

République Algérienne Démocratique et Populaire  
Université Abou Bakr Belkaid– Tlemcen  
Faculté des Sciences  
Département d'Informatique

**Mémoire de fin d'études**  
**pour l'obtention du diplôme de Master en Informatique**

*Option: Réseaux et Systèmes Distribués (RSD)*

*Thème*

**Conception d'un algorithme de routage basé sur  
l'heuristique du recuit simulé pour les réseaux de  
capteurs à grande échelle**

**Réalisé par :**

- **ABDESSELAM Abdelhalim**
- **BELOUATEK Mohammed**

*Présenté le 02 Juillet 2013 devant le jury composé de MM.*

- *Mr BENMAMMAR...Badr ..... (Président)*
- *Mr LEHSAINI...Mohamed..... (Encadreur)*
- *Mr BELHOUCINE...Amine ..... (Examineur)*
- *Mme...BELHABI...AMAL..... (Examineur)*

Année universitaire : 2012-2013

## Remerciement

En préambule à ce travail, nous souhaitons adresser nos remerciements les plus sincères aux personnes qui nous ont apporté leur aide et leur soutien et qui ont contribué à l'élaboration de ce projet de fin d'études.

Nos premiers remerciements à **ALLAH** le Tout Puissant.

Nous tenons pour commencer à vous adresser, chers président et les membres du jury, nos sincères remerciements pour votre lecture, votre présence et les remarques qui viendront enrichir ce travail que nous aurons plaisir à partager et discuter avec vous.

Un grand remerciement à Monsieur LEHSAINI Mohammed, l'encadreur de notre recherche de ce mémoire, qui s'est montré à l'écoute et disponible, ainsi pour l'inspiration, son aide précieuse et pour le temps qu'elle a bien voulu nous consacrer.

Nous remercions encore fois le professeur de l'Université Abou Bakr Belkaid-Tlemcen qui est su nous aiguiller vers des spécialistes et nous donner des références essentielles, particulièrement monsieur LEHSAINI Mohammed.

Merci à notre amis BADAOUI Abdellatif et MOSTEFAOUI Mohammed el Amine, étudiantes en Master 2 Abou Bakr Belkaid-Tlemcen travaillant tout presque comme nous sur le mémoire, qui nous a soutenus dans nos recherches.

Nous remercions également notre camarade de Master II Réseau Système Distribué, et nos amis qui ont été terminé son étude au License, pour leurs conseils et leurs idées. Nous le tous remercions pour nous avoir soutenues, encouragé ainsi que pour nous avoir fait partager ses remarques lors de la relecture de ce mémoire.

Un grand Merci chaleureux et de tous nos cœur à nos familles "ABDESSELAM" et "BELOUTEK", sans qui nous ne serais absolument pas où nous en sommes aujourd'hui. Nous les remercions sincèrement pour leur gentillesse et leur soutien inconditionnel et constant, pour nous avoir donné du courage et de l'espoir, pour être toujours présents même à distance. Nous leur devons ce que nous sommes. Aussi un Merci de tout mon cœur à nos chers frères pour leur gentillesse et leur encouragement à continuer mes études. Nous vous remercions de tous nos cœurs.

Nous saisons cette occasion pour exprimer notre profonde gratitude en vous témoignant notre reconnaissance et notre respect.

*ABDESSELAM & BELOUATEK*

## *Dédicace*

*Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire " Ya Kayoum "*

*Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à mes chers parents qui m'ont éclairé le chemin de la vie par leur grand soutien et leurs encouragements, par leurs dévouements exemplaires et les énormes sacrifices qu'ils m'ont consentis durant mes études et qui ont toujours aimé me voir réussir .Je les remercie pour tout ce qu'ils m'ont fait.*

*Que dieu les gardes et les protège.*

*A mes frères : Mounir, Ismail et mes sœurs*

*A mon binôme Mohammed qui m'a supporté durant le travail.et chez qui j'ai trouvé l'entente dont j'avais besoin.*

*A mes cousins*

*A mes oncles : Abderrahmane, Laredj, Boumediene,Sidi Mohammed*

*A tous mes amis de SIC, MID, RSD et surtout notre groupe 'BALDA' : AZAIZ*

*Benyahia, ABDERRAHIM Mohammed el Amine,BEDJBOUJA Mohammed,*

*BENAMMAR Abdeladim, BENYAHIA Belahcen, BENHAMIDA Mahmoud,*

*BADAOUI Abdellatif, MOSTEFAOUI Mohammed El Amine. METAHRI*

*Abdelmadjid*

*ABDESSELAM A*

## *Dédicace*

*Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire " Ya Kayoum "*

*Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à mes chers parents qui m'ont éclairé le chemin de la vie par leur grand soutien et leurs encouragements, par leurs dévouements exemplaires et les énormes sacrifices qu'ils m'ont consentis durant mes études et qui ont toujours aimé me voir réussir .Je les remercie pour tout ce qu'ils m'ont fait.*

*Que dieu les garde et les protège.*

*A mon frère Bilal et mes sœurs*

*A mon binôme Abdelhalim qui m'a supporté durant le travail.et chez qui j'ai trouvé l'entente dont j'avais besoin.*

*A mes cousins Okacha,Aissa,Youcef,Ahmed,Abdelkader*

*A mes oncles*

*A tous mes amis de SIC, MID, RSD et surtout notre groupe 'BALDA' : AZAIZ Benyahia, ABDERRAHIM Mohammed el Amine,BEDJBOUJA Mohammed, BENAMMAR Abdeladim, BENYAHIA Belahcen, BENHAMIDA Mahmoud, BADAOUI Abdellatif, MOSTEFAOUI Mohammed El Amine,METAHRI Abdelmadjid ,BELHADJ Abdelkader ,DAHAWI Mohammed et a tous qui connaitre BELOUATEK Mohammed et ABDESSELAM Abdelhalim*

*BELOUATEK M*

## *Table des matières*

|   |    |
|---|----|
| Introduction générale.....  | 3  |
| I Chapitre I Les protocoles de routage dans les réseaux sans fil..... | 5  |
| I.1 Introduction .....  | 5  |
| I.2 Les réseaux de capteurs sans fil.....                             | 5  |
| I.2.1 Description d'un nœud capteur .....                             | 6  |
| I.2.2 Caractéristiques d'un RCSF.....                                 | 7  |
| I.2.3 Domaines d'applications des RCSF .....                          | 8  |
| I.3 L'architecture des RCSF .....                                     | 10 |
| I.3.1 Architecture de communication .....                             | 10 |
| I.3.2 Collecte d'informations.....                                    | 11 |
| I.3.3 Architecture protocolaire.....                                  | 12 |
| I.4 Routage dans les RCSF .....                                       | 14 |
| I.4.1 Approches d'établissement de routes .....                       | 14 |
| I.4.2 Protocoles basés sur l'architecture réseau .....                | 15 |
| I.4.3 Les protocoles de routage hiérarchiques .....                   | 16 |
| I.5 Les limites des protocoles de routage dans les RCSF .....         | 17 |
| I.6 Conclusion .....  | 18 |
| II Chapitre II Présentation du protocole GSR .....                    | 19 |
| II.1 Introduction .....   | 19 |
| II.2 Modèle de réseau .....   | 19 |
| II.3 Présentation de GSR.....   | 20 |
| II.3.1 Schéma algorithmique de GSR .....                              | 21 |
| II.3.2 Diffusion de l'information .....                               | 22 |
| II.3.3 Le plus court chemin dans GSR.....                             | 22 |
| II.3.4 Points forts et limitations de.....                            | 23 |
| II.4 Analyse de la complexité de GSR .....                            | 23 |

|         |  |    |
|---------|--|----|
| II.5    | Conclusion .....   | 24 |
| III     | Chapitre III            Implémentation de GSR modifié..... | 25 |
| III.1   | Introduction .....   | 25 |
| III.2   | Développement de GSR_M.....                                | 25 |
| III.3   | Les heuristiques .....                                     | 28 |
| III.3.1 | Heuristique à une solution unique.....                     | 29 |
| III.3.2 | Heuristiques à solutions multiples.....                    | 29 |
| III.3.3 | Domaines d'utilisation des heuristiques .....              | 30 |
| III.4   | Heuristique utilisée : Recuit simulé.....                  | 30 |
| III.4.1 | Principes du recuit simulé .....                           | 31 |
| III.5   | Evaluation des performances.....                           | 33 |
| III.5.1 | Evaluation du temps d'exécution .....                      | 33 |
| III.5.2 | Evaluation du coût moyen des routes.....                   | 35 |
| III.5.3 | Evaluation du taux de précision .....                      | 36 |
| III.6   | Conclusion .....   | 38 |
|         | Conclusion générale .....                                  | 39 |
|         | Références Bibliographiques.....                           | 40 |
|         | Liste des figures .....                                    | 42 |
|         | Liste des tableaux .....                                   | 43 |
|         | Liste des abréviations .....                               | 44 |

# Introduction générale

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

Les réseaux de capteurs sans fil (RCSF) sont généralement composés d'un grand nombre de nœuds capteurs qui sont déployés aléatoirement dans une zone intérêt pour surveiller un tel phénomène. Ces capteurs collaborent entre eux et remontent l'information de nœud en nœud selon le mode multi-sauts et via une connexion sans fil jusqu'à l'aboutissement à la station de base. Or, ce schéma de communication est coûteux en termes d'énergie puisque tout nœud joue le rôle d'un nœud relai puisqu'il renvoie l'information reçue pour la première à ses voisins. Ce processus de communication se répète jusqu'à l'arrivée de l'information à la station de base. De ce fait, pour se remédier à cet inconvénient, plusieurs protocoles de routage ont été proposés dans la littérature d'une part pour communiquer l'information rapidement à la station de base et d'autre part pour communiquer l'information à la station de base à un moindre coût. Dans cette optique, nous avons proposé une version du protocole GSR qui est conçu pour les réseaux ad hoc et la faire adapter aux réseaux de capteurs. Donc, nous avons proposé une version simplifiée et améliorée de ce protocole appelée GSR\_M. Dans cette version, les tables de routage seront allégées puisque nous intéressons à l'établissement de routes entre tout nœud du réseau et la station de base contrairement à la version originale dans laquelle les routes sont établies entre tous les nœuds. En outre, dans la version originale pour l'établissement de routes entre nœuds l'algorithme de Dijkstra qui a été utilisé. Or, la complexité de cet algorithme est consistante en termes de temps de calcul et surtout lorsqu'on passe à l'échelle. De ce fait, pour réduire le temps d'établissement de routes entre tout nœud et la station de base nous avons fait appel à l'heuristique du recuit simulé. Puis nous avons évalué les performances de GSR\_M et du schéma de routage basé sur le recuit simulé pour montrer l'apport en termes de temps de calcul de notre contribution.

Notre contribution consiste à réduire le temps pour remonter l'information à la station de base tout en essayant de réduire le coût du chemin parcouru comparativement à la version GSR\_M. Ceci rend notre contribution adaptée aux applications orientées événements dans lesquelles le facteur temps est important pour remonter l'information à la station de base.

Ce mémoire s'articule autour de trois chapitres. Dans le premier chapitre, nous présentons les réseaux sans fil, leurs caractéristiques et nous faisons le point sur le routage dans les réseaux sans fil en particulier dans les réseaux de capteurs. Le deuxième chapitre est une présentation du protocole de routage GSR. Dans le troisième chapitre, nous proposons une version améliorée de GSR qui est adaptée aux RCSF et une autre basée sur l'heuristique du recuit simulé, et nous évaluons la version améliorée et celle basée sur le recuit simulé. Enfin, nous concluons notre travail et nous présentons quelques perspectives dans cette optique.

# Chapitre I

## Les protocoles de routage dans les réseaux sans fil

### I.1 Introduction

Dans les réseaux sans fil, les protocoles de routage permettent d'établir des routes entre les nœuds pour acheminer les paquets entre eux. Cependant, dans les réseaux de capteurs, les protocoles de routage établissent des routes entre tout nœud du réseau et la station de base pour assurer la fidélité de routage. Dans cette optique, plusieurs protocoles ont été proposés dans la littérature. Il y en a de nouveaux protocoles qui ont été développés comme il y en a des protocoles qui sont des améliorations des autres et essaient de combler les limitations des versions originales. Par exemple, le protocole GSR est une amélioration de protocole DSDV.

