



**HAL**  
open science

# Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil

Yaser Yousef

► **To cite this version:**

Yaser Yousef. Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil. Autre [cs.OH]. Université de Haute Alsace - Mulhouse, 2010. Français. NNT : 2010MULH3206 . tel-00590407

**HAL Id: tel-00590407**

**<https://theses.hal.science/tel-00590407>**

Submitted on 3 May 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITE DE HAUTE ALSACE  
Faculté des Sciences et Techniques

**Thèse de Doctorat**  
Spécialité INFORMATIQUE

présentée Par :

**YOUSEF Yaser**

Sujet de la thèse :

**Routage pour la Gestion de l’Energie dans les Réseaux de  
Capteurs Sans Fil**

Soutenue le 08/07/2010 devant le jury :

Rapporteurs : M. Zoubir MAMMERY, Professeur, IRIT, Université Paul Sabatier  
M. Joel RODRIGUES, Professeur, Université de Beira Interior de  
Covilhã (Portugal)

Examineur : M. Tijani CHAHED, Professeur, Telecom SudParis

Directeur : M. Pascal LORENZ, Professeur, Université de Haute Alsace

Co-encadrant : M. Marc GILG, Maître de conférences, Université de Haute Alsace



# Remerciements

*C'est avec un grand plaisir que je réserve ces lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail. Ce travail, qui a été réalisé au sein du Groupe de Recherche en Réseaux et Télécommunications de Colmar (GRTC), m'a permis de rencontrer des personnes, qui m'ont apporté beaucoup d'aide et de soutien dans mes recherches et que je souhaite aujourd'hui remercier.*

*Je veux exprimer ma gratitude et ma reconnaissance à mon directeur de thèse Monsieur Pascal LORENZ pour m'avoir accueilli dans son établissement et pour son encouragement, sa disponibilité, ses idées et ses conseils qui m'ont permis de mener à bien cette thèse.*

*J'adresse mes remerciements les plus sincères à Monsieur Marc GILG, pour ses conseils lucides et pertinents, sa patience et son précieux suivi du mémoire.*

*Mes remerciements les plus sincères vont à monsieur Zoubir MAMMARI, Professeur à l'Université Paul Sabatier et Monsieur Joel RODRIGUES, Professeur à l'Université de Beira Interior de Covilhã (Portugal), pour avoir accepté d'être mes rapporteurs et d'avoir consacré une partie de leur temps à la lecture de mon manuscrit.*

*Je voudrais aussi remercier monsieur Tijani CHAHED, Professeur à Telecom SudParis, qui a contribué à la diversité de mon jury de thèse en y apportant sa grande expérience dans le domaine des réseaux de télécommunication. Je lui en suis très reconnaissant.*

*Je souhaite également remercier tous les membres du laboratoire GRTC, chercheurs, thésards, stagiaires et visiteurs, pour leur disponibilité, leur sympathie et pour l'ambiance amicale qui règne au sein du laboratoire GRTC.*

*Enfin je ne trouverais sans doute pas les mots pour remercier assez les personnes qui me sont les plus chères : les membres de ma famille, pour leur sacrifices, leur patience, et tout ce qu'ils ont fait pour m'apporter le bonheur. Qu'ils sachent à travers ces quelques mots combien je leur suis reconnaissant et combien je sais tout ce que je leur dois.*

# *Dédicaces*

*À ma mère ;*

*À mon père ;*

*À mes cinq sœurs ;*

*À mes six frères ;*

*À toute ma famille ;*

*À tous ceux qui me sont cher ;*

# Table des matières

<b>Introduction générale .....</b>	<b>1</b>
1.1. Contexte général .....	2
1.2. Motivations et problématique .....	3
1.3. Contributions et structure de la thèse .....	4
<b>Les Réseaux de Capteurs Sans Fil (RCSF) .....</b>	<b>6</b>
2.1 Introduction .....	8
2.2 Qu'est-ce qu'un capteur sans fil .....	9
2.3 Quelques applications des réseaux de capteurs .....	10
2.4 Architecture d'un RCSF .....	11
2.5 Contraintes de conception des RCSF .....	12
2.6 Consommation d'énergie dans les RCSF .....	14
2.6.1 Energie de capture .....	14
2.6.2 Energie de traitement .....	14
2.6.3 Energie de communication .....	14
2.7 Techniques de minimisation de la consommation d'énergie .....	15
2.8 Les principaux facteurs de protocoles de routage .....	17
2.9 Les principaux protocoles de routage pour les réseaux ad hoc .....	18
2.9.1 Les protocoles de routage proactifs .....	18
2.9.1.1 Le protocole de routage «DSDV» .....	19
2.9.1.2 Le protocole de routage « FSR » .....	19
2.9.1.3 Le protocole de routage « OLSR » .....	20
2.9.2 Les protocoles de routage réactifs .....	21

2.9.2.1	Le protocole de routage « DSR ».....	21
2.9.2.2	Le protocole de routage « AODV ».....	21
2.10	Les principaux protocoles de routages dans les RCSF .....	22
2.10.1	Les Protocoles hiérarchiques.....	23
2.10.1.1	Le protocole de routage «LEACH » .....	24
2.10.1.2	Les protocoles de routage «PEGASIS & Hierarchical-PEGASIS».....	24
2.10.1.3	Les protocoles de routage « TEEN and APTEEN » .....	25
2.10.2	Les protocoles de routage basés sur la localisation.....	25
2.10.2.1	Le protocole de routage « MECN ».....	26
2.10.2.2	Le protocole de routage « GAF » .....	26
2.10.2.3	Le protocole de routage «GEAR ».....	27
2.10.3	Les protocoles de routage ‘ <i>data-centric</i> ’ .....	27
2.10.3.1	Le protocole de routage « SPIN ».....	28
2.10.3.2	La diffusion dirigée.....	29
2.10.3.3	Le protocole de routage par rumeur.....	30
2.11	Les principaux algorithmes de conservation d’énergie .....	31
2.11.1	L’algorithme de routage « EARLEAHSN » .....	33
2.11.2	L’algorithme de routage «EARCBSN ».....	33
2.11.3	L’algorithme de routage « GBR ».....	34
2.12	Conclusion .....	34
<b>Techniques de traitement d’image et modélisation.....</b>		<b>37</b>
3.1	Introduction .....	39
3.2	Techniques de traitement d’image.....	39
3.2.1	Représentation d’une image en niveau de gris.....	39
3.2.2	Histogramme .....	40
3.2.3	Détection de contours.....	41
3.2.4	Le gradient d’une image .....	41
3.2.5	Les filtres en traitement d’image.....	42

3.2.5.1	Les filtres de convolution.....	43
3.2.5.2	Exemples des filtres en traitement d'image .....	44
3.2.5.2.1	Exemple de filtre passe-bas .....	44
3.2.5.2.2	Exemple de filtre moyen.....	45
3.2.5.2.3	Exemple de filtre gradient.....	46
3.2.5.2.4	Exemple de filtre Sobel.....	47
3.3	Modélisation .....	49
3.3.1	Modèle du réseau .....	49
3.3.2	Modèle énergétique .....	49
3.3.3	Analogie entre les RCSF et le niveau de gris d'une image.....	50
3.3.3.1	Les secteurs d'énergie.....	51
3.3.3.2	La matrice M.....	52
3.3.3.3	Le produit de convolution.....	52
3.3.3.3.1	Le filtre moyen.....	53
3.3.3.3.2	Le filtre de Sobel.....	54
3.3.4	Étude de gradient d'énergie dans les RCSF.....	58
3.4	Conclusion .....	70
	<b>La gestion de l'énergie dans les RCSFs par les filtres de convolution .....</b>	<b>72</b>
4.1	Introduction .....	74
4.2	L'algorithme de routage basé sur le gradient d'énergie .....	74
4.2.1	L'algorithme de Sobel en RCSF .....	75
4.2.1.1	Les paramètres de l'algorithme.....	75
4.2.1.2	Les étapes de l'algorithme de routage par Sobel .....	75
4.2.1.2.1	Calculer le gradient d'énergie.....	75
4.2.1.2.2	L'implémentation de l'algorithme de traitement d'image dans le protocole AODV.....	76

4.2.2	Simulation de l’algorithme de routage basé sur Sobel.....	77
4.2.3	Conclusion de l’algorithme de routage basé sur Sobel .....	80
4.3	L’algorithme de routage basé sur le filtre moyen.....	81
4.3.1	Les paramètres de l’algorithme .....	82
4.3.2	Les étapes de l’algorithme de routage.....	82
4.3.3	Simulation de l’algorithme de routage basé sur le filtre moyen .....	83
4.3.3.1	OMNeT++ .....	83
4.3.3.2	Les simulations de l’algorithme par OMNeT++.....	84
4.3.4	Conclusion de l’algorithme de routage basé sur le filtre moyen.....	89
4.4	Algorithme hybride.....	89
4.4.1	Introduction .....	89
4.4.2	Les secteurs de l’énergie et la matrice associée.....	90
4.4.3	L’algorithme de routage basé sur les deux filtres (moyen et Sobel).....	90
4.4.3.1	L’algorithme hybride .....	91
4.4.3.2	Le routage des paquets.....	92
4.4.4	Simulation de l’algorithme hybride.....	94
4.4.5	Simulation de l’algorithme hybride avec dix nœuds.....	105
4.4.6	Conclusion de l’algorithme hybride.....	114
4.5	Conclusion.....	114
	<b>Conclusion et Perspectives .....</b>	<b>116</b>
5.1	Conclusion.....	117
5.2	Perspectives .....	118
	<b>REFERENCES .....</b>	<b>127</b>

# Chapitre 1

## Introduction générale

### Sommaire

---

1.1	Contexte général .....	2
1.2	Motivations et problématique .....	3
1.3	Contributions et structure de la thèse .....	4

## 1.1. Contexte général

Les progrès réalisés dans le domaine des réseaux sans fil ont contribué à l'évolution de l'Internet en facilitant l'accès aux usagers indépendamment de leur position géographique. L'évolution dans le domaine des communications sans fil et l'informatique mobile gagne de plus en plus une popularité et les composants mobiles deviennent de plus en plus fréquents. Comme beaucoup de développements technologiques, les réseaux de capteurs sans fil ont émergé pour des besoins militaires tels que la surveillance sur le terrain de combat. Puis, ils ont trouvé leur chemin pour des applications civiles. Aujourd'hui, les réseaux de capteurs sans fil sont devenus une technologie clé pour les différents types d' "environnements intelligents". De nos jours, ils nous aident par exemple à avoir un bon système de sécurité à la maison. Ces réseaux sont d'une importance particulière quand un grand nombre de nœuds de capteurs doivent être déployés, dans des situations dangereuses. Par exemple, pour une gestion des catastrophes, un grand nombre de capteurs peuvent être largués par un hélicoptère. Ces capteurs peuvent aider à réaliser des opérations de sauvetage en localisant les survivants, pour l'identification des zones à risque ou pour renseigner l'équipe de secours. Cette demande de réseaux de capteurs peut non seulement accroître l'efficacité des opérations de sauvetage, mais aussi assurer la sécurité de l'équipe de secours.

