l'enregistrement de SIGMOD, Septembre 2002.

[46] **Sadagopan N., Krishnamachari B.** et **Helmy A.** "The ACQUIRE Mechanism for Efficient Querying in Sensor Networks", In Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications (SNPA), pp. 149–155,Anchorage, AK, May, 2003,

[47] **Y. YU, D. ESTRIN** et **R. GOVINDAN,** «Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks» rapport technique du department d'informatique de l'université UCLA, May 2001.

[48] **I. F. AKYILDIZ, W. SU, Y. SANKARASUBRAMANIAM** et **E. CAYIRCI**, « Wireless Sensor networks: a Survey » revue des réseaux informatiques, Vol. 38, N° 4, pp. 393-422, Mars 2002,

[49] **Shio Kumar SINGH, M P SINGH** et **D K SINGH**, « Routing Protocols in Wireless Sensor Networks –A Survey » revue international de l'informatique et de l'enquête d'ingenierie (IJCSES) Vol.1, N°2, Novembre 2010.

[50] **Luis JAVIER García VILLALBA** et **Ana Lucila SANDOVAL OROZCO**, Alicia TRIVIÑO CABRERA et Cláudia Jacy BARENCO ABBAS, « Review: Routing Protocols in Wireless Sensor Networks », 26 Octobre 2009, [En ligne], www.mdpi.com/journal/sensors.

[51] **Mohammad ILYAS** et **Imad MAHGOUB**, «Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems», CRC Press LLC, 2005.

[52] **T. He** et **al.** , «SPEED: A stateless Protocol for real-time communication in Sensor networks», dans les démarches de la conférence internationale sur des systèmes de l'informatique répartie, Providence, RI, Mai 2003.

[53] **A. MANJESHWAR** et **D.P. AGRAWAL**, « APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks », 16ème Colloque International sur le traitement parallèle et distribué (IPDPS-02), 2002.

[54] **A. MANJESHWAR** et **D.P. AGRAWAL**, « TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks », 15ème colloque international sur le traitement parallèle et distribute (IPDPS-01), 2001.

[55] **Rekha CHAKRAVARTHI, C. GOMATHY, Suraj K. SEBASTIAN, K. PUSHPARAJ** et **Vinu Binto MON**, «A Survey on Congestion Control in Wireless Sensor Networks», Revue internationale de l'informatique et de la communication, Vol. 1, N°1, pp. 161-164, Janvier-Juin 2010.

[56] **R.THEN MALAR** « Congestion Control in Wireless Sensor Networks Based Multi-Path Routing In Priority Rate Adjustment Technique»

[57] **Bhaskar KRISHNAMACHARI**, «Networking Wireless Sensors », Cambridge University Press, 2005.

[58] **G. Bianchi** «Performance analysis of the IEEE 802.11 distributed coordination function»,IEEE Journal on Selected Areas in Communications, March 2000.

[59] **C.Y. WAN, S. B. EISENMAN** et **A. T. CAMPBELL**, « CODA: Congestion Detection and Avoidance in Sensor Networks » démarches de ACM SenSys, Novembre 2003.

[60] **C.-T. EE** et **R. BAJCSY**, « Congestion Control and Fairness for Many-to-One Routing in Sensor Networks », démarches de ACM SenSys, November 2004.

[61] **C. HAOWEN** et **A. PERRIG**, « Security and privacy in sensor networks », Computer, vol. 36, pp. 103–105, 2003.

[62] **R. SZEWCZYK, E. OSTERWEIL, J. POLASTRE, M. HAMILTON, A. MAINWARING** et **D. ESTRIN**, « Habitat monitoring with sensor networks », Communications de l'ACM, vol. 47, pp. 34–40, 2004.

[63] **Z. FENG, S. JAEWON** et **J. REICH**, « Information-driven dynamic sensor collaboration », Magazine: Signal Processing, vol. 19, pp. 61–72, 2002, IEEE.

[64] **A. D. JOSEPH**, « Energy harvesting projects », Pervasive Computing, vol. 4, pp. 69–71, 2005, IEEE.

[65] **Sudip MISRA, Isaac WOUNGANG** et **Subhas Chandra MISRA**, « Guide to Wireless Sensor Networks », Springer, 2009.

[66] **Aldar CHAN** et **Claude CASTELLUCCIA**, « On the security of concealed data aggregation », dans le Colloque européen sur la recherche dans le domaine de la sécurité d'ordinateur (ESORICS 2007), Septembre 2007, Dresden, Allemagne.

[67] **Elena FASOLO, Michele ROSSI, Jörg WIDMER** et **Michele ZORZI**, «In-network Aggregation Techniques for Wireless Sensor Networks: A Survey ».