Le routage dans les RCSF doit prendre en compte deux contraintes. Premièrement, remonter rapidement l'information d'un nœud à la station de base en empruntant le plus court chemin. Deuxièmement, remonter l'information à la station de base à moindre coût en termes de consommation d'énergie.

Dans les premiers protocoles de routage conçus aux RCSF, chaque nœud détecte et participe au routage de l'information. De ce fait, quand un nœud reçoit un paquet pour la première fois, il le retransmet dans son voisinage sinon il l'ignore. Ce processus d'acheminement de l'information se répète jusqu'à l'aboutissement à la station de base.

Dans ce chapitre, nous présentons tout d'abord les réseaux de capteurs sans fil, leurs caractéristiques et leurs concepts. Cette description nous permet de connaître la particularité des protocoles de routage conçus pour ce type de réseaux. Puis, nous présentons les protocoles de routage et les différentes architectures sur lesquelles ils se basent.

### I.2 Les réseaux de capteurs sans fil

Les réseaux capteurs sans fil sont des réseaux ad-hoc sans infrastructure qui sont constitués d'un grand nombre de nœuds appelés nœuds capteurs. Ces derniers sont des petits dispositifs qui sont déployés aléatoirement dans un champ de captage pour

surveiller un tel phénomène et faire remonter une grandeur physique, physiologique ou environnementale telle que glycémie, température, ou pression atmosphérique, etc. à un centre de contrôle distant. Les capteurs se caractérisent par une unité de calcul très limitée, une mémoire de faible capacité et ils sont dotés de batteries qui sont généralement non rechargeables et non remplaçables puisque les capteurs sont fréquemment déployés dans des zones à accès difficile pour l'être humain.

### I.2.1 Description d'un nœud capteur

Un capteur sans fil est un équipement physique utilisé dans les réseaux de capteurs sans fil. Cet équipement est doté d'une unité de communication qui utilise les ondes radio pour recevoir ou transmettre les données vers d'autres capteurs ou à la station de base. Il est composé de quatre unités dont chacune a un rôle bien déterminé [4] comme s'est illustré par la figure I.1.

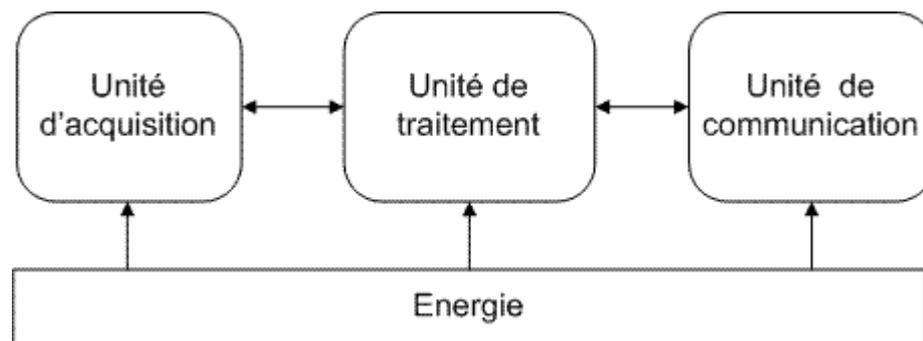


Figure I.1: Eléments d'un capteur sans fil [12]

- **L'unité d'acquisition** : est généralement composée de deux sous-unités : les capteurs (les senseurs) et les convertisseurs analogique-numérique (ADCs<sup>1</sup>). Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les ADCs convertissent ces signaux analogiques en signaux numériques.
- **L'unité de traitement** : Cette unité est composée de deux interfaces : une interface avec l'unité d'acquisition et une autre avec le module de transmission. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition, et stocke les données collectées.

---

<sup>1</sup> ADCs: Analog/Digital Converter

- **Un module de communication (Transceiver) :** il est responsable de toutes les communications via un support de communication radio qui relie le nœud aux autres nœuds du réseau.
- **Batterie :** alimente les unités citées précédemment.

Il existe des capteurs qui sont dotés d'autres composants additionnels : les systèmes de localisation tels que GPS (Global Position System) et un mobilisateur lui permettant le déplacement.

## I.2.2 Caractéristiques d'un RCSF

Un réseau de capteurs présente des caractéristiques particulières comparativement aux autres réseaux sans fil. Dans cette section, nous présentons les principales caractéristiques de ces réseaux :

- **Sans infrastructure :** Les RCSF appartiennent à la famille des réseaux sans fil sans infrastructure dit ad-hoc de type many-to-one. Les capteurs sont généralement déployés aléatoirement dans des zones hostiles ce que nécessite qu'ils doivent s'auto-configurer et s'auto-organiser sans intervention humaine.
- **Scalabilité (Passage à l'échelle) :** Dans les RCSF, les capteurs sont déployés généralement en grand nombre pour garantir la couverture totale de la zone d'intérêt et faire face aux pannes puisque les capteurs peuvent cesser de fonctionner pour différentes causes. Nous pouvons avoir dans certains cas des RCSF de haute densité dont la taille dépasse mille capteurs voire un million de capteurs.
- **Interférences :** La notion d'interférences apparaît dans la plus part des réseaux sans fil en particulier dans les RCSF où deux capteurs voisins peuvent transmettre dans le même bond de fréquences ce qui peut causer des interférences.
- **Topologie dynamique :** Les capteurs sans fil peuvent être placés sur des objets mobiles par exemple sur des animaux pour les surveiller à distance sans perturber leur comportement. Ce type de scénario génère une topologie qui n'est pas statique dite dynamique [1].
- **Contrainte d'énergie, de stockage et de calcul :** La caractéristique la plus critique dans les RCSF c'est la limite des ressources énergétiques car chaque la plupart des capteurs sont dotés de piles à énergie limitée. A cet effet, dans la plupart des travaux

de recherche, une problématique est traitée conjointement avec l'économie de l'énergie.

### **I.2.3 Domaines d'applications des RCSF [15]**

Le faible coût et la communication sans fil ont permis aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre le domaine des applications existantes. Parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions, on peut citer le militaire, la santé, l'environnemental, et les maisons intelligentes....

#### **a. Applications militaires**

Le déploiement rapide, l'auto-organisation et la tolérance aux pannes sont des caractéristiques qui ont rendu les réseaux de capteurs efficaces pour les applications militaires.

Plusieurs projets ont été lancés pour aider les unités militaires dans un champ de bataille et protéger les villes contre des attaques, telles que les menaces terroristes. Le projet DSN (Distributed Sensor Networks) au DARPA<sup>2</sup> était l'un des premiers projets dans les années 80 ayant utilisés les réseaux de capteurs pour rassembler des données distribuées.

De même, un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations). L'armée américaine a réalisé des tests dans le désert de Californie.

#### **b. Applications à la surveillance**

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

---

<sup>2</sup> DARPA: Defense Advanced Research Projects Agency

### c. Applications environnementales

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermocapteurs dans une forêt peut aider à détecter un éventuel début d'incendie et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains peut aider à détecter la pollution et analyser la qualité d'air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

En outre, dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir convenablement aux changements climatiques par exemple le processus d'irrigation lors de la détection de zones sèches dans un champ agricole. Cette expérimentation a été réalisée par Intel Research Laboratory and Agriculture and Agri-Food Canada sur une vigne à British Columbia.

### d. Applications médicales

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers à une étape précoce, ..). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, ... à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, ...) chez les personnes dépendantes (handicapées ou âgées). En outre, des chercheurs américains ont implanté des capteurs à l'échelle nano dans le sang pour détecter quelques maladies à une étape précoce.

Comme il existe d'autres applications telles que celles qui permettent la surveillance des lieux contre les intrusions ou les risques. Par exemple, les capteurs placés dans les conduites permettant de transporter le gaz.

### **I.3 L'architecture des RCSF**

Puisque les RCSF se caractérisent par l'absence d'une infrastructure déterminée au préalable, les nœuds capteurs la construisent tout en permettant l'interaction avec l'environnement où ils appartiennent et en répondant aux différentes requêtes venant des utilisateurs ou des réseaux externes. Par ailleurs, les nœuds capteurs comme tout autre composant de télécommunication adhèrent à une architecture protocolaire spécifique. La réalisation de cette dernière requiert la mise en œuvre de techniques développées pour les réseaux ad hoc. Cependant, de nouveaux problèmes apparaissent engendrés entre autre par la sévérité des contraintes dues aux limitations de ressources physiques des RCSF. C'est la raison pour laquelle, il est commode que la conception des protocoles de communication soit faite d'une manière optimale.

#### **I.3.1 Architecture de communication**

Un réseau des capteurs est composé d'un grand nombre des nœuds dispersés dans une zone géographique. Le nombre des capteurs déployés est défini selon la taille de la zone de couverture et chaque nœud a la possibilité de détecter ou collecter certaine information (température, humidité, ...) et de la transmettre aux voisins jusqu'à l'aboutissement à la station de base ou nœud puits [5]. La distance entre deux nœuds est estimée en fonction de la puissance de transmission via les ondes radio.

Le nœud puits est un nœud qui n'est pas comme les autres en termes d'énergie. Son but est de recevoir les informations qui y arrivent des autres nœuds et les transmettre à son tour à un poste de contrôle par le biais d'une transmission satellitaire ou Internet. La figure I.2 illustre un schéma de communication de données dans un RCSF.

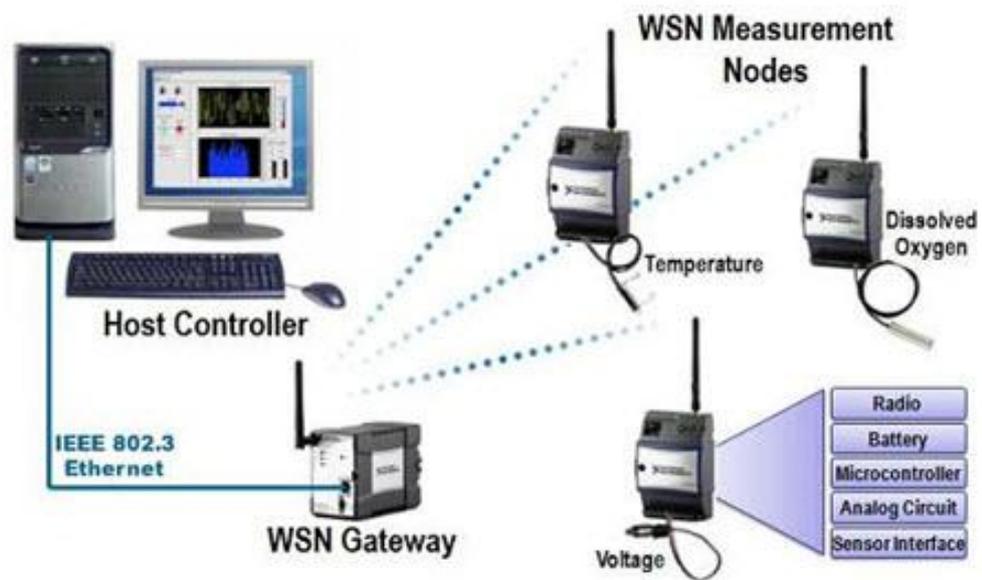


Figure I.2: Architecture de communication []

### I.3.2 Collecte d'informations

Il y a deux méthodes pour collecter les informations dans un réseau de capteurs.

- **Collecte d'informations à la demande** : Lorsque l'on souhaite connaître l'état de la zone d'intérêt à l'instant  $t$ , le nœud puits émet une requête en broadcast vers tous les nœuds déployés dans la zone d'intérêt pour que ces derniers remontent leur dernier relevé vers le nœud puits. Les informations sont alors acheminées vers le nœud puits par le biais d'une communication multi-sauts. La figure I.3 illustre ce mode de communication [13].