Les réseaux de capteurs sont constitués de plusieurs capteurs minuscules ou nœuds ayant une caractéristique essentielle résidant dans l'absence d'infrastructure fixe et ayant une topologie changeante due à la mobilité des capteurs et pose le problème de l'épuisement de leurs batteries.

Dans de nombreuses applications, on veut recueillir les données de tous les capteurs dans une station spécifique pour le traitement, ou pour l'archivage.

Les principaux problèmes dans les réseaux de capteurs sans fil ou les WSNs "Wireless Sensor Networks" sont le protocole de routage, l'énergie consommée par le nœud, la sécurité, l'agrégation de données, la mobilité imprévisible des nœuds, etc. Ces capteurs sont parfois déployés dans des zones hostiles. Il est donc nécessaire d'avoir une stratégie efficace qui

prend en considération l'énergie du réseau pour augmenter sa durée de vie en réduisant la perte d'énergie tout en étant réactif aux changements de l'environnement.

Les progrès récents dans les réseaux de capteurs sans fil ont conduit au développement de nombreux protocoles spécifiquement conçus pour les réseaux de capteurs où la présentation de l'énergie est un facteur essentiel. Le protocole Wifi 802.11, qui est conçu pour les réseaux LANs sans fil, est également utilisé pour les réseaux de capteurs. Cependant, la consommation d'énergie élevée et le débit excessif rendent ces protocoles non appropriés aux réseaux de capteurs et ceci a conduit les chercheurs à concevoir des protocoles d'accès efficaces en énergie. L'objectif principal de la plupart des protocoles de routage est de trouver des algorithmes et des techniques efficaces en terme d'énergie afin que la durée de vie du réseau soit maximisée.

## **1.2. Motivations et problématique**

La mise en œuvre des traitements, de stockage, de détection et de communication dans des dispositifs de petite taille, à faible coût et leur intégration dans les réseaux de capteurs sans fil ouvre la porte à une multitude de nouvelles applications.

La motivation principale de notre travail est l'efficacité dans l'utilisation de ces dispositifs. Cette efficacité est d'une importance vitale étant donné que ces capteurs fonctionnent généralement sur piles et que dans de nombreuses ces piles applications ne peuvent pas être remplacées ou rechargées.

Les techniques conçues pour les réseaux ad hoc traditionnels ne sont pas bien adaptées aux réseaux de capteurs. De nombreuses contraintes doivent être résolues pour les réseaux de capteurs sans fil. Nous nous intéressons, dans cette thèse, aux contraintes posées par le routage et par la gestion de l'énergie dans ce type de réseaux pour prolonger la durée de vie de ces réseaux. Pour cela, notre objectif est de proposer des algorithmes de routage basés sur la conservation de l'énergie en faisant participer des nœuds ayant des batteries pleines par rapport à leurs voisins tout en évitant les zones pauvres en énergie ou en capteurs.

Nos algorithmes sont basés sur les mécanismes et les techniques de traitement d'images afin de trouver une nouvelle manière de router l'information. L'idée principale de notre travail est de traiter les capteurs dans un réseau de capteurs tel que cela se fait pour les pixels dans une image. Dans cette représentation, les zones claires de l'image représentent les capteurs qui ont une batterie pleine et les zones sombres représentent soit les régions du réseau sans capteur ou les régions qui ont des capteurs épuisés. Ainsi, le gris dans une image représentera les différents niveaux de capacité énergétique des capteurs.

Le principe de fonctionnement de notre système de routage, à l'aide de cette représentation, est de détecter les zones claires du réseau afin d'envoyer les paquets de la source vers la destination en passant par ces zones.

### **1.3. Contributions et structure de la thèse**

Dans ce mémoire de thèse, notre contribution se concentre sur la réalisation des algorithmes de routage dont l'objectif est de traiter le problème de gestion de ressources afin maximiser la durée de vie du réseau. Pour cela, nous avons proposé des algorithmes basés sur la fusion de deux aspects :

- le routage dans les réseaux de capteurs sans fil.
- les techniques de traitement d'image.

Le principe de ces algorithmes est de considérer les nœuds dans un réseau de capteurs comme une carte de pixels pour une image. Ensuite, à l'aide des filtres de convolution, nous définissons une nouvelle métrique pour choisir le chemin en fonction de l'énergie disponible et nous utilisons AODV pour acheminer les paquets en utilisant cette nouvelle métrique. De cette façon, le routage de paquets se réalise en passant par les nœuds ayant une importante valeur énergétique. Les filtres employés dans nos contributions déterminent la répartition de l'énergie autour d'un nœud central afin de choisir les nœuds voisins qui vont participer au routage des paquets.

Ce mémoire de thèse se divise en cinq chapitres :

- Nous présentons dans le **premier chapitre** une introduction générale qui constitue le contexte général, les motivations et la problématique de ce travail.
- Dans la première partie du **deuxième chapitre**, nous définissons les réseaux de capteurs sans fil, leurs domaines d'applications, leur architecture et les contraintes liées à ce type de réseau. Nous nous concentrons sur leurs contraintes basées sur la consommation d'énergie. Nous présentons les différentes opérations par lesquelles l'énergie est consommée et les techniques de minimisation de cette consommation d'énergie. Nous enchaînons ensuite, dans la deuxième partie, sur les protocoles de routage les plus connus dans les réseaux ad hoc puis dans les réseaux de capteurs. Nous détaillons les principaux algorithmes réalisés dans le but de conserver l'énergie des nœuds de capteurs.
- Le **troisième chapitre** introduit une analogie avec les différents composants des réseaux de capteurs sans fil et le niveau de gris d'une image. Dans ce chapitre, nous présentons certaines techniques de traitement d'image qui nous seront utiles telles que le gradient d'une image, les filtres de convolution, etc. Nous donnons par la suite la formulation mathématique et le détail des calculs avec des preuves théoriques de nos algorithmes proposés dans le quatrième chapitre.
- Dans le **quatrième chapitre**, nous détaillons tout d'abord notre algorithme de routage basé sur la direction du gradient et AODV pour réaliser un routage économe en terme d'énergie. Notre deuxième contribution est détaillée dans la deuxième partie de ce chapitre à travers la présentation d'un nouvel algorithme de routage basé sur le filtre moyen de convolution. Enfin nous introduisons une troisième contribution à travers la réalisation d'un algorithme hybride. Les performances de nos algorithmes en terme d'énergie sont détaillées à travers l'outil de simulation OMNeT++.
- Le **cinquième chapitre** conclut ce mémoire de thèse et rappelle les différentes contributions réalisées tout au long de ce travail de recherche. Nous proposons également des perspectives de recherche futures.

# Chapitre 2

## Les Réseaux de Capteurs Sans Fil (RCSF)

### Sommaire

---

2.1 Introduction .....	8
2.2 Qu'est-ce qu'un capteur sans fil .....	9
2.3 Quelques applications des réseaux de capteurs .....	10
2.4 Architecture d'un RCSF .....	12
2.5 Contraintes de conception des RCSF .....	12
2.6 Consommation d'énergie dans les RCSF .....	14
2.6.1 Energie de capture .....	14
2.6.2 Energie de traitement .....	15
2.6.3 Energie de communication .....	15
2.7 Techniques de minimisation de la consommation d'énergie .....	15
2.8 Les principaux facteurs de protocoles de routage .....	17
2.9 Les principaux protocoles de routage pour les réseaux ad hoc .....	19
2.9.1 Les protocoles de routage proactifs .....	19
2.9.1.1 Le protocole de routage «DSDV» .....	20
2.9.1.2 Le protocole de routage « FSR » .....	20
2.9.1.3 Le protocole de routage « OLSR » .....	21
2.9.2 Les protocoles de routage réactifs .....	22
2.9.2.1 Le protocole de routage « DSR » .....	22
2.9.2.2 Le protocole de routage « AODV » .....	22
2.10 Les principaux protocoles de routages dans les RCSF .....	23
2.10.1 Les Protocoles hiérarchiques .....	24

2.10.1.1	Le protocole de routage «LEACH » .....	25
2.10.1.2	Les protocoles de routage «PEGASIS & Hierarchical-PEGASIS».....	25
2.10.1.3	Les protocoles de routage « TEEN and APTEEN » .....	26
2.10.2	Les protocoles de routage basés sur la localisation .....	26
2.10.2.1	Le protocole de routage « MECN » .....	27
2.10.2.2	Le protocole de routage « GAF » .....	27
2.10.2.3	Le protocole de routage «GEAR » .....	28
2.10.3	Les protocoles de routage ‘Data-centric’ .....	28
2.10.3.1	Le protocole de routage « SPIN » .....	29
2.10.3.2	La diffusion dirigée .....	30
2.10.3.3	Le protocole de routage par rumeur .....	31
2.11	Les principaux algorithmes de conservation d’énergie .....	33
2.11.1	L’algorithme de routage « EARLEAHSN » .....	34
2.11.2	L’algorithme de routage «EARCBSN » .....	34
2.11.3	L’algorithme de routage « GBR » .....	35
2.12	Conclusion .....	35

## 2.1 Introduction

De nombreux systèmes nécessitent de prendre en compte l'environnement. Les progrès de ces dernières années en microélectronique ont permis de fabriquer des capteurs de plus en plus petits, de plus en plus performants et avec des autonomies énergétiques grandissantes. D'autre part, les techniques de réseaux mobiles permettent désormais de s'affranchir des fils et donc de déployer facilement des réseaux de capteurs, dans des endroits même difficiles d'accès.

Un réseau de capteurs peut être vu comme un réseau de microsystemes disséminés dans un espace donné et communicant entre eux via une liaison sans fil. L'espace où agissent les capteurs s'appelle un champ de captage. Ce qui est intéressant dans les réseaux de capteurs, c'est que les nœuds sont souvent composés d'un grand nombre de micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome.

Par conséquent, on peut définir un Réseau de Capteurs Sans Fil (RCSF) ou "Wireless Sensor Network" (WSN) comme un ensemble de dispositifs très petits, nommés nœuds capteurs, variant de quelques dizaines d'éléments à plusieurs milliers. Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil.