[68] **Kiran MARAIYA, Kamal KANT** et **Nitin GUPTA**, « Wireless Sensor Network: A Review on Data Aggregation », revue internationale de la recherche scientifique et technologique, Vol. 2, N°4, Avril 2011.

[69] **T. ARAMPATZIS , J. LYGEROS** et **S. MANESIS**, « A Survey Of Applications Of Wireless Sensors And Wireless Sensor Networks », dans la conference méditerranéenne sur la commande et l'automatisation MED05, 2005, Nicosia, Cyprus.

[70] **I. SOLIS** et **K. OBRACZKA**, « The Impact of Timing in Data Aggregation for Sensor Networks » , dans IEEE ICC 2004, Juin 2004, Paris, France.

[71] **X. JIANBO, Z. SILIANG** et **Q. FENGJIAO**, « A new In-network data Aggregation technology of wireless sensor networks », Dans IEEE SKG'06, Novembre 2006, Guilin, China.

[72] **K. AKKAYA et M. YOUNIS**, « A Survey of Routing protocols in wireless Sensor networks », revue Elsevier Ad Hoc Network, vol. 3, N° 3, pp. 325–349, Mai 2005.

[73] **C. INTANAGONWIWAT, R. GOVINDAN, D. ESTRIN, J. HEIDEMANN et F. SILVA**, « Directed diffusion for wireless sensor Networking », IEEE/ACM Trans. Networking, vol. 11, N°1, pp. 2–16, Février 2002.

[74] **S. LINDSEY, C. RAGHAVENDRA et K. M. SIVALINGAM**, « Data Gathering Algorithms in Sensor Networks using Energy Metrics », IEEE Trans. Parallel Distrib. Syst., vol. 13, N°. 9, pp. 924–935, Septembre 2002.

[75] **Y. XU, J. HEIDEMANN et D. ESTRIN**, « Geographic-Informed Energy Conservation for Ad Hoc Routing », dans ACM/SIGMOBILE MobiCom 2001, Juillet 2001, Rome, Italie.

[76] **S. MADDEN, M. J. FRANKLIN, J. M. HELLERSTEIN et W. HONG**, «TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks», dans OSDI 2002, Décembre 2002 Boston, MA, US.

[77] **S. NATH, P. B. GIBBONS, Z. R. ANDERSON** et **S. SESHAN**, « Synopsis Diffusion for Robust Aggregation in Sensor Networks », dans ACM SenSys 2004, Novembre 2004, Baltimore, MD, US.

[78] **A. MANJHI, S. NATH** et **P. B. GIBBONS**, « Tributaries and Deltas: Efficient and Robust Aggregation in Sensor Network Stream », dans ACM SIGMOD 2005, Juin 2005, Baltimore, MD, US.

[79] **A. Sharaf, J. Beaver, A. Labrinidis** et **K. Chrysanthis**, "Balancing energy efficiency and quality of aggregate data in sensor networks," The VLDB Journal, vol. 13, no. 4, pp. 384–403, Dec. 2004.

[80] **E. CAYIRCI**, « Data Aggregation and Dilution by modulus addressing in wireless sensor networks », Lettres De Communications d'IEEE, vol. 7, N° 8, pp. 355–357, Aout 2003.

[81] **N. SHRIVASTAVA, C. BURAGOHAIN, D. P. AGRAWAL** et **S. SURI**, « Medians and Beyond: New Aggregation Techniques for Sensor Networks », dans ACM SenSys 2004, Novembre 2004, Baltimore, MD, US.

[82] **Emmanuel Conchon**, « Définition et mise en oeuvre d'une solution d'émulation de réseaux sans fil », Institut National Polytechnique De Toulouse, Octobre 2006

[83] **Gianni A. Di Caro**, « Analysis of simulation environments for mobile ad hoc Networks »,Technical Report No. IDSIA-24-03, Dalle Molle Institute for Artificial Intelligence, Galleria 2, 6928 Manno, Switzerland, 2003.

[84] **Marek Małowidzki**, « Network simulators: a developer's perspective »

[85] **Johannes Lessmann, Peter Janacik, Lazar Lachev et Dalimir Orfanus**, « Comparative Study of Wireless Network Simulators»,Seventh International Conference on Networking,Octorber 2008.

[86] http://sites.google.com/site/jsimofficial/

[87] Tcl Developper Site. [En ligne] Mai 2011. http://www.tcl.tk/.

[88] **Ahmed Sobeih et Jennifer C. Hou**, « A simulation framework for sensor networks in j -sim »

[89] http://www.eclipse.org/

[90] http://netbeans.org/

[91] **Karl Baumgartner** « Réseaux de capteurs sans fil »