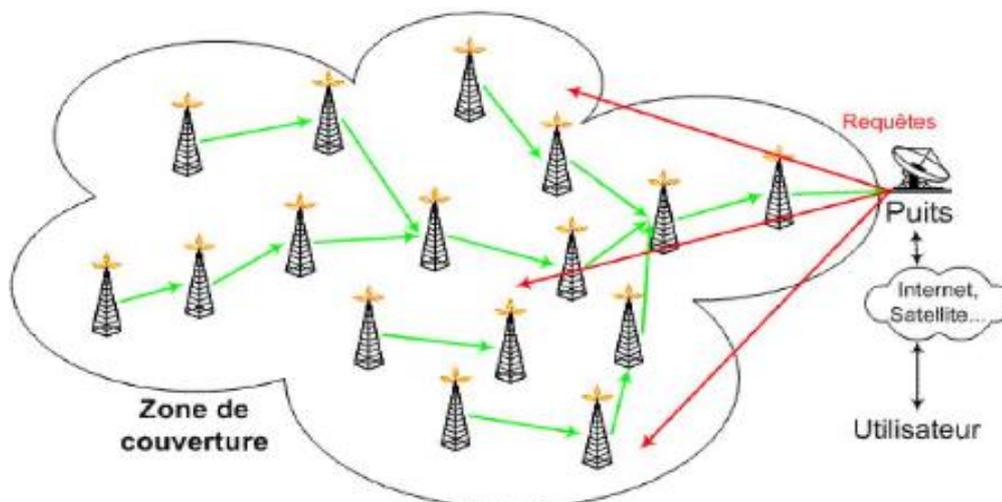


Figure I.3: Collecte d'informations à la demande [13]

- **Suite à un événement** : Un événement se produit en un point de la zone d'intérêt (changement brusque de température, détection d'un mouvement...), les capteurs situés à proximité de ce point cible remontent alors les informations relevées et les acheminent jusqu'au nœud puits. La figure I.4 symbolise comment se fait la communication de l'information au centre de contrôle dans ce type de réseaux [13].

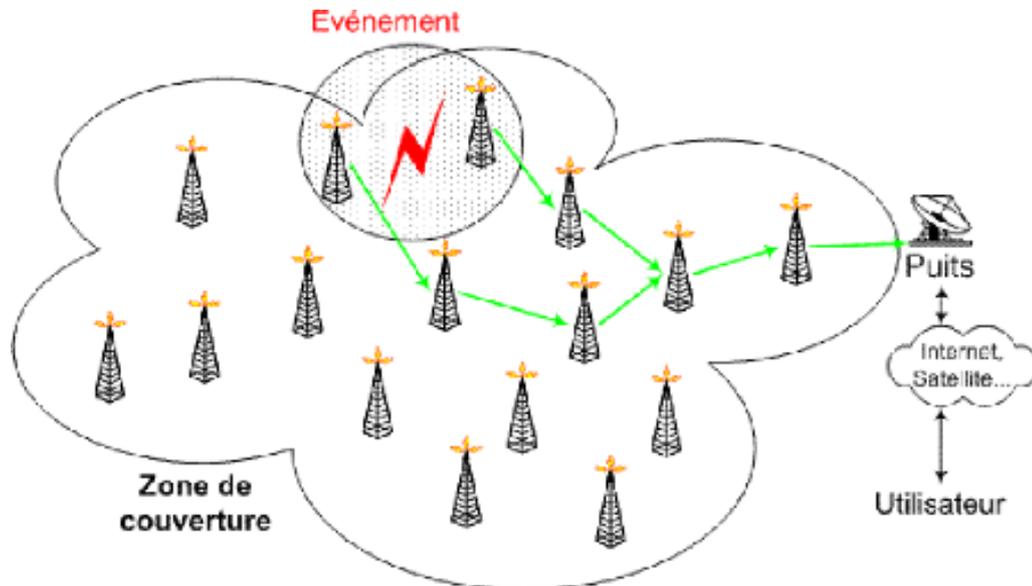


Figure I.4: Application orientée événement [13]

### I.3.3 Architecture protocolaire [14]

Dans le but d'un établissement efficace d'un RCSF, une architecture en couches est adoptée afin d'améliorer la robustesse du réseau. Une pile protocolaire de cinq couches est donc utilisée par les nœuds du réseau : la couche application, la couche transport, la couche réseau, la couche liaison de données et la couche physique.

De plus, cette pile possède trois plans de gestion: le plan de gestion des tâches qui permet de bien affecter les tâches aux nœuds capteurs, le plan de gestion de mobilité qui garde une image sur la localisation des nœuds pendant la phase de routage, et le plan de gestion de l'énergie qui permet de préserver le maximum d'énergie.

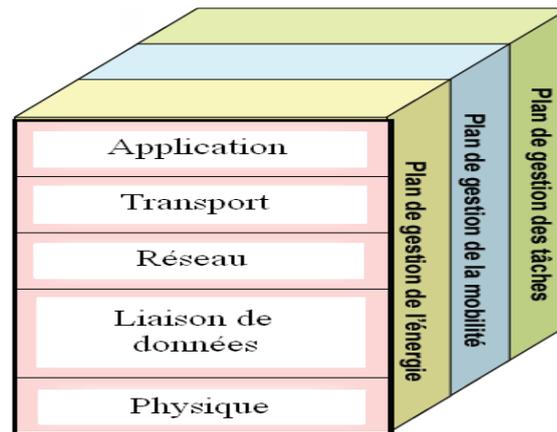


Figure I.5: Pile protocolaire dans les RCSF [14]

- **Couche application** : Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs, gérée directement par les logiciels. Parmi les protocoles d'application, nous citons : SMP et TADAP.
- **Couche transport** : Elle vérifie le bon acheminement des données et la qualité de la transmission. Dans les RCSF, la fiabilité de transmission n'est pas garantie. Ainsi, les erreurs et les pertes sont tolérées c'est la raison pour laquelle nous déployons les capteurs en grand nombre. Par conséquent, un protocole de transport proche du protocole UDP appelé UDP-Like est utilisé. Cependant, comme le protocole de transport TCP est universel, les RCSF doivent donc posséder, lors d'une communication avec un réseau externe, une interface TCP-splitting pour vérifier la compatibilité entre ces deux réseaux communicants.
- **Couche réseau** : Elle s'occupe du routage de données fournies par la couche transport. Elle établit les routes entre les nœuds capteurs et le nœud puits, et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc. Les protocoles de routage conçus pour les RCSF sont différents de ceux conçus pour les réseaux ad hoc puisque les RCSF sont différents selon plusieurs critères comme :
  - l'absence d'adressage fixe des nœuds tout en utilisant un adressage basé-attribut.
  - l'établissement des communications multi-sauts.
  - l'établissement des routes liant plusieurs sources à une seule destination pour agréger des données similaires, etc.
 Parmi ces protocoles, nous citons : LEACH et SAR.

- **Couche liaison de données** : Elle est responsable de l'accès au media physique, de la détection et de la correction d'erreurs intervenues sur la couche physique. De plus, elle établit une communication saut-par-saut entre les nœuds i.e. elle détermine les liens de communication entre eux dans une distance d'un seul saut. Parmi les protocoles de liaison de données, nous citons : SMACS (Self-organizing Medium Access Control for Sensor networks) et EAR (Eavesdrop And Register).
- **Couche physique** : Cette couche permet de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences.

## I.4 Routage dans les RCSF

Dans les RCSF, les capteurs sont déployés en grand nombre pour surveiller un tel phénomène et faire remonter l'information à un centre de contrôle distant. Pour atteindre cette finalité, les capteurs ont la capacité de communiquer et collaborer entre eux pour acheminer l'information collectée à la station de base en garantissant sa fiabilité et en empruntant le plus court chemin entre le nœud qui a détecté ce phénomène et la station de base. Cette opération permet le routage de l'information entre le nœud détecteur et le nœud puis et elle consiste à trouver les routes les plus courtes. Dans cette optique, plusieurs protocoles de routage ont été proposés dans la littérature.

Les contraintes présentées dans les RCSF ont données naissance à des protocoles de routage différents que ceux des autres réseaux sans fil puisque la contrainte énergétique se pose avec force dans les RCSF. De ce fait, les protocoles de routage conçus pour les RCSF doivent garantir l'acheminement de l'information entre tout nœud du réseau et la station de base à moindre coût en termes d'énergie.

Dans ce que suit, nous présentons quelques approches et techniques sur lesquelles se basent les protocoles de routage dans les réseaux sans fil et en particulier dans les réseaux de capteurs.

### I.4.1 Approches d'établissement de routes

#### a. Approche basée sur l'état du lien

Chaque nœud de réseau maintient une vue globale de la topologie du réseau qui lui permet de calculer les chemins vers toutes les destinations. Les nœuds diffusent

périodiquement (par inondation) l'état des liens avec leurs voisins à tous les nœuds du réseau. La diffusion peut aussi être produite lorsqu'un des nœuds détecte un nouveau lien vers un de ses voisins corrompus ou lorsqu'une mise dans sa table de topologie est faite. Le problème de cette approche, les messages de contrôle surchargent le réseau et surtout quand il s'agit d'un réseau avec un grand nombre des nœuds.

#### **b. Approche basée sur le vecteur de distance**

Dans cette approche, chaque nœud transmet à ses voisins la distance en termes de nombre de sauts qui le sépare de chaque destination dans le réseau et le nœud voisin utilisé pour atteindre cette destination. Donc, un nœud établit les routes vers les autres destinations en se basant sur les informations reçues depuis tous ses voisins. Ce dernier calcule le chemin le plus court vers n'importe quelle destination dans le réseau. Si la distance séparant deux nœuds change on répète le processus de calcul.

L'approche du vecteur des distances évite l'inondation, mais elle est moins précise que l'approche basée sur l'état du lien puisqu'il est aussi difficile de trouver des routes alternatives en cas de coupure de routes.

### **I.4.2 Protocoles basés sur l'architecture réseau**

Dans les réseaux sans fil, nous distinguons deux catégories de protocoles de routages : hiérarchiques et non hiérarchiques (ou à plat).

#### **a. Les protocoles de routage à plat**

Ces protocoles de routage sont des protocoles non hiérarchiques dans lesquels les nœuds jouent le même rôle et ont la même tâche. L'acheminement des paquets de la part d'un nœud dépend de sa position dans le réseau. Nous présentons dans ce qui suit quelques protocoles qui sont fondés sur cette architecture :

- **DSDV (Destination Sequenced Distance Vector) :** DSDV est un protocole proactif de routage basé sur l'approche du vecteur à distance et dans lequel les nœuds sont déployés aléatoirement. Dans DSDV, chaque nœud possède une table de routage contenant le saut suivant et le nombre de sauts vers une destination bien précise. La table de routage est actualisée périodiquement à tout moment. En outre, pour éviter les bouclages DSDV attribue un numéro de séquence pour chaque route. Ainsi la route qui dispose d'un numéro de séquence plus grand, sera favorisée [1].

- **DSR (Dynamic Source Routing)** : DSR est un protocole de routage réactif qui utilise la technique de routage source. Cette technique permet à la source de déterminer le chemin ou bien la séquence complète des nœuds parcourus pour atteindre une destination [8].
- **OLSR (Optimized Link State Routing)** : OLSR est un protocole proactif basé sur des relais multipoints (MPR). MPR consiste essentiellement, pour un nœud donné de ne pas prendre en considération tous les liens vers les voisins mais seulement une partie dans laquelle ces nœuds peuvent atteindre tous les voisins des voisins i.e. les voisins à deux sauts du nœud source (2-voisins). Cette technique permet aux nœuds de calculer les plus courts chemins. Après qu'un nœud choisit un ensemble de nœuds relais, il annonce aux réseaux qu'il est capable d'atteindre les nœuds élus dans l'ensemble de leur MPR [1].

### I.4.3 Les protocoles de routage hiérarchiques

Dans les RCSF, la majorité des protocoles de routage sont des protocoles hiérarchiques. Ces protocoles sont conçus dans le but de réduire la consommation d'énergie lors de l'établissement de routes et surtout dans les réseaux à grande échelle.

Le routage hiérarchique permet de partitionner le réseau en sous ensemble pour faciliter la gestion du réseau et préserver par la suite l'énergie. Le partitionnement du réseau permet d'organiser le réseau de telle sorte chaque sous ensemble de nœuds a un super nœud appelé cluster-head. Ce dernier gère cet ensemble de nœuds et communique avec d'autres cluster-heads.