Il existe plusieurs types de protocole de routage dans le réseau de capteurs sans fil. Ces protocoles de routage sont créés afin de permettre la prévention des collisions et l'économie d'énergie. Avant d'entrer dans l'étude détaillée des protocoles de réseau de capteurs sans fil, nous commençons par présenter un capteur sans fil, ses applications et son architecture. Ensuite nous expliquons les différentes contraintes dans un réseau de capteur et particulièrement la consommation d'énergie et les différentes sources causant la perte énergétique et les techniques pour minimiser cette perte.

## 2.2 Qu'est-ce qu'un capteur sans fil

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, etc.) et de la communiquer à un centre de contrôle via une station de base. Les progrès conjoints de la microélectronique, des technologies de transmission sans fil et des applications logicielles ont permis de produire à coût raisonnable des micro-capteurs de quelques millimètres cubes de volume, susceptibles de fonctionner en réseaux [1]. Un capteur est composé de quatre unités de base (voir figure 1):

1. **L'unité d'acquisition** : elle est généralement composée de deux sous-unités qui sont les capteurs et les convertisseurs analogique-numérique ADCs (Analog-Digital Converter). Les capteurs obtiennent des mesures sur les paramètres environnementaux et les transforment en signaux analogiques. Les ADCs convertissent ces signaux analogiques en signaux numériques.
  2. **L'unité de traitement** : elle est composée de deux interfaces qui sont 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 stocker les données collectées.
  3. **Un module de communication (Transceiver)** : il est composé d'un émetteur/récepteur permettant la communication entre les différents nœuds du réseau via un support de communication radio.
  4. **Batterie** : elle alimente les unités que nous avons citées et elle n'est généralement ni rechargeable ni remplaçable. La capacité d'énergie limitée au niveau des capteurs représente la contrainte principale lors de conception de protocoles pour les réseaux de capteurs.
- Il existe des capteurs qui sont dotés d'autres composants additionnels tels que les systèmes de localisation GPS (Global Position System).

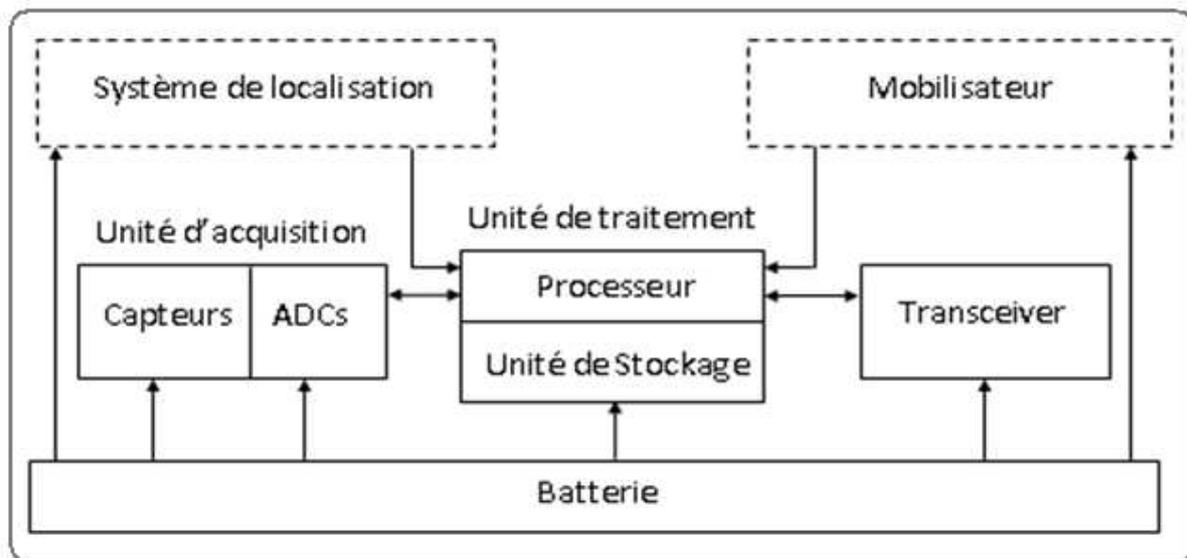


Figure 1 Les composants d'un nœud capteur

## 2.3 Quelques applications des réseaux de capteurs

La diminution de taille et de coût des micro-capteurs, l'élargissement de la gamme des types de capteurs disponibles (thermique, optique, vibrations, ...) et l'évolution des supports de communication sans fil ont élargi le champ d'application des réseaux de capteurs. Les RCSF peuvent être utilisés dans plusieurs applications [2, 3, 4, 5]. Parmi elles, nous citons :

- **Découverte de catastrophes naturelles** : on peut créer un réseau autonome en dispersant les nœuds dans la nature. Des capteurs peuvent ainsi signaler des événements tels que les feux de forêts, les tempêtes ou les inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours [6].
- **Détection d'intrusions** : en plaçant à différents points stratégiques des capteurs, on peut ainsi prévenir des cambriolages ou des passages de gibier sur une voie de chemin de fer (par exemple) sans avoir à recourir à de coûteux dispositifs de surveillance vidéo.
- **Gestion de stock** : on pourrait imaginer devoir stocker des denrées nécessitant un certain taux d'humidité et une certaine température. Dans ces applications, le réseau doit pouvoir collecter ces différentes informations et alerter en temps réel si les seuils critiques sont dépassés.

- **Contrôle de la pollution** : des capteurs au-dessus d'un emplacement industriel offrent la possibilité de détecter et de contrôler des fuites de gaz ou de produits chimiques. Ces applications permettent de donner l'alerte en un temps record et de pouvoir suivre l'évolution de la catastrophe [7].
- **Agriculture** : des nœuds peuvent être incorporés dans la terre et on peut interroger le réseau sur l'état du champ et déterminer par exemple les secteurs les plus secs afin de les arroser en priorité. On peut aussi imaginer équiper des troupeaux de bétail de capteurs pour connaître en tout temps, leur position ce qui éviterait aux éleveurs d'avoir recours à des chiens de berger.
- **Surveillance médicale** : en implantant sous la peau de mini capteurs vidéo, on peut recevoir des images d'une partie du corps en temps réel sans aucune chirurgie. On peut ainsi surveiller la progression d'une maladie ou la reconstruction d'un muscle [8].
- **Surveillance de barrages** : on peut inclure sur les parois des barrages des capteurs qui permettent de calculer en temps réel la pression exercée. Il est donc possible de réguler le niveau d'eau si les limites sont atteintes. On peut aussi imaginer inclure des capteurs entre les sacs de sables formant une digue de fortune. La détection rapide d'infiltration d'eau peut servir à renforcer le barrage en conséquence. Cette technique peut aussi être utilisée pour d'autres constructions tels que ponts, voies de chemins de fer, routes de montagnes, bâtiments et autres ouvrages d'art.

## 2.4 Architecture d'un RCSF

Tous les capteurs respectent globalement la même architecture basée sur un noyau central autour duquel s'articulent les différentes interfaces d'entrée-sortie, de communication et d'alimentation [9, 10]. La figure 2 montre un exemple d'un réseau de capteurs.

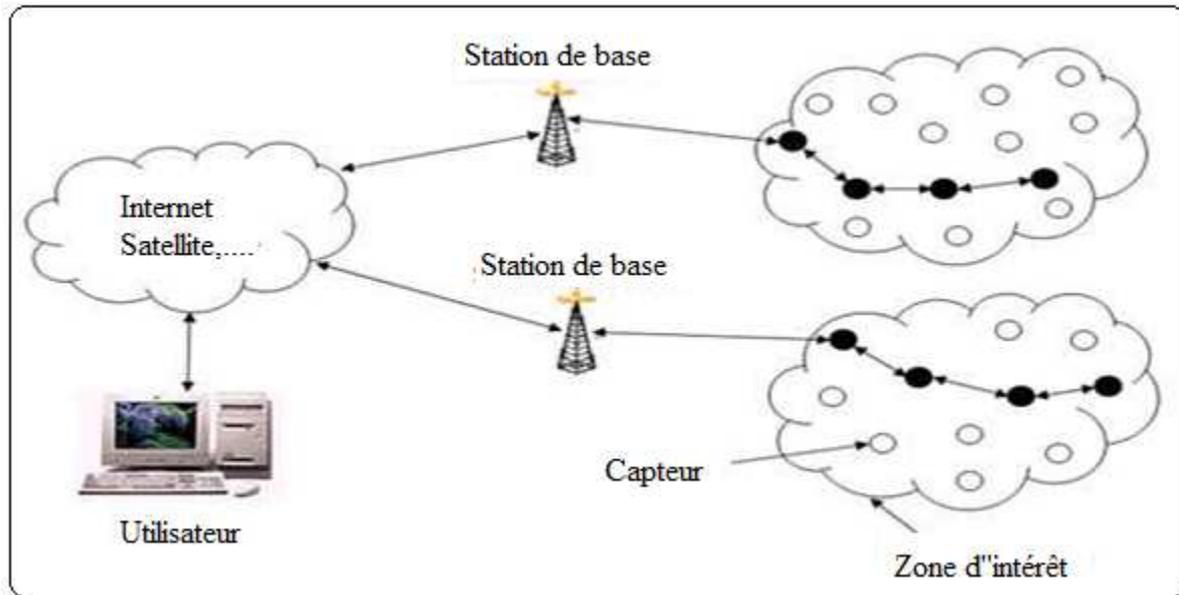


Figure 2 Exemple de réseaux de capteurs

Un RCSF est composé d'un ensemble de nœuds capteurs qui sont organisés en champs «Sensor Fields». Chacun de ces nœuds a la capacité de collecter des données et de les transférer au nœud passerelle par l'intermédiaire d'une architecture multi-sauts. Le nœud passerelle transmet ensuite ces données par Internet ou par satellite à l'ordinateur central «Gestionnaire de tâches» pour analyser ces données et prendre des décisions.

## 2.5 Contraintes de conception des RCSF

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit [1]:

- **La tolérance aux fautes** [11, 12] : la tolérance aux fautes est la capacité de maintenir les fonctionnalités du réseau en présence de fautes. La fiabilité des réseaux de capteurs sans fil est affectée par des défauts qui se produisent à cause de diverses raisons telles que le mauvais fonctionnement du matériel ou à cause d'un manque d'énergie. Ces problèmes n'affectent pas le reste du réseau.
- **Le facteur d'échelle** (Scalability) [13]: le nombre de nœuds de capteurs augmente sur un réseau sans fil et ce nombre peut atteindre le million. Un nombre aussi important de nœuds engendre beaucoup de transmissions entre les nœuds et peut imposer des difficultés pour le transfert de données.