Nous présentons des exemples de protocoles de routage hiérarchiques :

- **HEED (Hybrid, Energy-Efficient, Distributed approach)** : HEED est un protocole de routage conçu pour les RCSF. Il est basé sur un algorithme de clustering distribué. HEED ne dépend pas de la topologie du réseau ni de sa taille mais il suppose que les capteurs sont intelligents i.e. ont la possibilité de modifier leur puissance de transmission. HEED construit les cluster-heads selon un critère hybride regroupant le degré des nœuds et leurs énergies restantes. Il vise à réaliser une distribution uniforme des clusters dans le réseau et à générer des clusters équilibrés en taille. Dans HEED, un nœud  $u$  est élu comme cluster head avec une probabilité  $P_{CH}$  égale à :

$$P_{CH} = C_{prob} \frac{E_u}{E_{Total}}$$

où  $E_u$  est l'énergie restante du nœud  $u$ ,  $E_{Total}$  l'énergie globale dans le réseau et  $C_{prob}$  est le nombre optimal de clusters.

Dans HEED, les cluster-heads communiquent directement avec la station de base ce qui permet de minimiser la consommation d'énergie. Cependant, dans HEED, la topologie en clusters ne réalise pas la consommation minimale d'énergie dans les communications intracluster et les clusters générés ne sont pas tellement équilibrés en taille.

- **LEACH (Low Energy Adaptive Clustering Hierarchy):** LEACH est basé sur un algorithme de clustering distribué. C'est un protocole conçu pour le routage dans les réseaux de capteurs homogènes où les capteurs ont les mêmes caractéristiques et les mêmes capacités. Dans LEACH, les cluster-heads sont choisis aléatoirement et communiquent directement avec la station de base ce qui permet de minimiser la consommation et réduire la quantité d'informations envoyées à la station de base.
- **PEGASIS (Power-Efficient Gathering in Sensor Information Systems) :** PEGASIS est une version améliorée du protocole LEACH. Ce protocole forme une chaîne entre les nœuds de sorte que les nœuds reçoivent et communiquent juste des données des nœuds voisins. Les données collectées sont transmises d'un nœud à un autre jusqu'à l'arrivée à un nœud particulier qui va à son tour les transmettre vers la station de base. Les nœuds particuliers sont choisis à tour de rôle selon la politique Round-robin pour le but d'économiser l'énergie. Contrairement au protocole LEACH, PEGASIS ne fixe pas le rôle à un seul nœud mais ils affectent d'autres rôles en fonction de leurs énergies restantes. En plus, PEGASIS suppose que les nœuds peuvent modifier leur puissance de transmission [1].

## I.5 Les limites des protocoles de routage dans les RCSF

La plupart des protocoles de routage proposés pour les réseaux sans fil en particulier pour les RCSF, ont été conçus pour des réseaux de taille modeste. Cependant, lorsque le nombre de nœuds augmente les performances de ces protocoles commencent à se dégrader et on parle de la contrainte de passage à l'échelle.

Le passage à l'échelle est fait de mettre en place un grand nombre de nœuds dans une zone de couverture. Or, quand le nombre de nœuds augmente cela pourra avoir

une influence sur les performances du réseau en termes de latence, etc. En outre, la majorité des protocoles de routage ont été développés pour des réseaux de taille modeste et nous avons remarqué que lorsque le nombre de nœuds augmente les performances de ces protocoles se dégradent grandement.

Dans les applications orientées événements (Event-Driven), le temps nécessaire pour remonter une information au centre de contrôle doit être réduit pour que l'équipe d'intervention arrive à temps et minimise l'occurrence d'une catastrophe. Cependant, nous avons remarqué que lorsque le nombre de nœuds augmente dans certains protocoles le temps pour envoyer l'information à la station de base devient assez grand et par suite l'intervention dans certains cas devient utile. De ce fait, les protocoles doivent garantir un temps réduit pour alerter le centre de contrôle. Pour se faire, nous avons proposé d'utiliser des heuristiques pour réduire ce temps dans les réseaux denses.

## **I.6 Conclusion**

Dans les réseaux sans fil, plusieurs protocoles de routage ont été conçus et ces protocoles donnent de bons résultats en termes de temps nécessaire pour envoyer une information à la station de base et d'énergie dissipée pour les réseaux de taille modeste. Cependant, ces protocoles perdent leurs performances lorsque nous passons à l'échelle puisqu'ils utilisent des versions itératives pour établir les routes entre les nœuds et la station de base.

Pour montrer le handicap présenté par la version itérative dans les protocoles de routage, nous avons analysé GSR qui est un protocole de routage conçu pour les réseaux ad hoc. GSR fournit de bonnes performances dans le cas des réseaux de petite taille. Nous avons modifié ce protocole de telle sorte qu'il soit adapté aux réseaux de capteurs. Puis, nous avons développé un protocole basé sur une heuristique pour réduire le temps d'établissement de routes entre tous les nœuds et la station de base.

Dans le chapitre suivant, nous présentons le protocole GSR et sa version simplifiée qui est adaptée aux réseaux de capteurs.

# Chapitre II

## Présentation du protocole GSR

### II.1 Introduction

Le protocole GSR est un protocole de routage conçu pour les réseaux ad-hoc. Il appartient à la famille des protocoles basés sur l'état de liens dans laquelle chaque nœud a une vue globale sur la topologie du réseau. Ce protocole a une approche déterministe qui utilise Dijkstra qui est un algorithme itératif pour l'établissement des plus courts chemins entre les nœuds.

GSR prend en compte la mobilité des nœuds pour construire les routes entre les nœuds. Dans notre contexte, nous ignorons le caractère mobilité des nœuds et nous mettons le point seulement sur la partie établissement de routes entre les nœuds.

### II.2 Modèle de réseau [10]

Pour illustrer le fonctionnement de GSR, nous le modélisons tout d'abord en utilisant le modèle du disque unitaire dans lequel si la distance séparant deux nœuds est inférieure ou égale à la portée de transmission, ces deux derniers sont considérés comme des voisins.

Un réseau sans fil peut être modélisé par un graphe non orienté  $G = (V, E)$ , où  $V$  est un ensemble de nœuds et  $E$  est un ensemble de arêtes reliant les nœuds dans  $V$ . Chaque nœud dispose d'un identifiant unique et représente un hôte mobile mené d'un dispositif de communication sans fil dont la portée de transmission est  $R$ , et un espace de mémoire de l'infini. Les nœuds peuvent se déplacer et changer leur vitesse et leur direction indépendante. Un lien non orienté  $(i, j)$  reliant deux nœuds  $i$  et  $j$  est formé lorsque la distance entre  $i$  et  $j$  est inférieure ou égale à  $R$ . La référence  $(i, j)$  est supprimée de  $E$  lorsque la distance séparant les nœuds  $i$  et  $j$  devient supérieur à la gamme de transmission  $R$ .

Pour chaque nœud  $i$ , une liste et trois tables sont attribuées : une liste des voisins  $A_i$ , une table de topologie  $TT_i$ , une table du prochain saut  $Next_i$  et une table des

distances  $D_i$ .  $A_i$  est définie comme un ensemble de nœuds qui sont adjacents au nœud  $i$  comme montre l'équation (1).

$$A_i = \{j \in V : dist(i, j) \leq R\} \quad (1)$$

Chaque destination  $j$  possède une entrée dans la table  $TT_i$  qui contient deux parties :  $LS(j)$  et  $SEQ(j)$ .  $LS(j)$  désigne l'information d'état de lien indiqué par le nœud  $j$ , et  $SEQ(j)$  désigne l'horodatage indiquant le temps de mise à jour de cette information d'état de liaison. De même, pour chaque destination  $j$ ,  $Next_i(j)$  désigne le prochain saut du nœud  $i$  pour atteindre le nœud  $j$  en empruntant le plus court chemin de  $i$  vers  $j$ . Tandis que  $D_i(j)$  désigne la distance du plus court chemin de  $i$  à  $j$ .

En outre, une fonction de coût est utilisée pour calculer la distance entre les nœuds. S'il existe un lien direct entre deux nœuds alors cette fonction retourne la valeur 1 sinon elle retourne l'infinie et suppose que le nœud est inaccessible à partir du nœud d'origine.

### II.3 Présentation de GSR [10]

GSR s'exécute selon la démarche suivante : au début, chaque nœud  $i$  initialise la liste de ses voisins  $A_i$  par une liste vide, sa table de topologie  $TT_i$  par une table vide. Puis le nœud  $i$  initialise ses variables locales comme s'est décrit dans la procédure *NodeInit(i)* et il écoute les paquets reçus dans son voisinage pour établir la liste de ses voisins. Le nœud  $i$  invoque par la suite *PktProcess(i)* pour connaître les messages de routage qui contiennent les informations sur l'état de lien de chaque nœud voisin. Le nœud compare le numéro de séquence de la route pour une éventuelle mise à jour de la route. Ceci est fait pour chaque destination dans le réseau. Si le message de découverte de route contient un nouveau numéro de séquence pour une route vers une telle destination ( $j$ ), le nœud mettra à jour le numéro de séquence. De ce fait, il remplace respectivement les contenus de  $TT_i.LS(j)$  par  $pkt.LS(j)$  et  $TT_i.SEQ(j)$  par  $pkt.SEQ(j)$ . Après avoir examiné les messages de routage, le nœud construit sa table de routage en se basant sur la nouvelle table de topologie et il diffuse ses informations dans son voisinage.

### II.3.1 Schéma algorithmique de GSR [10]

La figure II.1 illustre l'algorithme par chaque nœud pour construire sa liste de voisins, la table de topologie et la table de routage vers les autres destinations.

```

proc Node(i) ≡
  NodeInit(i);
  while TRUE do
    if PktQueue ≠ ∅  !! packet received
      foreach pkt ∈ PktQueue do
         $A_i \leftarrow A_i \cup \{pkt.source\}$ 
        PktProcess(i, pkt)
      od;
    fi
    FindSP(i);
    if (clock() mod UpdateInterval) = 0
      RoutingUpdate(i);
    fi
    CheckNeighbors(i);
     $TT_i.LS(i) \leftarrow A_i$ ;
  od

```

Figure II.1: Algorithme GSR [10]

Les éléments de cet algorithme sont :

- *i* : l'identifiant du nœud.
- NodeInit(*i*) : la procédure qui initialise les données du capteur *i*. Cette initialisation concerne la table de topologie, la liste des voisins, la table des distances vers les nœuds de destination, le nœud pour le prochain saut vers une destination (Next) et le numéro de séquence de chaque entrée de la table des distances.
- PktQueue : le paquet reçu contient les données et l'identifiant du nœud source. Les données sont la table de topologie et la table des liaisons du nœud source.
- PktProcess(*i*,pkt) : pour chaque paquet reçu le nœud *i* vérifie les entrées de la table de topologie reçue avec celles sa table de topologie. Si l'identifiant du nœud source est différent à l'identifiant du nœud *i* et le numéro de séquence vers une destination de sa table de topologie est différent à celui contenu dans sa table de topologie alors il met à jour le numéro de séquence de la route et l'état de la liaison.

- FindSP(i) : le nœud i exécute l'algorithme de Dijkstra pour calculer les plus courts chemins vers toutes les autres destinations.
- Clock() : c'est un laps de temps qui est défini à chaque fois que le nœud i exécute RoutingUpdate.
- RoutingUpdate(i) : le nœud i construit un message contenant leur identifiants, les tables de numéro de séquence et la table de liaison.
- CheckNeighbors(i) : cette procédure s'exécute pour vérifier les liaisons avec les voisins du nœud i. Si un lien vers un des voisins est rompu donc le nœud i supprime ce voisin de sa liste de voisins.

### II.3.2 Diffusion de l'information [10]

La principale différence entre GSR et LS traditionnels est la manière dont les informations de routage sont diffusées. En LS, les paquets d'état de liaison sont générés et inondés dans le réseau à chaque fois qu'un nœud détecte les changements de topologie alors que GSR n'inonde pas les paquets d'état de liaison de cette manière. Dans GSR, les nœuds maintiennent la table d'état des liens en fonction de la place des informations reçues des nœuds voisins, et échangent cette table régulièrement avec leurs voisins locaux seulement. L'information est diffusée en tenant compte du numéro de séquence de l'état de liaison. Ainsi les numéros de séquence les plus grands remplacent ceux qui ont des numéros de séquence plus petits. À cet égard, il est semblable à DSDV où la valeur des distances est remplacée selon l'horodatage du numéro de séquence.

### II.3.3 Le plus court chemin dans GSR [10]

Dans GSR, la détermination du plus court chemin se fait par la procédure FindSP(i). Cette procédure crée une arborescence des plus courts chemins enracinés en i vers tous les nœuds du réseau. Pour cela, il applique l'algorithme de Dijkstra pour déterminer le plus court chemin entre un couple de nœuds i et j. L'exécution de Dijkstra se fait avec quelques modifications de sorte que la table de saut suivant ( $Next_i$ ) sera établie et les tables de distance ( $D_i$ ) sont calculées en parallèle avec la reconstruction de l'arbre.

### II.3.4 Points forts et limitations de GSR

Dans GSR, la mise à jour se fait lors de l'occurrence d'un événement qui cause un changement de la topologie. Il permet aussi de réduire considérablement l'overhead lors de la construction de routes. Cependant, GSR épuise la bande passante puisque lors d'une mise à jour toute la table de topologie sera diffusée et le temps nécessaire pour l'état de lien dépend de l'intervalle de mise à jour.

## II.4 Analyse de la complexité de GSR [10]

Dans cette section, nous analysons la complexité du protocole GSR et le comparons avec deux autres schémas de routage : DBF (présenté en annexe) et LS. La complexité est étudiée sous cinq aspects :

1. Complexité de calcul (CC) : le nombre d'étapes de calcul d'un nœud pour effectuer un calcul de route après la réception d'un message de mise à jour.
2. La complexité de la mémoire (MC) : l'espace mémoire nécessaire pour stocker les informations de routage.
3. La complexité de données (DC) : la taille totale des paquets de contrôle échangés par un nœud dans chaque intervalle de temps (time slot).
4. Complexité du paquet (PC) : le nombre moyen de paquets de routage échangés par un nœud dans chaque intervalle de temps.
5. Le temps de convergence (CT) : le temps nécessaire pour détecter un changement de la liaison.

**Tableau II-1 : Comparaison des complexités de GSR, LS et DBF [10]**

| Protocol | CC       | MC             | DC         | PC     | CT             |
|----------|----------|----------------|------------|--------|----------------|
| GSR      | $O(N^2)$ | $O(N \cdot d)$ | $O( E )/I$ | $O(1)$ | $O(D \cdot I)$ |
| LS       | $O(N^2)$ | $O(N^2)$       | $O( E )/I$ | $O(N)$ | $O(D)$         |
| DBF      | $O(N)$   | $O(N)$         | $O(N)/I$   | $O(1)$ | $O(N \cdot I)$ |

Le tableau II-1 présente les résultats de la comparaison des complexités. Dans ce tableau,  $N$  représente le nombre de nœuds de réseau ( $|V|$ ),  $D$  désigne le diamètre du réseau i.e. la distance maximale en nombre de sauts dans le réseau entre les nœuds,  $d$  et

$I$  désignent respectivement le degré de connectivité des nœuds et l'intervalle de mise à jour de routage.

GSR et LS ont les mêmes complexités en termes de calcul et de mémoire pour la maintenance de la topologie de tout le réseau et pour le calcul des plus courts chemins en utilisant l'algorithme de Dijkstra. L'algorithme de Dijkstra nécessite généralement  $O(N^2)$  étapes pour calculer les plus courts chemins d'une source à toutes les destinations, même s'il est possible de le réduire à  $O(n \log(n))$ .  $O(N^2)$  c'est l'espace de mémoire exigé pour mémoriser la topologie du réseau représentée par une matrice de connexion. Comme pour DBF, il a une complexité de  $O(N)$  pour le calcul et la mémoire, puisqu'il ne conserve que les informations de distance pour chaque destination, et calcule les plus courts chemins dans un modèle distribué.

## II.5 Conclusion

Dans ce chapitre, nous avons présenté le protocole GSR et l'analyse de sa complexité comparée à d'autres protocoles. Cette étude de GSR nous a permis de tirer profit des bonnes performances fournies par GSR. C'est la raison pour laquelle, on a choisi ce protocole pour l'adapter aux réseaux de capteurs dans une architecture à plats.

Dans le chapitre suivant, nous proposons une version améliorée et simplifiée de ce GSR et un algorithme de routage basé sur l'heuristique du recuit simulé.

## Chapitre III

# Implémentation de GSR modifié

### III.1 Introduction

Les RCSF et les réseaux ad-hoc ont de nombreux points communs : réseaux sans infrastructure, architecture décentralisée, autonomie, utilisation des ondes radio pour communiquer. En outre, les RCSF dédiés pour les applications orientées événements visent à minimiser le temps pour remonter l'information collectée à la station de base. Dans cette optique, nous avons essayé d'optimiser le temps de construction de routes entre tout nœud et la station de base.

Dans ce chapitre, nous allons implémenter une version modifiée de GSR qui est adaptée aux RCSF, appelée GSR\_M. Dans cette version, nous intéressons à établir des routes entre tout nœud du réseau et un nœud bien particulier qui est la station de base. A cet effet, le nœud source est un nœud quelconque alors que le nœud de destination est représenté par la station de base. D'où, nous n'avons pas besoin de mémoriser la trace vers la station de base mais seulement du nœud du prochain saut dans la direction de la station de base.

Au début nous développons la version adaptée aux RCSF. Puis, nous évaluons les performances de GSR\_M lorsque nous passons à l'échelle dans un réseau. Dans GSR\_M, c'est l'algorithme de Dijkstra qui est appliqué pour calculer les coûts optimaux des nœuds vers la station de base. Puis nous proposons un algorithme de routage basé sur l'heuristique recuit simulé.

### III.2 Développement de GSR\_M

Dans la version originale du protocole de routage GSR, chaque nœud contient une liste des voisins, une table des destinations, et une table Next\_hop dans laquelle est mentionné le prochain nœud pour chaque destination.

- Liste des voisins : est une liste qui contient tous les identifiants (ID) des voisins du nœud courant et les coûts associés à ces voisins.
- Table de destination : contient les distances entre le nœud courant vers tous les autres

nœuds du réseau.

- Enregistrement suivant (next hop) : contient ID du nœud suivant avec le coût.

Le protocole GSR s'exécute en trois étapes :

- a) Première étape : au cours de cette étape, nous assistons à l'initialisation de chaque nœud. Les nœuds sont déployés aléatoirement dans une zone d'intérêt et ils sont repérés par les coordonnées  $(x,y)$ . Ces coordonnées nous permettent de calculer les distances euclidiennes entre les nœuds. Soient A et B qui ont respectivement les coordonnées suivantes  $(x_a, y_a)$  et  $(x_b, y_b)$  alors la distance entre ses deux nœuds est calculée par l'équation (1).

$$dist(A, B) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (1)$$

Dans GSR, les nœuds ont la même portée de transmission et de réception (R). D'où si la distance euclidienne entre deux nœuds A et B est inférieure ou égale à R alors ces deux nœuds sont considérés des voisins. Après la découverte des voisins par chaque nœud alors chacun d'eux établit la table des distances de la manière suivante : si le nœud est voisin alors cette entrée de la table est initialisée par  $dist(A, B)$  sinon par l'infini.

- b) Deuxième étape : au cours de cette étape, GSR fait appel à l'algorithme de Dijkstra par chaque nœud pour mettre à jour la table des distances.

Le schéma algorithmique est donné par la figure III.1.

**ALGO DIJKSTRA***données* :  $E, \Gamma, l, i \in E$ *Résultats* :  $\pi, S$  $S := \{i\}; \pi(i) := 0; k := 1; x_1 := i;$ **pour tout**  $x \in E \setminus \{i\}$  **faire**  $\pi(x) := \infty$  **fin pour tout****tant que**  $k < n$  et  $\pi(x_k) < \infty$  **faire****pour tout**  $y \in \Gamma(x_k)$  tel que  $y \notin S$  **faire** $\pi(y) := \text{Min}[\pi(y), \pi(x_k) + l(x_k, y)]$ **fin pour tout**extraire un sommet  $x \notin S$  tel que  $\pi(x) = \text{Min}[\pi(y); \forall y \notin S];$  $k := k + 1; x_k := x; S := S \cup \{x_k\}$ **fin tant que****FIN ALGO****Figure III.1: Algorithme de Dijkstra**

Après la mise de la table des distances, chaque nœud échange cette table avec ses voisins. Ainsi, chacun d'eux compare chacune des entrées de sa table des distances avec les autres tables des voisins et vérifie pour chaque destination le coût. Si le coût vers une destination  $j$  dans la table de distance d'un nœud est inférieur au le coût entre ce nœud et le nœud voisin plus le coût entre le voisin et la destination  $j$  aucune mise à jour ne sera effectuée au niveau de l'entrée correspondante sinon ce nœud modifie le coût vers la destination  $j$ . Le nouveau coût est égal au coût entre ce nœud et le nœud voisin plus le coût entre le voisin et la destination  $j$ .

Après l'exécution de l'algorithme Dijkstra les tables de distance des nœuds contiennent les coûts optimaux vers tous les nœuds destinations. Nous nous intéressons seulement à la station de base comme nœud de destination. Pour cela, chaque nœud parcourt sa table de distance et enregistre le coût tel que la destination est égale ID.

- c) Troisième partie : Dans GSR\_M, tout nœud a comme destination la station de base. Ainsi, il parcourt sa liste de voisins et choisit parmi eux le nœud suivant qui vérifie la condition suivante :

Le coût total du nœud considéré vers la station de base est égal au coût du nœud du voisin le plus proche à la station de base plus un.

Après ces trois étapes, chaque nœud a toutes les informations pour atteindre la station de base à moindre coût en passant par le voisin le plus proche de la station de base.

### III.3 Les heuristiques

Les heuristiques sont des stratégies qui permettent de guider la recherche d'une solution optimale.

On peut partager les heuristiques en deux grandes familles : les méta-heuristiques à solution unique et celles à solutions multiples ou dite population de solutions. La figure III.2 illustre la classification des heuristiques.

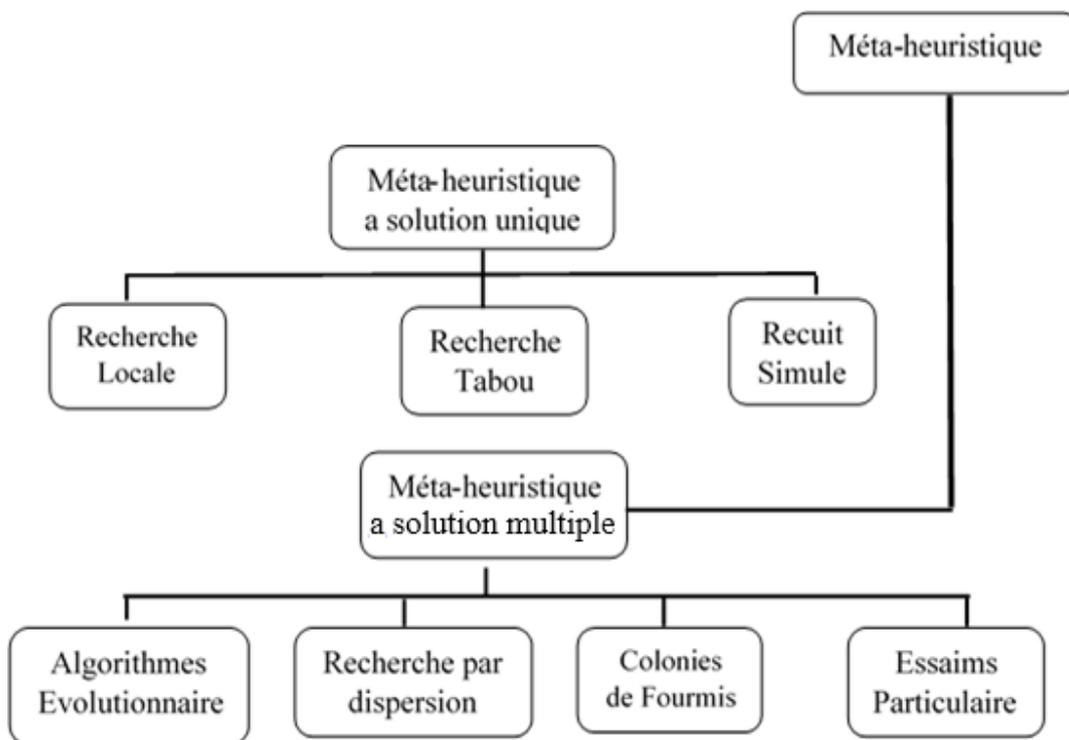


Figure III.2: Classification des méta-heuristiques [8]

### III.3.1 Heuristique à une solution unique

#### a. Recherche Locale

Cette méthode génère au début un ensemble de solutions initiales aléatoires par l'utilisation de l'algorithme de glouton aléatoire. Puis, l'algorithme choisit une prochaine solution de telle sorte qu'elle doit être meilleure en respectant le principe "le meilleur pour avancer".