- **Les coûts de production** [10]: souvent les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un nœud est critique afin de pouvoir concurrencer un réseau de surveillance traditionnel.
- **L'environnement** : les capteurs sont souvent déployés en masse dans des endroits tels que des champs de bataille, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement souillés [15],... Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées.
- **La topologie de réseau** [16]: le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases : déploiement, post-déploiement (les capteurs peuvent bouger, ne plus fonctionner,...) et redéploiement de nœuds additionnels.
- **Les contraintes matérielles** [17]: la principale contrainte matérielle est la taille du capteur. Les autres contraintes sont la consommation d'énergie qui doit être moindre pour que le réseau survive le plus longtemps possible, qu'il s'adapte aux différents environnements (fortes chaleurs, eau,..), qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles.
- **Les médias de transmission** : dans un réseau de capteurs, les nœuds sont reliés par une architecture sans fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de transmission doit être standardisé. On utilise le plus souvent l'infrarouge, le Bluetooth [18] et les communications radio Zig Bee [19].
- **La consommation d'énergie** [21]: un capteur, de par sa taille, est limité en énergie (<1.2V). Dans la plupart des cas le remplacement de la batterie est impossible. Ce qui veut dire que la durée de vie d'un capteur dépend grandement de la durée de vie de la batterie. Dans un réseau de capteurs (multi-sauts) chaque nœuds collecte des données et envoie/transmet des valeurs. Le dysfonctionnement de quelques nœuds nécessite un changement de la topologie du réseau et un re-routage des paquets. Toutes ces opérations sont gourmandes en énergie, c'est pour cette raison que les recherches

actuelles se concentrent principalement sur les moyens de réduire cette consommation [22].

## **2.6 Consommation d'énergie dans les RCSF**

La première étape dans la conception de système énergétique de capteurs consiste à analyser les caractéristiques de consommation d'énergie d'un nœud de capteur sans fil. Cette analyse systématique de l'énergie d'un nœud capteur est extrêmement importante pour identifier les problèmes dans le système énergétique pour permettre une optimisation efficace. L'énergie consommée par un capteur est principalement due aux opérations suivantes : la détection, le traitement et la communication [96].

### **2.6.1 Energie de capture**

Les sources de consommation d'énergie des nœuds pour les opérations de détection ou de capture sont : l'échantillonnage, la conversion analogique-numérique, le traitement de signal et l'activation de la sonde de capture [26].

### **2.6.2 Energie de traitement**

L'énergie de traitement est composée de deux sortes d'énergie: l'énergie de commutation et l'énergie de fuite. L'énergie de commutation est déterminée par la tension d'alimentation et la capacité totale commutée au niveau logiciel (en exécutant un logiciel). Par contre, l'énergie de fuite correspond à l'énergie consommée lorsque l'unité de calcul n'effectue aucun traitement. En général, l'énergie de traitement est faible par rapport à celle nécessaire pour la communication.

### **2.6.3 Energie de communication**

L'énergie de communication se décline en trois parties : l'énergie de réception, l'énergie de l'émission et l'énergie en état de veille. Cette énergie est déterminée par la quantité des données à communiquer et la distance de transmission, ainsi que par les propriétés physiques du module radio. L'émission d'un signal est caractérisée par sa

puissance ; quand la puissance d'émission est élevée, le signal aura une grande portée et l'énergie consommée sera plus élevée. Notons que l'énergie de communication représente la portion la plus grande de l'énergie consommée par un nœud capteur.

## **2.7 Techniques de minimisation de la consommation d'énergie**

Dans les réseaux ad hoc, la consommation de l'énergie a été considérée comme un facteur déterminant mais pas primordial car les ressources d'énergie peuvent être remplacées par l'utilisateur. Ces réseaux se focalisent plus sur la QoS (Quality of Service) que sur la consommation de l'énergie. Par contre, dans les réseaux de capteurs, la consommation d'énergie est très importante puisque généralement les capteurs sont déployés dans des zones inaccessibles. Ainsi, il est difficile voire impossible de remplacer les batteries après leur épuisement. De ce fait, la consommation d'énergie au niveau des capteurs a une grande influence sur la durée de vie du réseau.

Après la description des principales causes de consommation d'énergie dans les RCSF, nous présentons dans ce qui suit les différentes techniques utilisées pour minimiser cette consommation. Ces techniques sont appliquées soit au niveau de la couche liaison soit au niveau de la couche réseau. Le schéma suivant donne un aperçu global de ces mécanismes :

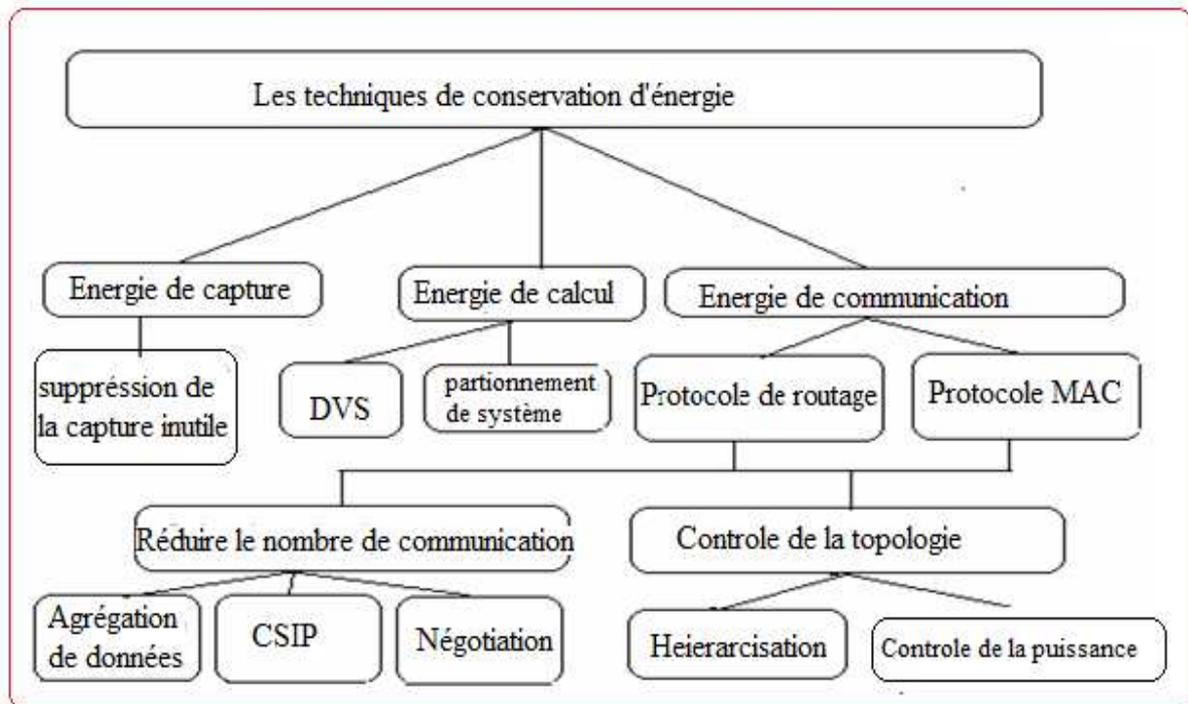


Figure 3 Les techniques de conservation d'énergie

L'énergie du capteur peut être économisée soit au niveau de la capture, au niveau de traitement ou au niveau de la communication.

A. La seule solution apportée pour la minimisation de la consommation d'énergie au niveau de la capture consiste à réduire les fréquences et les durées de captures.

B. L'énergie de calcul peut être optimisée en utilisant deux techniques :

- L'approche DVS (Dynamique Voltage Scaling) [43] qui consiste à ajuster de manière adaptative la tension d'alimentation et la fréquence du microprocesseur pour économiser la puissance de calcul sans dégradation des performances.
- L'approche de partitionnement de système qui consiste à transférer un calcul prohibitif en temps de calcul vers une station de base qui n'a pas de contraintes énergétiques et qui possède une grande capacité de calcul [44].

C. La minimisation de la consommation d'énergie pendant la communication est étroitement liée aux protocoles développés pour la couche réseau et la sous-couche MAC. Ces protocoles se basent sur plusieurs techniques : l'agrégation de données, la négociation et à la technique CSIP (Collaborative Signal and Information Processing).

Cette dernière technique est une discipline qui combine plusieurs domaines [45] : la communication et le calcul à basse puissance, le traitement de signal, les algorithmes distribués, la tolérance aux fautes, les systèmes adaptatifs et la théorie de fusion des capteurs et des décisions. Ces techniques ont le but de réduire le nombre d'émission/réception des messages.

## 2.8 Les critères de performance des protocoles de routage en RCSF

La performance des réseaux de capteurs sans fil est fondée sur les facteurs suivants :

- **Evolutivité** : l'évolutivité est un facteur important dans les réseaux de capteurs sans fil. Une zone de réseau n'est pas toujours statique, elle change selon les besoins des utilisateurs. Tous les nœuds dans le domaine du réseau doivent être évolutifs ou être en mesure de s'adapter aux changements dans la structure du réseau en fonction de l'utilisateur.
- **L'énergie** : chaque nœud utilise peu d'énergie pour des activités telles que la détection, le traitement, le stockage et la transmission. Un nœud dans le réseau doit savoir combien d'énergie sera utilisée pour effectuer une nouvelle tâche à laquelle il est soumis. L'énergie consommée peut varier selon le type de fonctionnalité ou l'activité qu'il a à accomplir.
- **Le temps de traitement**: il se réfère au temps pris par le nœud dans le réseau pour assurer l'ensemble de l'opération commençant par la détection, le traitement des données ou le stockage de données, la transmission ou la réception sur le réseau.
- **Le schéma de transmission**: la transmission de données par les nœuds de capteurs vers la destination ou la station de base se fait par un schéma de routage à un seul saut ou à multi-saut.
- **La capacité du réseau** : tous les nœuds du réseau de capteurs utilisent certaines ressources du réseau qui les aident à accomplir certaines activités comme la détection ou la transformation.
- **Synchronisation** : dans les communications radio entre les nœuds de capteurs d'un WSN, les capteurs écoutent en permanence les transmissions et consomment de l'énergie s'ils ne

sont pas synchronisés les uns les autres. Pour cela, un nœud doit avoir la même notion de temps pour se mettre en veille et se réveiller que ses voisins.