La méthode de recherche locale la plus élémentaire est la méthode de descente. Elle peut être décrite comme suit [6] :

- a) Choisir une solution  $s \in S$ .
- b) Déterminer une solution  $s'$  qui minimise  $f$  dans  $N(s)$ .
- c) Si  $f(s') < f(s)$  alors poser  $s := s'$  et retourner à 2. Sinon STOP

#### b. Recherche Tabou

La recherche Tabou a été inventée par F.Glover [11] et a montré sa performance sur nombreux problèmes d'optimisation. Le principe de cette heuristique est le suivant : À partir d'une position donnée, explorer le voisinage et choisir la position dans ce voisinage qui minimise la fonction objectif.

#### c. Recuit Simule

C'est la première heuristique inventée dans la famille des méta-heuristiques nommée dans d'autre littérature méthode de Monte-Carlo.

L'expression "recuit simulé" est la traduction de Simulated Annealing. Dans la réalité, le terme "recuit" correspond à un réchauffement suivi d'un refroidissement lent. Un tel procédé est souvent utilisé dans l'industrie en particulier dans la métallurgie où les métaux chauffés sont refroidis rapidement ou lentement [7].

Le recuit simule est utilisé en métallurgie pour améliorer la qualité d'un solide.

### III.3.2 Heuristiques à solutions multiples

#### a. Recherche par dispersion

Cette heuristique a été proposée par Glover en 1977 [11]. Elle opère sur l'ensemble des points de références et elle consiste à créer un ensemble  $D_i$  des points dispersés à partir des points de références. Les points dispersés sont obtenus en

effectuant les combinaisons linéaires des points références. Ces combinaisons linéaires notées par  $C_i$  n'ont pas forcément un ensemble de résultats admissibles. On applique une procédure de réparation sur chaque point de  $C_i$  pour obtenir un ensemble  $A_i$  de points admissibles. Les points de  $A_i$  sont finalement optimisés à l'aide d'une Recherche locale pour obtenir l'ensemble  $D_i$  des points dispersés. Le nouvel ensemble  $R_{i+1}$  de points de références est obtenu en sélectionnant des points dans  $R_i \cup D_i$  [6].

#### **b. Colonie de fourmis**

L'heuristique de la colonie de fourmis est basée sur des expériences appliquées sur les colonies de fourmis. La première version de cette heuristique proposée dans la littérature est appelée "Ant System". Dans cette version, les auteurs ont visé notamment à résoudre le problème du voyageur de commerce dont le but est de trouver le plus court chemin permettant de relier un ensemble de villes.

### **III.3.3 Domaines d'utilisation des heuristiques**

Les chercheurs ont fait appel aux heuristiques pour traiter les problèmes dont la résolution se fait dans un temps qui n'est pas raisonnable. Par exemple, la mise en place d'un réseau sans fil au niveau d'une ville nécessite le parcours des milliards de combinaisons si on tient en compte la configuration des points d'accès et les points dans lesquels nous les plaçons parmi les sites potentiels [4]. De même pour le problème de routage dans les réseaux à grande échelle tels que les réseaux de capteurs où le nombre de combinaisons est très grand et le parcours de toutes ces combinaisons nécessite un temps non raisonnable.

Dans notre contexte, nous utilisons l'heuristique recuit simulé pour traiter le problème de routage dans les RCSF à grande échelle et qui sont conçus pour des applications orientées événements dont lesquelles où le temps est facteur précieux pour remonter l'information rapidement à la station de base.

### **III.4 Heuristique utilisée : Recuit simulé**

Le recuit simulé a des avantages par rapport les autres heuristiques en particulier à l'heuristique gloutonne puisqu'il accepte la dégradation de la solution. Cette caractéristique lui permet d'éviter le problème d'optimum local.

### III.4.1 Principes du recuit simulé

L'idée générale est déterminée un ensemble de chemins aléatoires vers la station de base à partir d'un nœud source. Parmi, cet ensemble de chemins nous choisissons un chemin aléatoire proche de la médiane. Ce chemin est représenté par un vecteur initial utilisé. Si la différence de coût entre le vecteur initial et le vecteur choisi est négative cela veut dire que le vecteur initial représente la meilleure solution. De ce fait, nous gardons cette solution au contraire si la différence est positive nous appliquons le nouveau vecteur comme vecteur initial et nous continuons la démarche de l'algorithme. En plus nous déterminons une variable  $T_0$  (température choisie au début). Cette variable est initialisée de la manière suivante :

- On génère un certain nombre de vecteurs  $X$  aléatoires à partir d'un nœud source,
- Pour chacun de ces vecteurs on détermine  $F(X)$ ,
- Ensuite on détermine la valeur médiane  $M$ , (la valeur qui partage la distribution en deux parties égales):

$$P = e^{-\frac{m}{T_0}} = 0,5$$

$$T_0 \approx 1,44 * M$$

où:

- $P$  : est la probabilité d'acceptation d'un chemin. Nous avons pris la valeur 0.5 car nous supposons que le chemin choisi est de longueur moyenne.
- $M$  : est la médiane des coûts de l'ensemble des chemins choisis

Au niveau de la décroissance du paramètre  $T$ , on peut prendre une décroissance géométrique de la forme :

$$T_{n+1} = R_T * T_n$$

$$T_n = T_0 * R_T^n$$

Pour le critère d'arrêt :

- On peut arrêter les itérations lorsque la température  $T_n$  est inférieure à une fraction donnée de la température initiale  $T_n < T_{Ratio} * T_0$ , avec  $T_{Ratio} = 10^{-6}$  par exemple.

On peut déterminer  $R_T$  :

$$R_T = T_{Ratio}^{1/N_T}$$

$$N_T = 100 \text{ (par exemple)}$$

$$R_T = (10^{-6})^{1/100} \approx 0,87$$

L'heuristique recuit simulé donne la meilleure solution parmi les solutions visitées. La figure III.3 illustre l'heuristique recuit simulé. Cet algorithme possède un vecteur initial  $X_0$  comme entrée et retourne un vecteur  $X$  qui représente le meilleur chemin entre le nœud source et la station de base.

- $F(X)$  : une fonction qui retourne le coût du chemin représenté par le vecteur  $X$ .
- $N_T$  : la cardinalité d'ensemble choisi aléatoirement au début.

```

Entrée :  $X_0$ 
Sortie :  $X$  qui minimise  $F$ 

DEBUT
   $X \leftarrow X_0$ 
   $T \leftarrow T_0$ 
  while Non condition d'arrêt do
     $m \leftarrow 0$ 
    repeat
       $Y \leftarrow \text{voisin}(X)$ 
       $dF \leftarrow F(Y) - F(X)$ 
      if Accepte( $dF, T$ ) then
         $X \leftarrow Y$ 
      end if
       $m \leftarrow m + 1$ 
    until  $m = N_T$  {nombre d'itérations maximum à la température  $T$ }
     $T \leftarrow \text{Decroissance}(T)$ 
  end while
FIN

```

Figure III.3: Algorithme du recuit simulé

La fonction Accepte reçoit comme paramètre  $dF$  qui représente la différence entre les coûts du deux vecteur  $X$  et  $Y$ , et  $T$  la température, elle retourne une variable booléen. La figure III.4 montre comment est exécutée la fonction  $dF$ .

```
Entrée :  $dF, T$ 

Résultat : VRAI si la fonction F décroît
           VRAI en fonction d'une certaine probabilité si F croît
           FAUX dans les autres cas

DEBUT
  if  $dF < 0$  then
    Retourner VRAI
  else
     $A \leftarrow e^{-\frac{dF}{T}}$ 
    if Alea(0,1) < A then
      Retourner VRAI
    else
      Retourner FAUX
    end if
  end if
FIN
```

Figure III.4: La fonction Accepte

### III.5 Evaluation des performances

Dans cette section, nous avons évalué les performances des deux propositions : GSR\_M et celle basée sur l'heuristique du recuit simulé. Puis, nous avons comparé les résultats obtenus pour le même scénario de déploiement.

- Le même nombre de nœuds
- La même topologie (doit être uniforme)
- La portée des nœuds est la même

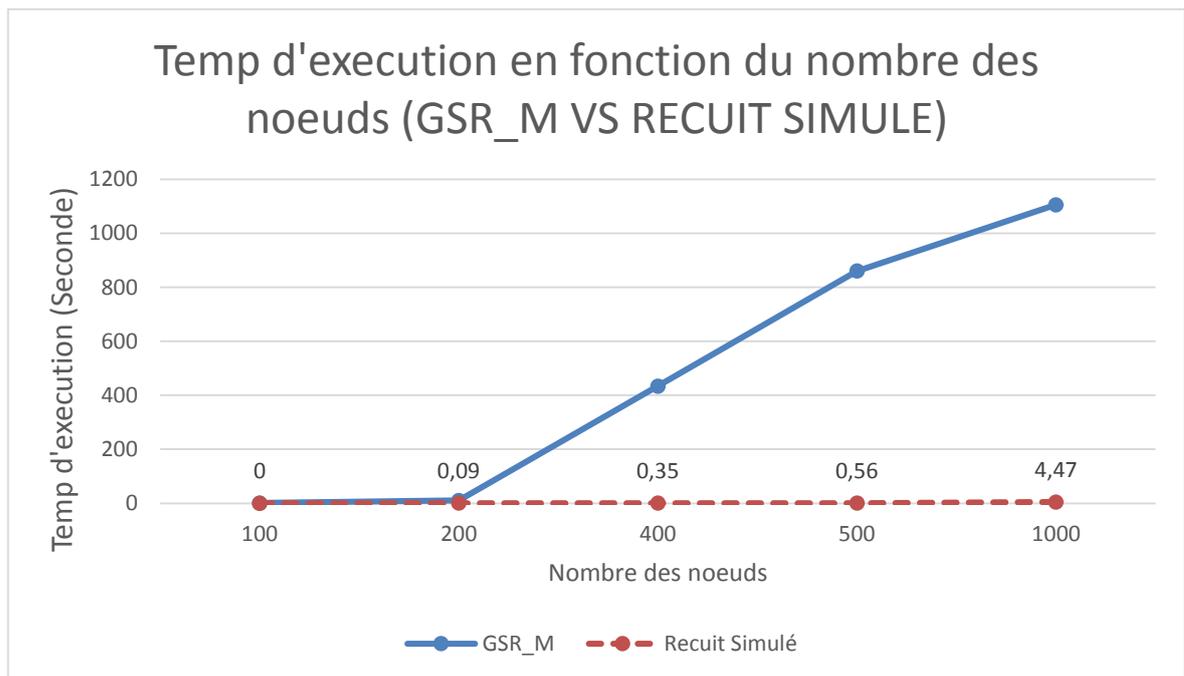
Pour montrer la robustesse de chacune des deux propositions, nous avons fait varier la taille du réseau et nous avons estimé respectivement le coût associé au chemin trouvé et le temps nécessaire pour trouver ce chemin.

#### III.5.1 Evaluation du temps d'exécution

Le tableau III-1 représente le résultat d'exécutions des deux approches en fonction de temps.