- **Contrôle de paquets:** un paquet envoyé avant la transmission entre deux nœuds est appelé le paquet de contrôle. Le paquet de contrôle contient le nombre de bits de données envoyés, l'adresse du nœud de destination et certaines informations qui contribuent à éviter les collisions pendant la transmission.

## 2.9 Les principaux protocoles de routage pour les réseaux ad hoc

Les RCSF partagent avec les MANET (Mobile Ad hoc NETWORKS) plusieurs propriétés en commun, telles que l'absence d'infrastructure et les communications sans fil. Mais l'une des différences clé entre les deux architectures est le domaine d'application.

Les réseaux ad hoc, dans leur configuration mobile, sont connus sous le nom de MANET (pour *Mobile Ad hoc NETWORKS*) [41]. Chaque nœud communique directement avec son voisin et pour communiquer avec d'autres nœuds, il lui est nécessaire de faire passer les données par d'autres nœuds qui se chargeront de les acheminer. Pour cela, il est primordial que les nœuds se situent les uns par rapport aux autres et soient capables de construire des liens entre eux, c'est le rôle du protocole de routage.

Suivant la manière dont sont créées et maintenues les routes lors de l'acheminement des données, les protocoles de routage peuvent être séparés en deux catégories, les protocoles proactifs et les protocoles réactifs. Les protocoles proactifs établissent les routes à l'avance en se basant sur l'échange périodique des tables de routage, alors que les protocoles réactifs cherchent les routes à la demande.

### 2.9.1 Les protocoles de routage proactifs

Les protocoles de routage proactifs pour les réseaux mobiles ad hoc sont basés sur la même philosophie que les protocoles de routage utilisés dans les réseaux filaires conventionnels. Les deux principales méthodes utilisées dans cette classe de protocoles proactifs sont:

- La méthode Etat de Lien ("Link State") [102].
- La méthode du Vecteur de Distance ("Distance Vector") [102].

Ces méthodes sont utilisées aussi dans les réseaux sans fil. Parmi les protocoles de routage proactifs les plus connus on citera le DSDV, FSR, OLSR [54]

### 2.9.1.1 Le protocole de routage «DSDV»

Destination Sequence Distance Vector (DSDV) est un protocole proactif unicast mobile ad hoc qui est basé sur l'algorithme de Bellman-Ford [97]. Dans les tables de routage de DSDV on trouve :

- Toutes les destinations possibles.
- Le nombre de nœuds (ou de sauts) nécessaire pour atteindre la destination.
- Le numéro de séquences (SN : sequence number) qui correspond à un nœud destination.

Les numéros de séquence sont utilisés dans DSDV pour distinguer les anciennes et nouvelles routes et pour éviter la formation de boucles de parcours. Chaque nœud transmet périodiquement des mises à jour, y compris des informations de routage à ses voisins immédiats.

### 2.9.1.2 Le protocole de routage « FSR »

Le protocole FSR (Fisheye State Routing) [55] est basé sur l'utilisation de la technique "œil de poisson" (fisheye) et utilisé dans le but de réduire le volume d'information nécessaire pour représenter les données graphiques [56, 85]. Dans la pratique, l'œil d'un poisson capture avec précision, les points proches du point focal. La précision diminue quand la distance, séparant le point vu et le point focal augmente.

Dans le contexte du routage, l'approche du "fisheye" matérialise, pour un nœud, le maintien des données concernant la précision de la distance et la qualité du chemin d'un voisin direct, avec une diminution progressive du détail et de la précision, quand la distance augmente.

La diminution de la précision est assurée en changeant les fréquences de mise à jour, et cela en utilisant des périodes d'échanges différentes pour les différentes entrées de la table de routage. Les entrées qui correspondent aux nœuds les plus proches sont envoyées aux voisins avec une fréquence élevée et donc avec une période d'échange relativement petite.

### 2.9.1.3 Le protocole de routage « OLSR »

Le protocole OLSR (Optimized Link State Routing Protocol) [98] est un protocole à état de lien optimisé qui utilise les routes de plus court chemin. Alors que dans un protocole à état de lien, chaque nœud déclare ses liens directs avec ses voisins à tout le réseau, dans le cas d'OLSR, les nœuds ne déclarent qu'une sous-partie de leur voisinage grâce à la technique des relais multipoints. Ils consistent essentiellement, en un nœud donné, à ignorer un ensemble de liens et de voisins directs, qui sont redondants pour le calcul des routes de plus court chemin. Plus précisément, dans l'ensemble des voisins d'un nœud, seul un sous-ensemble des ces voisins est considéré comme pertinent. Ils sont choisis de façon à pouvoir atteindre tout le voisinage à deux sauts (tous les voisins des voisins). Cet ensemble est appelé l'ensemble des relais multipoints.

Ces relais multipoints sont utilisés de deux façons : pour diminuer le trafic dû à la diffusion des messages de contrôle dans le réseau et aussi pour diminuer le sous-ensemble des liens diffusés à tout le réseau puisque les routes sont construites à base des relais multipoint.

La diffusion d'un message à tout le réseau, par répétition, peut se faire par l'inondation classique : un nœud retransmet un message si et seulement si il ne l'a pas déjà reçu. La diffusion par relais multipoints diminue le nombre de retransmissions en utilisant la règle suivante : un nœud retransmet un message si et seulement si

- 1- il ne l'avait pas déjà reçu
- 2- il vient de le recevoir d'un nœud dont il est un relais multipoint.

## 2.9.2 Les protocoles de routage réactifs

Les protocoles de routage appartenant à cette catégorie créent et maintiennent les routes selon les besoins. Lorsqu'une source a besoin d'une route, une procédure de découverte globale de routes est lancée.

### 2.9.2.1 Le protocole de routage « DSR »

Le protocole "Routage à Source Dynamique" (DSR) [99] est basé sur l'utilisation de la technique "routage source". Dans cette technique, la source des données détermine la séquence complète des nœuds à travers lesquelles les paquets de données seront envoyés.

Un site initiateur de l'opération de « découverte de routes » diffuse un paquet requête de route. Si l'opération de découverte est réussite, l'initiateur reçoit un paquet réponse de route qui liste la séquence de nœuds à travers lesquelles la destination peut être atteinte. Le paquet requête de route contient donc un champ enregistrement de route dans lequel sera accumulée la séquence des nœuds visités durant la propagation de la requête dans le réseau.

L'utilisation de la technique "routage source" fait que les nœuds de transit n'aient pas besoin de maintenir les informations de mise à jour pour envoyer les paquets de données, puisque ces derniers contiennent toutes les décisions de routage.

Dans ce protocole, il y a une absence totale de boucle de routage car le chemin source-destination fait partie des paquets de données envoyés.

### 2.9.2.2 Le protocole de routage « AODV »

Le protocole AODV (Ad hoc On-demand Distance Vector) [100] représente essentiellement une amélioration de l'algorithme DSDV dans le contexte réactif. Il est spécialement conçu pour les réseaux mobiles pour créer et découvrir les liaisons entre la source et la destination [72, 73]. Il est utilisé pour des routages unicast et multicast en utilisant des requêtes de type (**route request / route reply**).

Avec AODV, chaque nœud a une table de routage qui donne des informations sur ses voisins, la table joue un rôle dans le choix d'un voisin qui va transmettre les paquets de la

source vers la destination. Lorsque la source a des données à envoyer vers une destination, elle diffuse une requête de type Route Request (RREQ). Lorsque le nœud reçoit RREQ, il met à jour ses informations pour le nœud source et il ajoute une nouvelle route valide à sa table de routage pour atteindre la source qui a envoyé RREQ. Lorsque RREQ arrive à la destination, celle-ci génère une réponse de type Route Reply (RREP). RREP est renvoyé vers la source comme le montre la figure 4. Chaque nœud possède un numéro de séquence qui permet de choisir la route la plus récente et de maintenir la consistance des informations de routage [75].

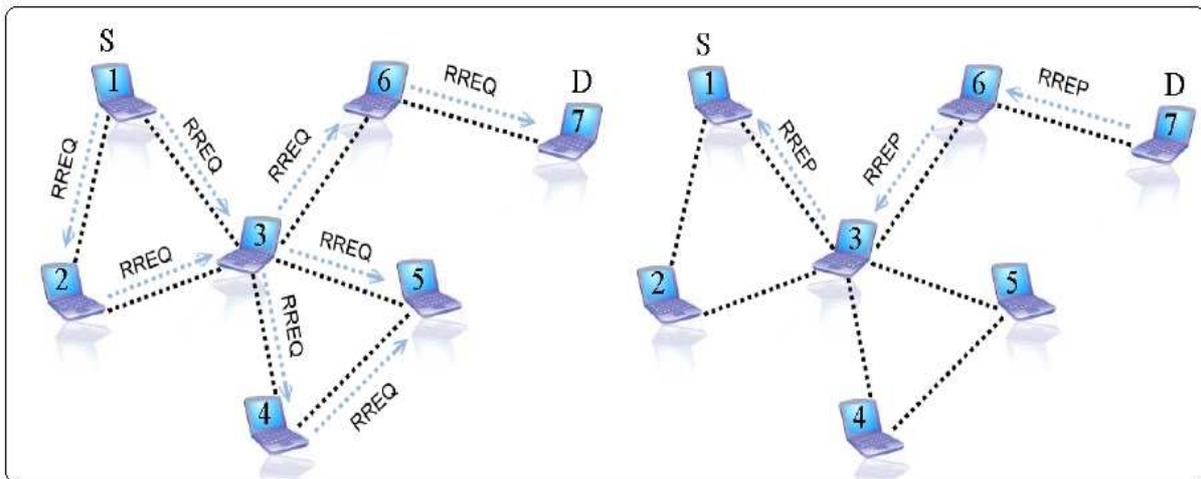


Figure 4 Les deux requêtes RREQ et RREP en AODV

Avec AODV, s'il existe plusieurs routes possibles de la source vers la destination, AODV choisit la route la plus courte (la route où il y a un minimum de sauts). Si un tour de routage échoue, la source relance un nouveau RREQ avec un temps T plus important. Si plusieurs séries de Route Request échouent, alors aucune route ne peut être trouvée [77, 84].

## 2.10 Les principaux protocoles de routage dans les RCSF

La figure suivante résume les principaux protocoles de routage [23]:

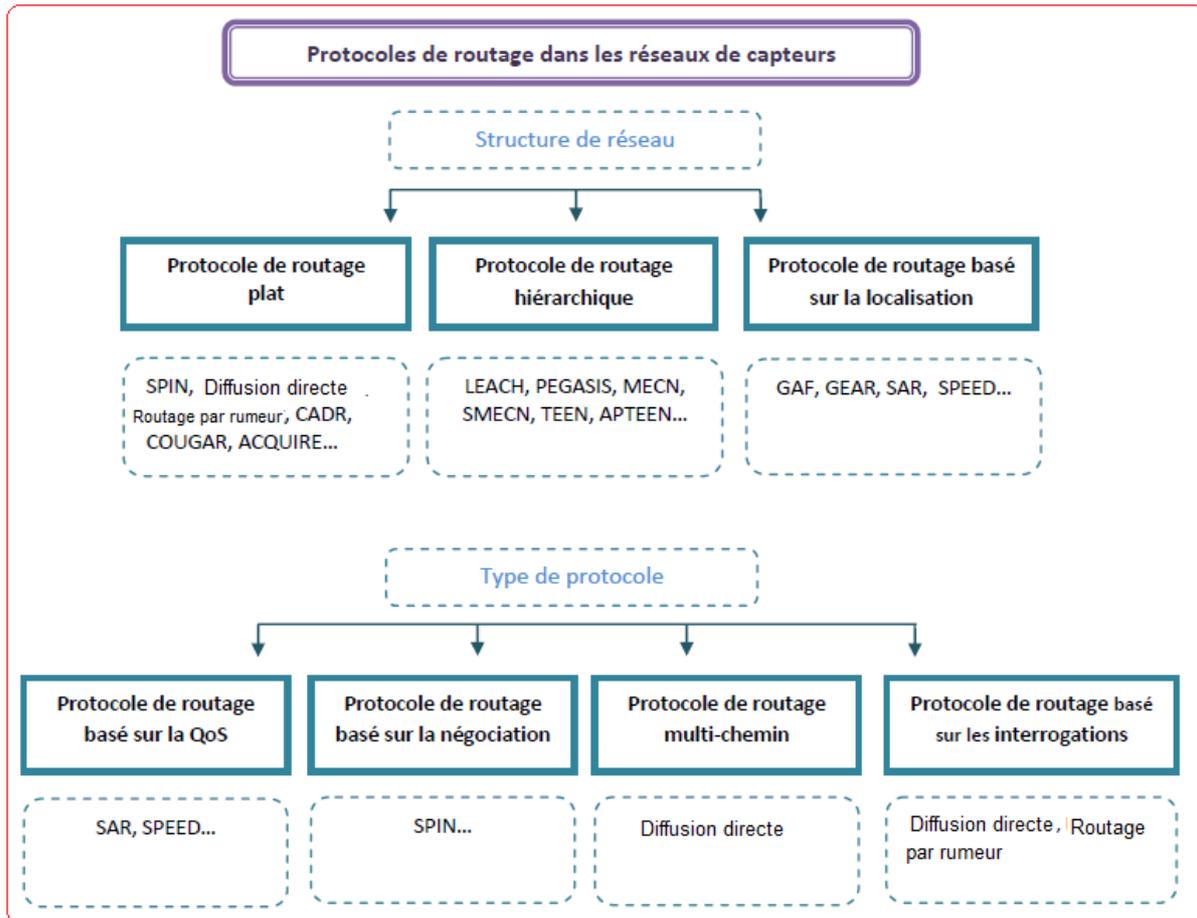


Figure 5 Les principaux protocoles de routages dans les RCSF

### 2.10.1 Les Protocoles hiérarchiques

L'objectif principal du routage hiérarchique [24, 25] est de maintenir efficacement la consommation d'énergie de nœuds de capteurs en les impliquant dans la communication multi-hop au sein d'un cluster et en effectuant l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis à la destination. La formation de clusters est généralement fondée sur la réserve d'énergie des capteurs et sur les capteurs qui sont à proximité de cluster-head (voir figure 6). LEACH (Low Energy Adaptive Clustering Hierarchical) [23] est l'une de premières approches de routage pour les réseaux de capteurs. L'idée proposée par LEACH a été une inspiration pour de nombreux protocoles de routage hiérarchique, bien que certains protocoles aient été développés de manière indépendante.

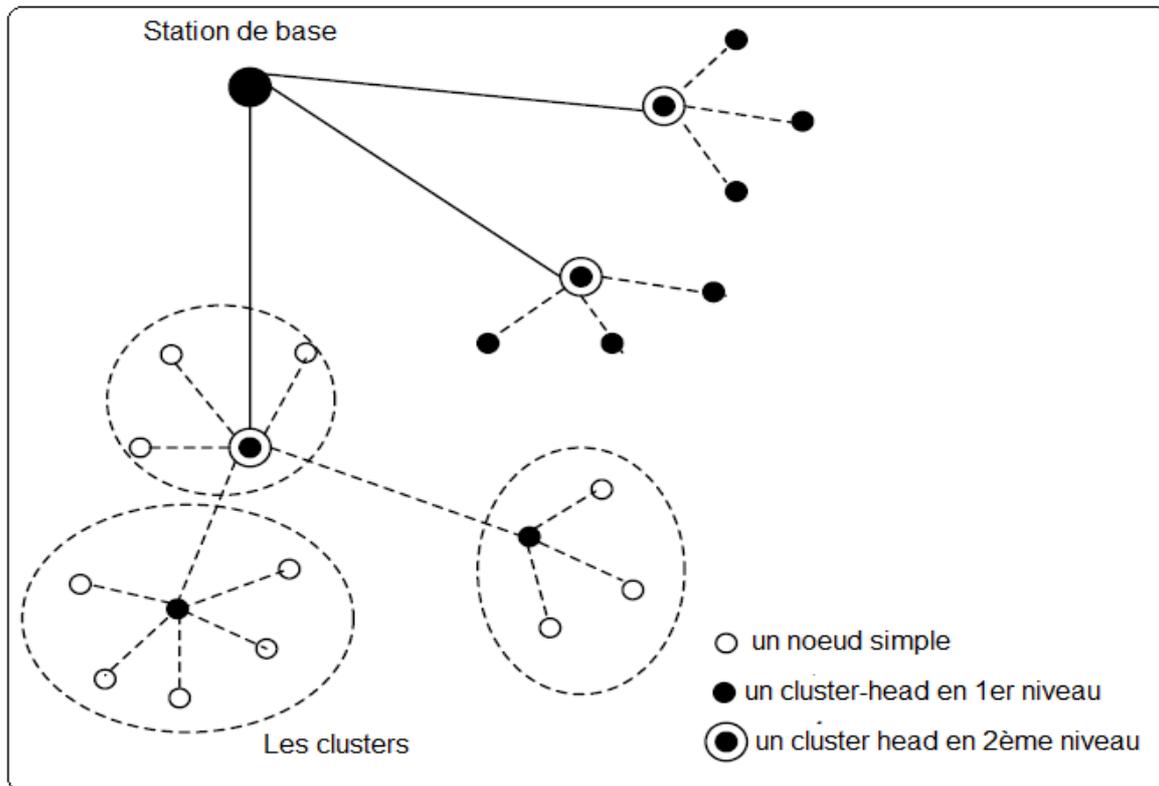


Figure 6 Topologie hiérarchique

### 2.10.1.1 Le protocole de routage «LEACH »

LEACH est l'un des algorithmes de routage hiérarchique le plus populaire pour les réseaux de capteurs [23, 24, 26]. L'idée est de former des clusters de nœuds de capteurs basés sur les zones où il y a un fort signal reçu, puis utiliser des clusters-heads locaux comme passerelle pour atteindre la destination. Cela permet d'économiser de l'énergie car les transmissions ne sont effectuées que par les cluster-head plutôt que par tous les nœuds de capteurs.

### 2.10.1.2 Les protocoles de routage «PEGASIS & Hierarchical-PEGASIS»

Power-Efficient GATHERing in Sensor Information Systems (PEGASIS) [24] est une version améliorée du protocole LEACH. PEGASIS forme des chaînes plutôt que des clusters de nœuds de capteurs afin que chaque nœud transmette et reçoive uniquement des données d'un voisin. Un seul nœud est sélectionné à partir de cette chaîne pour transmettre à la station de base. L'idée de PEGASIS est qu'il utilise tous les nœuds pour transmettre ou recevoir des

données avec ses plus proches voisins. Il déplace les données reçues de nœud à nœud, puis les données seront agrégées jusqu'à ce qu'elles atteignent tous la station de base. Donc, chaque nœud du réseau est tour à tour un chef de file de la chaîne, ainsi que responsable pour transmettre l'ensemble des données recueillies et fusionnées par la chaîne de nœuds au niveau de la station de base [27].

### **2.10.1.3 Les protocoles de routage «TEEN et APTEEN»**

Les protocoles Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [28] et Adaptive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) [29] conviennent pour les applications critiques. Dans les deux protocoles, le facteur clé est la valeur de l'attribut mesuré. La caractéristique supplémentaire d'APTEEN est la capacité de changer la périodicité et les paramètres de TEEN en fonction des besoins des utilisateurs et des applications.

TEEN est conçu pour être sensible à des changements soudains des attributs tels que la température. La réactivité est importante pour les applications critiques dont le réseau fonctionne dans un mode réactif. L'architecture du réseau de capteurs est basée sur un groupement hiérarchique où les nœuds forment des clusters et ce processus va se répéter jusqu'à ce que la station de base soit atteinte [24].

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

### **2.10.2 Les protocoles de routage basés sur la localisation**

Les protocoles de routage basés sur la localisation [24] utilisent les informations d'emplacement pour guider la découverte de routage et la transmission des données. Ils permettent la transmission directionnelle de l'information en évitant l'inondation d'information dans l'ensemble du réseau. Par conséquent, le coût de contrôle de l'algorithme est réduit et le

routage est optimisé. De plus, avec la topologie réseau basée sur des informations de localisation de nœuds, la gestion du réseau devient simple.

L'inconvénient de ces protocoles de routage est que chaque nœud doit connaître les emplacements des autres nœuds.

#### **2.10.2.1 Le protocole de routage « MECN »**

Minimum Energy Communication Network (MECN) [32] est un protocole de routage qui cherche à établir et à entretenir une énergie minimale pour les réseaux sans fil en utilisant des GPS de faible puissance. MECN utilise une station de base comme destination de l'information, ce qui est toujours le cas pour les réseaux de capteurs. MECN identifie une région de relais pour chaque nœud. La région de relais se compose de nœuds dans une zone périphérique où la transmission à travers ces nœuds est plus économe en énergie que la transmission directe. L'idée principale de MECN est de trouver un sous-réseau qui a moins de nœuds et qui nécessite moins d'énergie pour la transmission entre deux nœuds quelconques. Cela est effectué en utilisant une recherche localisée pour chaque nœud en prenant en considération sa région de relais [24].