**Tableau III-1: Comparaison des temps d'exécution (GSR\_M vs. Recuit simulé)**

| Nombre des nœuds | GSR_M   | Recuit Simulé |
|------------------|---------|---------------|
| 100              | 0,9     | 0,05          |
| 200              | 10,62   | 0,09          |
| 400              | 433,51  | 0,35          |
| 500              | 859,88  | 0,56          |
| 1000             | 1105,53 | 4,47          |

**Figure III.5: Temps d'exécution (GSR\_M vs. Recuit simulé)**

Le tableau III.1 et les courbes représentées dans la figure III.5 montrent que le temps nécessaire pour déterminer le meilleur chemin entre un nœud et la station de base est largement inférieur quand la recherche est lancée par l'heuristique recuit simulé. Cela se justifie du fait que dans GSR\_M, tous les chemins sont visités alors que dans le recuit simulé le nombre de chemins visités est limité et dépend du nombre d'exécution par palier de température.

Sur la courbe représentant GSR\_M, nous constatons qu'au départ le temps d'exécution est 0.9 secondes pour un réseau de 100 nœuds mais une fois que la taille du réseau s'est doublé le temps d'exécution s'est multiplié par 11 (10.62 secondes). Cependant dans le cas du recuit simulé la différence entre les deux temps est minime.

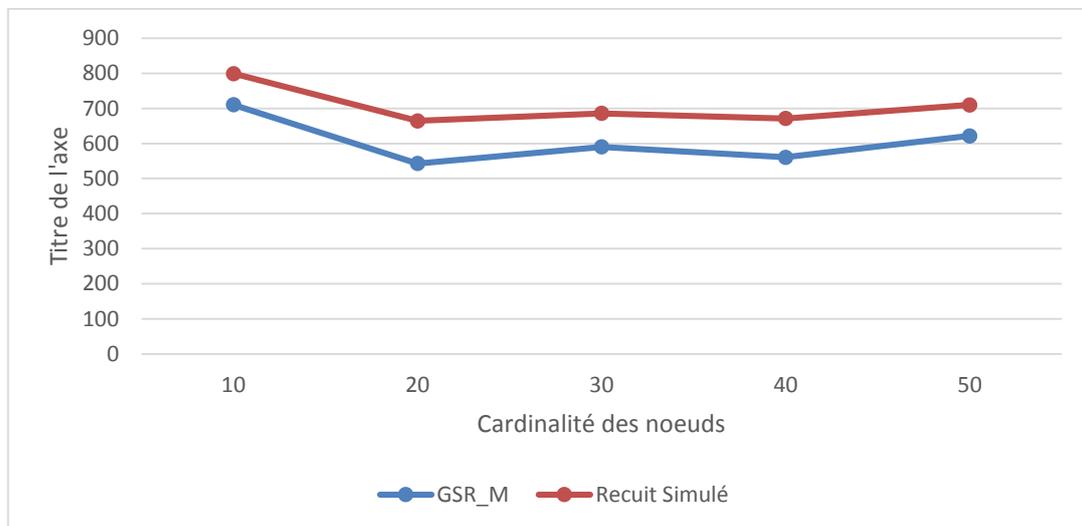
Nous avons de même estimé ce temps pour un réseau contenant 1000 nœuds ce temps est devenu 1105.53 secondes alors pour le recuit simulé est 4.47 secondes.

### III.5.2 Evaluation du coût moyen des routes

Le tableau III.2 représente le coût moyen calculé par les deux approches en fonction des cardinalités d'ensemble des nœuds. La taille du réseau dans ce contexte est 1000 nœuds.

**Tableau III-2 : Coût moyen (GSR\_M et Recuit simulé) vs. cardinalité**

| Cardinalité | Cout moyen |               |
|-------------|------------|---------------|
|             | GSR_M      | Recuit Simulé |
| 10          | 710,39     | 799,2         |
| 20          | 543,12     | 664,3         |
| 30          | 590,41     | 685,9         |
| 40          | 560,53     | 671,42        |
| 50          | 621,42     | 709,84        |



**Figure III.6: Coût moyen vs. cardinalité (1000 nœuds)**

Dans ces évaluations, le coût moyen représente la distance moyenne entre le nœud source et la base station. Ce coût est calculé en fonction de la cardinalité de l'ensemble des nœuds et les nœuds sources sont choisis d'une manière aléatoire.

Les résultats présentés dans le tableau et la figure montrent que GSR\_M fournit des résultats meilleurs puisqu'il calcule la longueur des chemins exactement alors que

dans l'approche recuit simulé s'éloigne un peu de la solution optimale fournie par GSR\_M.

### III.5.3 Evaluation du taux de précision

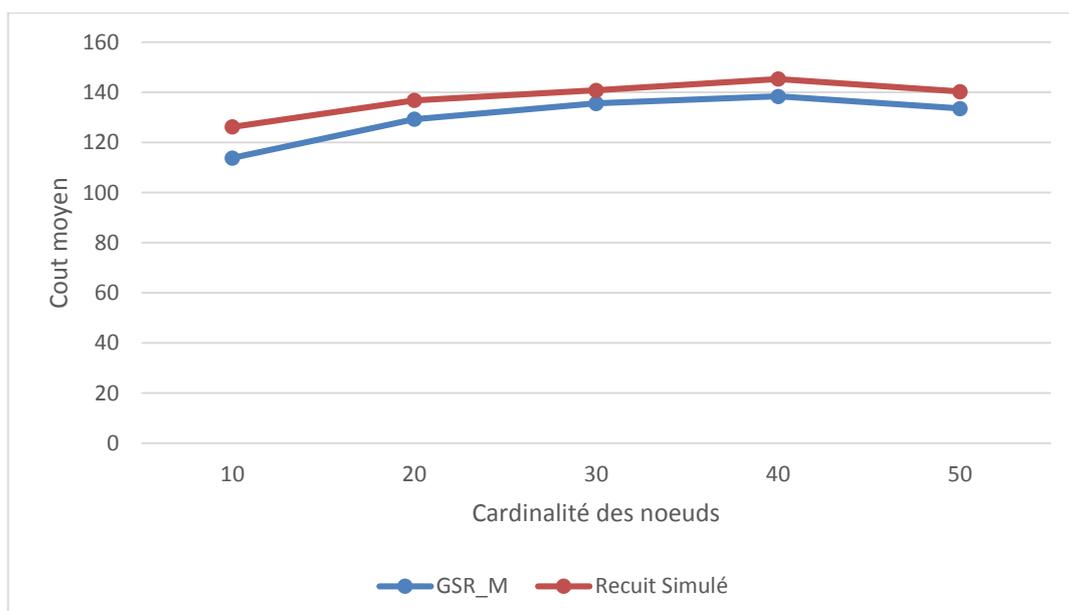
Pour connaître la qualité de la solution fournie par l'approche recuit simulé, nous avons évalué le taux de précision. Dans ce contexte, nous avons varié d'une part la taille du réseau et d'autre part la cardinalité i.e. le nombre de chemins calculés.

Le **tableau** présente le taux de précision pour un réseau contenant 200 nœuds qui sont déployés dans une zone de dimension 200x200 et tel que la portée des nœuds est 160.

Le coût moyen est la somme des coûts des chemins divisé par le nombre des chemins.

**Tableau III-3 : Taux de précision (200 nœuds)**

| Cardinalité | GSR_M  | Recuit Simulé | taux de précision |
|-------------|--------|---------------|-------------------|
| 10          | 113,8  | 126,2         | 0,9               |
| 20          | 129,24 | 136,8         | 0,94              |
| 30          | 135,6  | 140,83        | 0,96              |
| 40          | 138,35 | 145,35        | 0,97              |
| 50          | 133,54 | 140,32        | 0,98              |



**Figure III.7: Coût moyen (200 nœuds)**

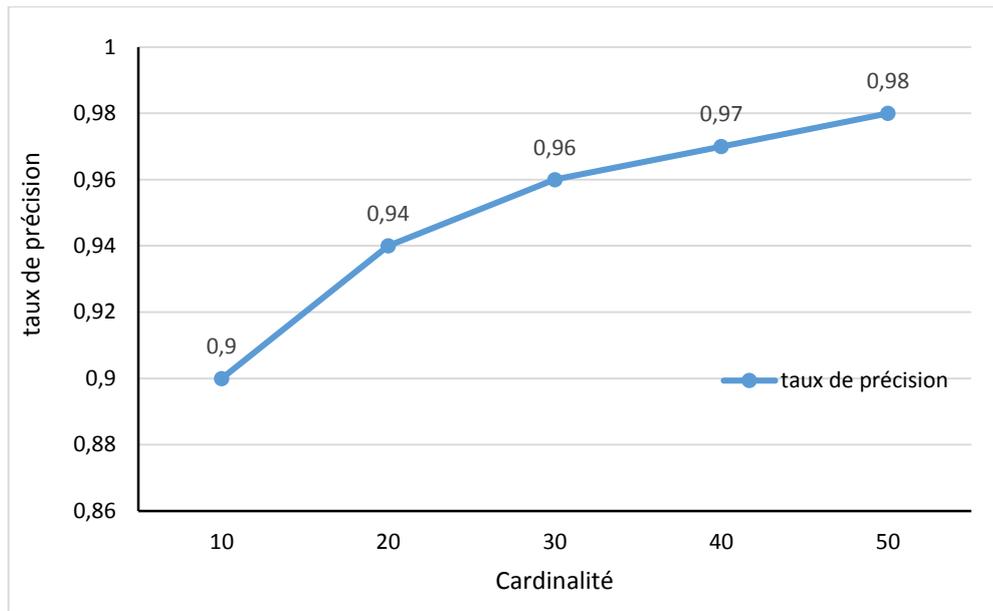


Figure III.8: Taux de précision (200 nœuds)

Tableau III-4: Coût moyen et taux de précision (500 nœuds)

| Cardinalité | GSR_M  | Recuit Simulé | taux de précision |
|-------------|--------|---------------|-------------------|
| 10          | 365,8  | 528,6         | 0,69              |
| 20          | 360,6  | 537,65        | 0,67              |
| 30          | 545,65 | 707,8         | 0,77              |
| 40          | 520,2  | 654,84        | 0,79              |
| 50          | 523,8  | 590,14        | 0,88              |

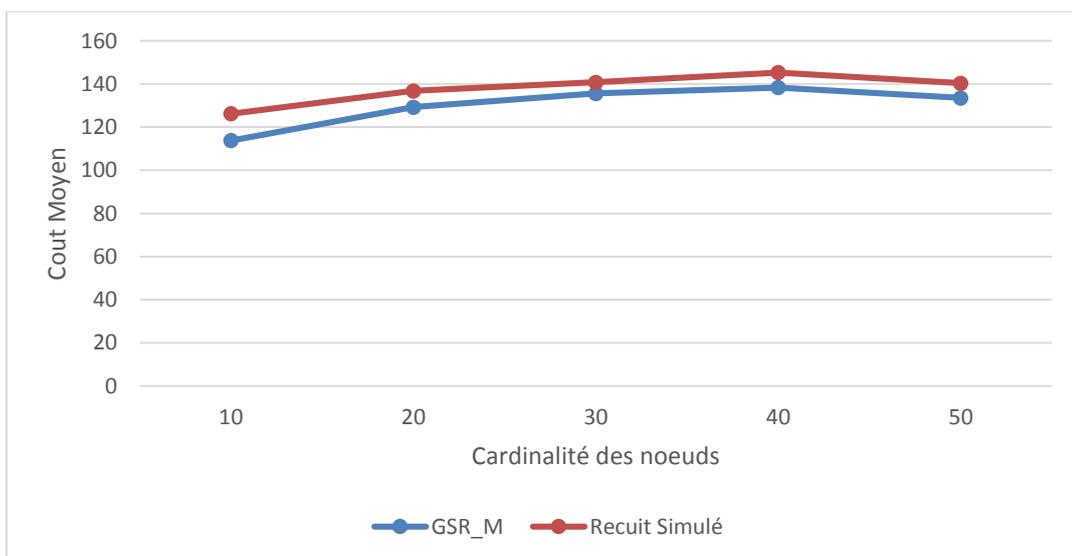
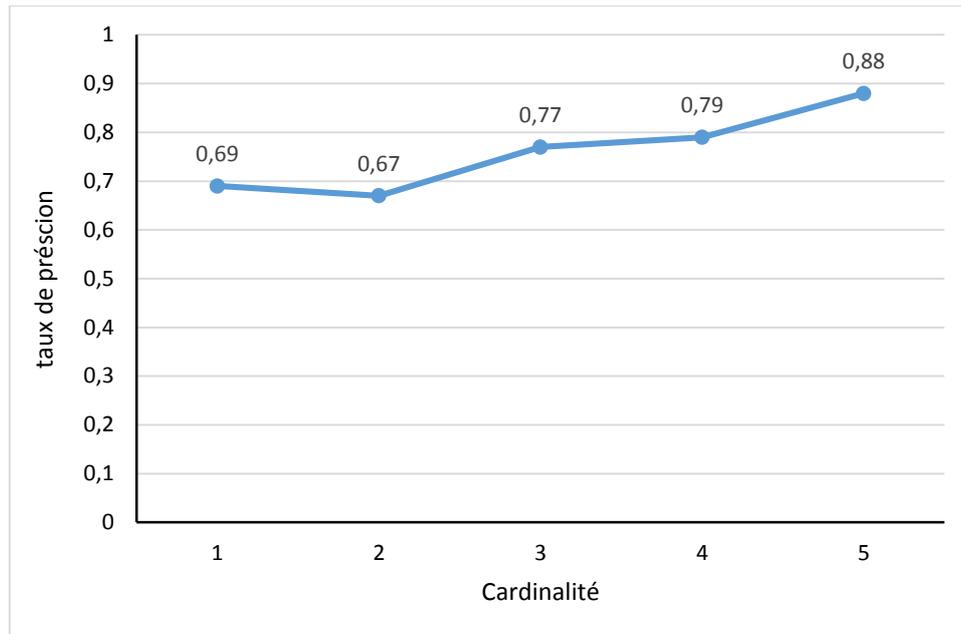