#### **2.10.2.2 Le protocole de routage « GAF »**

GAF (Geographic Adaptive Fidelity) [33] est un protocole de routage basé sur la localisation des nœuds. Il est conçu principalement pour les réseaux mobiles ad hoc, mais peut être applicable aux réseaux de capteurs. La localisation des nœuds dans GAF pourrait être fournie à l'aide d'un GPS ou d'autres techniques de localisation [24, 34, 49]. Il consiste à former des grilles virtuelles de la zone concernée en partitionnant cette zone où les nœuds sont déployés en de petites zones telles que, pour deux grilles adjacentes  $G_x$  et  $G_y$ , tous les nœuds de  $G_x$  peuvent communiquer avec tous les nœuds  $G_y$ . Ainsi, ce système de partitionnement GAF assure la fidélité du routage car il existe au moins un chemin entre un nœud et la station de base.

GAF peut augmenter considérablement la durée de vie du réseau. En effet, un seul nœud dans chaque grille reste à l'état actif en faisant passer les autres nœuds de la grille à l'état de sommeil pour une certaine période de temps tout en assurant la fidélité du routage.

Cependant, dans certains environnements où les nœuds sont fortement mobiles, la fidélité du routage pourrait être réduite si un nœud actif quitte la grille. Ainsi, le nombre de données perdues sera important.

### **2.10.2.3 Le protocole de routage «GEAR »**

Le protocole de routage GEAR (Geographic and Energy Aware Routing) [35, 36, 50] a été suggéré par Y. Yu, D. Estrin. Il consiste à utiliser l'information géographique lors de la diffusion des requêtes aux régions cibles car les requêtes contiennent souvent des données géographiques. L'idée est de restreindre le nombre de données dans la diffusion dirigée en prenant en considération uniquement une certaine région, plutôt que d'envoyer les données à l'ensemble du réseau.

Avec le protocole GEAR, chaque nœud maintient le coût pour atteindre la destination en passant par ses voisins. Ce coût est divisé en deux parties : un coût estimé et un coût d'apprentissage. Le coût estimé est une combinaison de l'énergie résiduelle et de la distance jusqu'à destination. Le coût d'apprentissage est un raffinement du coût estimé qu'un nœud dépense pour le routage autour des trous dans le réseau. Un trou se forme quand un nœud n'a pas de voisin proche par lequel il peut atteindre la région cible. S'il n'y a pas de trous, le coût estimé est égal au coût d'apprentissage. Le coût d'apprentissage se propage d'un saut à chaque fois qu'un paquet atteint la destination [24, 35].

### **2.10.3 Les protocoles de routage 'data-centric'**

Dans de nombreuses applications de réseaux de capteurs, vu le nombre élevé de nœuds déployés, il n'est pas possible d'attribuer des identificateurs globaux à chaque nœud. Cette absence d'identification globale avec le déploiement aléatoire de nœuds de capteurs font qu'il est difficile de sélectionner un ensemble spécifique de nœuds de capteurs à interroger. Par

conséquent, les données sont généralement transmises de chaque nœud de capteurs dans la région de déploiement avec une redondance importante. Cette réflexion a conduit au routage data-centric [95] qui est différent du traditionnel routage où les routes sont créées entre les nœuds adressables gérée dans la couche réseau. Le destinataire envoie des requêtes à certaines régions et attend à recevoir des données provenant des capteurs situés dans les régions sélectionnées. Comme les données sont demandées à travers des requêtes, le nommage est nécessaire pour préciser les propriétés des données [24].

Comme la montre l'exemple d'une approche data-centric dans la figure 7, les données provenant des deux sources sont agrégées au nœud B. Ensuite, la donnée combinée (1+2) est envoyée de B vers la destination.

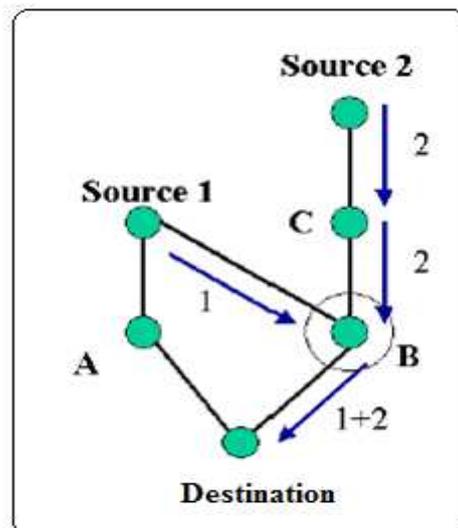


Figure 7 Le routage data-centric

### 2.10.3.1 Le protocole de routage « SPIN »

Un réseau de capteurs sans fil se compose de plusieurs capteurs qui sont déployés dans différentes régions. L'accès aux données d'un événement particulier pour une région ou une zone spécifique peut faire une grande différence. Des mesures peuvent être prises pour l'obtention de données à partir d'un endroit ciblé. Les données sont recueillies auprès de tous les capteurs et transmises de manière redondante sur le réseau, ce qui entraîne une utilisation

inefficace de l'énergie. Afin de résoudre ces types de problèmes, les systèmes de routage data-centric ont évolué : la destination envoie des requêtes pour les capteurs du réseau dans un endroit choisi. Les attributs sont utilisés pour demander des données provenant des capteurs. SPIN (Sensor Protocols for Information via Negotiation) [37] est le premier protocole data-centric qui a été conçu pour les réseaux de capteurs sans fil. Il présente de nombreuses similitudes par rapport à la diffusion dirigée. Il est efficace dans la réduction des données redondantes et économise l'énergie [24].

La diffusion est le processus de collecte des observations de l'ensemble des capteurs individuel qui sont déployées dans le réseau et où tous les capteurs sont traités comme des nœuds destination [38]. Les tâches confiées à ces capteurs sont de recueillir le point de vue complet de l'environnement sous la forme de données et de renforcer une structure de réseau avec une tolérance aux fautes. La consommation d'énergie durant les calculs et les communications doit être contrôlée afin de prolonger la durée de vie des capteurs au sein du réseau.

### **2.10.3.2 La diffusion dirigée**

La diffusion dirigée [39, 40] est un protocole important dans le routage data-centric des réseaux de capteurs. L'idée vise à diffuser des données aux nœuds en utilisant un schéma de nommage pour les données. La raison principale derrière l'utilisation d'un tel système est de se débarrasser des opérations inutiles de routage de couche réseau afin d'économiser l'énergie. La diffusion dirigée suggère l'utilisation de paires attribut-valeur pour les données et les requêtes des capteurs. Afin de créer une requête, un nœud est défini à l'aide d'une liste de paires attribut-valeur comme le nom des objets, l'intervalle, la durée, la zone géographique, etc. Un paquet est diffusé par ce nœud vers la destination à travers ses voisins. Chaque nœud qui reçoit les paquets peut les stocker pour une utilisation ultérieure. Les paquets stockés sont ensuite utilisés pour comparer les données reçues. La requête contient aussi plusieurs champs de gradient. Un gradient est un lien réponse avec un voisin dont le paquet a été reçu et qui est caractérisé par le débit, la durée et la date d'expiration de données. Ainsi, en utilisant les

intérêts et les gradients, les routes sont établies entre la destination et les sources. Plusieurs routes peuvent être établies de telle sorte que l'une d'elle est choisie par renforcement. La destination renvoie le message d'intérêt initial à travers la route choisie. Un intervalle plus petit renforce donc le nœud source sur ce chemin pour envoyer des données plus fréquemment [24].

### **2.10.3.3 Le protocole de routage par rumeur**

Le routage par rumeur [41, 42] est principalement destiné pour des applications où le routage géographique n'est pas faisable. En général, la méthode (diffusion dirigée) utilise l'inondation pour envoyer la requête à l'ensemble du réseau où il n'y a pas de critère géographique pour diffuser les tâches. Toutefois, dans certains cas, peu de données sont demandées par les nœuds, donc l'utilisation d'inondation est inutile [24].

L'idée clé de cette méthode est de trouver les routes pour les requêtes vers les nœuds qui ont observé un événement particulier, plutôt que d'inonder tout le réseau pour récupérer des informations sur les événements survenus. Afin de diffuser un événement sur le réseau, l'algorithme de routage par rumeur emploie des paquets appelés agents. Quand un nœud détecte un événement, il ajoute cet événement à sa table locale, appelée table d'événements et génère un agent. Cet agent parcourt le réseau afin de propager des informations sur des événements locaux pour les nœuds distants. Quand un nœud génère une requête pour un événement, les nœuds qui connaissent le chemin, répondent à la requête en inspectant leur table événement. Par conséquent, il n'est pas nécessaire d'inonder tout le réseau, ce qui réduit le coût de communication. D'autre part, ce routage n'utilise qu'un seul chemin entre la source et la destination au lieu de la diffusion dirigée où les données peuvent être acheminées par des routes multiples.

Les résultats de simulation ont montré que le routage par rumeur peut réaliser des économies d'énergie significatives par rapport à la méthode d'inondation et peut également préserver la vie du nœud. Toutefois, le routage par rumeur fonctionne bien uniquement

lorsque le nombre d'événements est faible. Pour un grand nombre d'événements, le coût du maintien des agents et des tables d'événements de chaque nœud devient impossible.

La table 1 représente un comparatif de quelques protocoles de routage dans les réseaux de capteurs sans fil.

	Classification	Mobilité	Basé sur la négociation	Agrégation	Mise à l'échelle	Multi-chemin	L'utilisation d'énergie
LEACH	Hiérarchique	Station de base fixe	Non	Oui	Bonne	Non	Max
PEGASIS	Hiérarchique	Station de base fixe	Non	Non	Bonne	Non	Max
Hierarchical-PEGASIS	Hiérarchique	Station de base fixe	Non	Non	Bonne	Non	Max
TEEN and APTEEN	Hiérarchique	Station de base fixe	Non	Oui	Bonne	Non	Max
MECN	Hiérarchique	Non	Non	Non	Faible	Non	Min
GAF	Géographique	Limitée	Non	Non	Bonne	Non	Limitée
GEAR	Géographique	Limitée	Non	Non	Limitée	Non	Limitée
SPIN	Plat	Possible	Oui	Oui	Limitée	Oui	Limitée
La diffusion dirigée	Plat	Possible	Oui	Oui	Limitée	Oui	Limitée
routage par rumeur	Plat	Très limitée	Non	Oui	Bonne	Non	Min
GBR	Plat	Limitée	Non	Oui	Limitée	Non	Min

Table 1. Classification et comparaison des protocoles de routages dans les réseaux de capteurs

## 2.11 Les principaux algorithmes de conservation d'énergie

De nombreux algorithmes de routage ont été spécifiquement conçus pour les réseaux de capteurs où la consommation d'énergie est un facteur essentiel. Ce facteur a posé de

nombreux défis à la conception et à la gestion des réseaux de capteurs. Ces défis nécessitent une gestion efficace de l'énergie pour toutes les couches de la pile de protocole réseau. Les problèmes liés aux couches liaison et physique sont généralement communs pour les différents types d'applications de capteurs. Pour la couche réseau, l'objectif principal est de trouver des moyens pour une mise en œuvre efficace de l'énergie et pour une diffusion fiable des données de la source vers la destination de sorte que la durée de vie du réseau soit maximisée.