Figure III.9: Coût moyen (500 nœuds)



**Figure III.10: Taux de précision (500 nœuds)**

Dans les tableaux et les figures présentés ci-dessus, nous avons évalué respectivement le coût moyen des chemins et le taux précision dans GSR\_M et le recuit simulé pour différentes tailles d'un réseau. Nous avons remarqué que le taux de précision est acceptable dans la plupart des scénarios.

### III.6 Conclusion

Dans ce chapitre, nous avons proposé une version de GSR qui est adaptée aux RCSF et un algorithme de routage basé sur l'heuristique recuit simulé.

L'évaluation de ces deux propositions dans un réseau de capteurs dense a montré que la version de GSR simplifiée demande plus de temps pour construire les routes entre les nœuds et la station de base alors celle basée sur l'heuristique recuit simulé nécessite un temps raisonnable. En outre, pour mettre en valeur l'apport de l'heuristique recuit simulé, nous avons estimé la qualité de la solution fournie par cette approche comparativement à GSR\_M. Les résultats étaient satisfaisants puisque le taux de précision est de l'ordre de 90%.

## Conclusion générale

La plupart des protocoles de routage conçus pour les réseaux sans fil et en particulier pour les RCSF ont été proposés pour des réseaux de taille moyenne. Ainsi, lors du passage à l'échelle les performances de ces protocoles se dégradent. Dans cette optique, nous avons évalué une version simplifiée de GSR dans un environnement dense. Les résultats ont montré que ce protocole perd ses performances en passant à l'échelle. A cet effet, nous avons conçu un algorithme de routage basé sur l'heuristique recuit simulé en visant à minimiser le temps nécessaire pour la construction des routes.

Les résultats obtenus en termes de temps nécessaire pour l'établissement de routes entre les nœuds et la station de base comparativement à la version simplifiée de GSR. En outre, pour mettre en valeur cette proposition, nous avons estimé la qualité de la solution fournie par l'heuristique utilisée en calculant le taux de précision. Le résultat obtenu est satisfaisant puisqu'il est au voisinage de 90% pour différentes tailles d'un réseau.

En perspectives, nous proposons d'évaluer d'autres protocoles de routage et les faire adapter à des environnements denses. En outre, utiliser d'autres heuristiques pour la conception de protocoles de routage pour des RCSF à grande échelle.

---

## Références Bibliographiques

- [1] Kamel BEYDOUN, "Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs", Thèse de Doctorat, U.F.R des Sciences et Techniques de l'université de Franche-Comté, Décembre 2009.
- [2] Binh Thanh DOAN, "Algorithme d'initialisation économe en énergie dans les réseaux radio multi-sauts", Mémoire de fin d'études Hanoi, Mars 2006.
- [3] Aurélie Bunel, "Les réseaux de capteurs sans fil", Exposé NT Système et réseaux, Université de Marne-la-Vallée, <http://www-igm.univ-mlv.fr/~dr/XPOSE2006/Bunel/Presentation.html>, Mars 2007.
- [4] Alexandre GONDRAN, "Modélisation et optimisation de planification des réseaux locaux sans fil", Thèse de Doctorat, Université de technologie Belford-Montebéliard de Franche-Comté, Décembre 2008.
- [5] Gérard CHALHOUB, "Les réseaux de capteurs sans fil", Conférence, Université de Clermont-Ferrand, Novembre 2010.
- [6] A.Hertz, "Les métaheuristiques : quelques conseils pour en faire bon usage, in Gestion de Production et Ressources Humaines : méthodes de planification dans les systèmes productifs", Presses Internationales de Polytechnique, 205-222, 2005.
- [7] [http://interstices.info/jcms/c\\_43811/le-recuit-simule](http://interstices.info/jcms/c_43811/le-recuit-simule), 11/06/2013.
- [8] Yaser YOUSEF, "Routage pour la Gestion de l'Energie dans les Réseaux de Capteurs Sans Fil, Thèse de Doctorat, Université de Haute Alsace, Faculté des Sciences et Techniques, Juillet 2010.
- [9] Lilia ZIANE KHODJA, "La structuration et la sécurisation des réseaux de capteurs", Master 2 Recherche Informatique, IFSIC.
- [10] Tsu-Wei Chen and Mario Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks", IEEE International Conference on Communications, Vol. 1, pp171-175, Atlanta, USA, Juin 1998.
- [11] Jordan LAETITIA, "Méta-heuristiques pour l'extraction de connaissance : Application à la génomique", Thèse de Doctorat, U.S.T.L de l'université de Lille, Novembre 2003.

- [12] Mohamed AISSANI, OPTIMISATION DU ROUTAGE DANS LES RESEAUX DE CAPTEURS POUR LES APPLICATIONS TEMPS, le grade de Docteur de l'Université Paris, le diplôme de Doctorat en Informatique (USTHB), le 13 mars 2011.
- [13] <http://igm.univ-mlv.fr/~dr/XPOSE2006/Bunel/Presentation.html#unreseau>, le 11/06/2013.
- [14] Samir ATHMANI, Protocole de sécurité Pour les Réseaux de capteurs Sans Fil, Pour l'obtention du Magistère en Informatique, le : 15/07/2010.
- [15] LEHSAINI Mohamed, Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique, Thèse de Doctorat, 2009.

---

## Liste des figures

|  |    |
|--|----|
| Figure I.1: Eléments d'un capteur sans fil.....                | 6  |
| Figure I.3: Collecte d'informations à la demande.....          | 11 |
| Figure I.2: Architecture de communication.....                 | 11 |
| Figure I.4: Application orientée événement.....                | 12 |
| Figure I.5: Pile protocolaire dans les RCSF.....               | 13 |
| Figure II.1: Algorithme GSR [10].....                          | 21 |
| Figure III.1: Algorithme de Dijkstra.....                      | 27 |
| Figure III.2: Classification des méta-heuristiques [8].....    | 28 |
| Figure III.3: Algorithme du recuit simulé.....                 | 32 |
| Figure III.4: La fonction Accepte.....                         | 33 |
| Figure III.5: Temps d'exécution (GSR_M vs. Recuit simulé)..... | 34 |
| Figure III.6: Coût moyen vs. cardinalité (1000 nœuds).....     | 35 |
| Figure III.7: Coût moyen (200 nœuds).....                      | 36 |
| Figure III.8: Taux de précision (200 nœuds).....               | 37 |
| Figure III.9: Coût moyen (500 nœuds).....                      | 37 |
| Figure III.10: Taux de précision (500 nœuds).....              | 38 |

## Liste des tableaux

|  |    |
|--|----|
| Tableau II-1 : Comparaison des complexités de GSR, LS et DBF.....                | 23 |
| Tableau III-1: Comparaison des temps d'exécution (GSR_M vs. Recuit simulé) ..... | 34 |
| Tableau III-2 : Coût moyen (GSR_M et Recuit simulé) vs. cardinalité .....        | 35 |
| Tableau III-3 : Taux de précision (200 nœuds) .....                              | 36 |
| Tableau III-4: Coût moyen et taux de précision (500 nœuds) .....                 | 37 |

## Liste des abréviations

RCSF : Réseaux de capteur sans fils

QOS : Quality Of Service

VPN: Virtual Protocol Network

SMP: Sensor Management Protocol

TADAP: Task Assignment and Data Advertisement Protocol

UDP: User Datagram Protocol

UDP-Like: User Datagram Protocol Like

TCP: Transmission Control Protocol

TCP-splitting: Transmission Control Protocol-splitting

SAR: Sequential Assignment Routing

SMACS: Self-organizing Medium Access Control for Sensor networks

EAR: Eavesdrop And Register

GPS: Global Positioning System

TC : Topology Control

DBF : Distributed Bellman-Ford.

LS : Link State

DSDV: Destination Sequenced Distance Vector

DSR: Dynamic Source Routing

OLSR: Optimized Link State Routing

HEED: Hybrid, Energy-Efficient, Distributed approach

LEACH: Low Energy Adaptive Clustering Hierarchy

PEGASIS: Power-Efficient Gathering in Sensor Information Systems

GSR : Global State Routing.

## **Résumé : Conception d'un algorithme de routage basé sur l'heuristique recuit simulé pour les réseaux de capteurs à grande échelle**

Dans les réseaux sans fil, la plupart des protocoles de routage ont été conçus pour des réseaux de taille moyenne. Ainsi, lors du passage à l'échelle les performances de ces protocoles se dégradent. Dans cette optique, nous avons proposé une version simplifiée de GSR. Puis, nous avons évalué cette version dans des environnements denses. Les résultats obtenus ont montré que le temps pour construire les routes entre les nœuds et la station de base était consistant. Pour réduire ce temps, nous avons proposé un algorithme de routage basé sur l'heuristique recuit simulé. Les résultats fournis étaient satisfaisants en termes de temps de construction de routes et de précision de résultats.

**Les mots clés :** Réseau de capteurs, Protocole de routage, GSR, Dijkstra, recuit simulé.

### **Abstract: Design of a routing algorithm based on simulated annealing heuristic for dense sensor networks**

In wireless networks, most routing protocols have been designed for medium-sized networks. Thus, when the number of nodes increases the performance of these protocols degrade. In this context, we proposed a simplified version of GSR. Then, we evaluated this version in dense environments. The results showed that the time to find paths between nodes and base station were consistent. To reduce this time, we have proposed a routing algorithm based on simulated annealing heuristic. The results obtained were better in terms of time of path construction and accuracy of results.

**Key words:** Wireless networks sensor, routing protocol, GSR, Dijkstra, Simulated Annealing.

الخلاصة: تصميم خوارزمية التوجيه على أساس محاكاة ارشادي الصلب لشبكات الاستشعار كثيفة

في الشبكات اللاسلكية، يتم تصميم معظم بروتوكولات التوجيه لشبكات المتوسطة الحجم. وكذلك، عند الانتقال الى شبكات ذات النطاق الواسع فإن أداء هذه البروتوكولات يتناقص. وفي هذا السياق، اقترحنا نسخة مبسطة من بروتوكول GSR. ثم قمنا بتقييم هذا الإصدار في البيئات الكثيفة. وأظهرت النتائج أن الوقت لبناء الطرق بين العقد ومحطة قاعدة كانت متنسقة. للحد من هذا الوقت، اقترحنا خوارزمية التوجيه على أساس محاكاة الصلب ارشادي. وكانت النتائج التي تم الحصول عليها مرضية من حيث الوقت من بناء الطرق ودقة النتائج.

الكلمات المفتاحية: الشبكات الاستشعارية، بروتوكولات التوجيه.