Quelques algorithmes ont été développés pour prendre soin du contrôle de la capacité. Certains se concentrent sur l'utilisation effective du temps d'activation des nœuds des capteurs [46]. L'auteur de [47] a développé un algorithme de diffusion efficace de l'énergie pour les réseaux de capteurs sans fil qui porte sur les techniques conservatrices pour diffuser des messages afin que les économies d'énergie puissent être réalisées. Il a souligné également que la technique de routage pour une énergie efficace pourrait améliorer l'efficacité du réseau [48]. Plusieurs paramètres tels que la distance de transmission, le nombre de sauts et le retard ont été pris en compte pour les économies d'énergie dans les réseaux de capteurs.

La recherche dans le domaine des protocoles d'énergie efficaces dans les réseaux de capteurs sans fil est relativement nouvelle. L'objectif principal de tous ces protocoles et algorithmes est de trouver les routes qui sont économes en énergie et donc de maximiser la durée de vie du réseau. Pour atteindre cet objectif, de nombreux protocoles ont été développés. Ces protocoles utilisent des stratégies différentes pour obtenir leurs routes. Certains de ces protocoles utilisent le clustering, des fonctions, des équations, des arbres de routage optimal, des recherches multi-paths ou un mécanisme efficace pour détourner les trous. Par exemple, le protocole LEACH utilise les clusters-head comme des routeurs pour assurer une communication avec une énergie efficace. Ainsi, l'énergie sera préservée car la communication sera effectuée uniquement entre les clusters-head.

Les algorithmes d'énergie efficace les plus connues sont:

### **2.11.1 L'algorithme de routage « EARLEAHSN »**

L'algorithme Energy Aware Routing for Low Energy Ad Hoc Sensor Networks (EARLEAHSN) [49] utilise un ensemble de sous-chemins optimaux afin d'augmenter la durée de vie du réseau. Ces chemins sont choisis au moyen d'une fonction de probabilité qui dépend de la consommation énergétique de chaque route [24]. La survie du réseau est la mesure principale de cette approche qui propose d'éviter l'utilisation permanente de la route la plus économe en énergie car cela épuise l'énergie des nœuds sur cette route. Au lieu de cela, l'un des trajets multiples est utilisé avec une certaine probabilité de sorte que la vie entière du réseau se prolonge. Le protocole suppose que chaque nœud est adressable par le biais d'une classe d'adresse qui comprend l'emplacement et les types de nœuds.

### **2.11.2 L'algorithme de routage «EARCBSN »**

L'algorithme Energy-Aware Routing in Cluster-Based Sensor Networks (EARCBSN) [48] propose un algorithme de routage hiérarchique basé sur une architecture à trois niveaux. Les capteurs sont regroupés en clusters avant l'exploitation du réseau. L'algorithme emploie les clusters-head comme des passerelles et ces clusters-head possèdent de l'énergie plus que les autres capteurs et connaissent l'emplacement des tous les capteurs.

Le routage nécessite une maintenance d'un cluster-head qui inclut tous les paramètres qui influent sur la décision de routage. Dans cet algorithme, ces paramètres sont l'état du capteur, sa localisation, l'énergie restante et le trafic des messages. Il y a une certaine imprécision dans le modèle d'énergie des passerelles dûe à la surcharge, la perte des paquets et au retard de propagation des messages [51, 52]. Le nœud passerelle agit comme un gestionnaire de cluster de réseau centralisé qui achemine les routes pour les données des capteurs, qui contrôle la latence dans tout le cluster et qui arbitre les accès entre les capteurs. Le nœud passerelle trace l'utilisation d'énergie de chaque nœud des capteurs et il contrôle aussi les changements dans l'environnement. De plus, il permet de configurer les capteurs et le réseau efficacement afin de prolonger la vie du réseau.

### 2.11.3 L'algorithme de routage « GBR »

L'algorithme Gradient-Based Routing (GBR) [53] ou le routage par gradient est une version légèrement modifiée de la diffusion dirigée [24]. L'idée de ce protocole est de maintenir le nombre de sauts lorsque le paquet est diffusé à travers le réseau. Ainsi chaque nœud peut découvrir le nombre minimal de sauts jusqu'à la destination. Ce nombre est appelé hauteur du nœud. La différence entre la hauteur d'un nœud et celui de son voisin est considérée comme le gradient sur ce lien. Un paquet est transmis sur un lien avec le gradient le plus grand. Le routage par gradient vise à utiliser certaines techniques auxiliaires telles que l'agrégation des données afin d'équilibrer le trafic de manière uniforme sur le réseau.

Trois techniques différentes pour gérer les données ont été présentées:

- **Stochastic Scheme** : quand il y a deux sauts ou plus avec le même gradient, le nœud choisit l'un d'eux au hasard.
- **Energy-based scheme** : lorsque l'énergie d'un nœud tombe en dessous d'un certain seuil, il augmente sa hauteur afin que les autres capteurs soient découragés d'envoyer des données à ce nœud.
- **Stream-based scheme** : l'idée est de détourner les nouveaux flux à partir de nœuds qui font actuellement parti de la trajectoire des autres filières

Les données s'efforcent alors de parvenir à une répartition égale de la circulation à travers l'ensemble du réseau, ce qui contribue à équilibrer la charge sur les nœuds des capteurs et augmente la durée de vie du réseau. Les techniques employées pour équilibrer la charge de trafic et la fusion de données sont également applicables aux autres protocoles de routage pour des performances améliorées.

## 2.12 Conclusion

Dans la première partie de ce chapitre, nous avons défini ce qu'est un réseau de capteurs sans fil que nous avons considéré comme un type particulier de réseau ad hoc. Puis, nous

avons décrit brièvement un réseau de capteur, ses applications, son architecture, ses principales contraintes de conception et la consommation d'énergie.

Dans la deuxième partie de ce chapitre, nous avons mis l'accent sur les principaux protocoles de routage dans les réseaux ad hoc et de capteurs. Nous avons résumé les protocoles de routage dans les réseaux de capteurs et classé les approches en trois catégories principales : les protocoles hiérarchiques, les protocoles basés sur la localisation et les protocoles data-centric. Enfin, Nous nous sommes concentrés sur les protocoles de routage qui s'intéressent à la consommation d'énergie.

Dans le prochain chapitre, nous présentons certaines techniques de traitement d'image que nous allons exploiter dans le routage des réseaux de capteurs sans fil.

# Chapitre3

## Techniques de traitement d'image et modélisation

### Sommaire

---

3.1	Introduction .....	39
3.2	Techniques de traitement d'image .....	39
3.2.1	Représentation d'une image en niveau de gris .....	39
3.2.2	Histogramme .....	40
3.2.3	Détection de contours .....	41
3.2.4	Le gradient d'une image .....	41
3.2.5	Les filtres en traitement d'image .....	42
3.2.5.1	Les filtres de convolution .....	43
3.2.5.2	Exemples des filtres en traitement d'image.....	44
3.2.5.2.1	Exemple de filtre passe-bas .....	44
3.2.5.2.2	Exemple de filtre moyen .....	45
3.2.5.2.3	Exemple de filtre gradient .....	46
3.2.5.2.4	Exemple de filtre Sobel .....	47
3.3	Modélisation .....	49
3.3.1	Modèle du réseau .....	49
3.3.2	Modèle énergétique .....	49
3.3.3	Analogie entre les RCSF et le niveau de gris d'une image .....	50
3.3.3.1	Les secteurs d'énergie .....	51
3.3.3.2	La matrice M .....	52
3.3.3.3	Le produit de convolution .....	52

3.3.3.3.1	Le filtre moyen.....	53
3.3.3.3.2	Le filtre de Sobel.....	54
3.3.4	Étude sur la direction du gradient d'énergie dans les RCSF .....	58
3.4	Conclusion .....	70

## 3.1 Introduction

Ces dernières décennies, l'image est devenue l'un des outils d'investigation les plus utilisés dans la recherche scientifique et technique. Grâce à sa nouvelle forme numérique et à son signal (lumineux, électrique ou magnétique), l'image peut être exploitée dans différents domaines d'applications [86].

Ce chapitre se divise en deux parties dont la première est une représentation de l'image en niveau de gris avec un aperçu de certaines techniques de traitement d'image comme l'histogramme, la détection de contours, le gradient d'une image, les filtres, etc. Dans la deuxième partie nous donnons une présentation du modèle du réseau et du modèle énergétique qui seront utilisés dans nos contributions dans le prochain chapitre. Ensuite nous avons créé une analogie entre un réseau de capteurs et le niveau de gris d'une image afin de traiter le routage basé sur l'énergie dans les réseaux de capteurs sans fil. Dans cette analogie nous avons décrit les secteurs d'énergie, la matrice d'énergie et le gradient d'énergie.

L'objectif de notre travail est de proposer une nouvelle manière de routage basé sur la direction du gradient d'énergie qui est utilisé par Sobel. Pour cela nous avons conclu ce chapitre par une étude mathématique afin de montrer que la direction du gradient d'énergie se pointe vers le nœud voisin ayant la plus grande capacité énergétique.

## 3.2 Techniques de traitement d'image

### 3.2.1 Représentation d'une image en niveaux de gris

On peut voir l'image numérique comme un tableau ou une carte de pixels carrés rangés dans des colonnes et des lignes. Dans cette représentation, chaque pixel peut être identifié par des coordonnées X et Y et par une valeur liée à sa luminosité.

Une image en niveaux de gris est ce que les gens appellent normalement une image en noir et blanc, mais le nom signifie qu'une telle image comprendra également de nombreuses nuances de gris. Ainsi dans une image de 8-bit représentée en niveaux de gris, chaque élément de l'image a une intensité variante de 0 à 255 (voir figure 8).

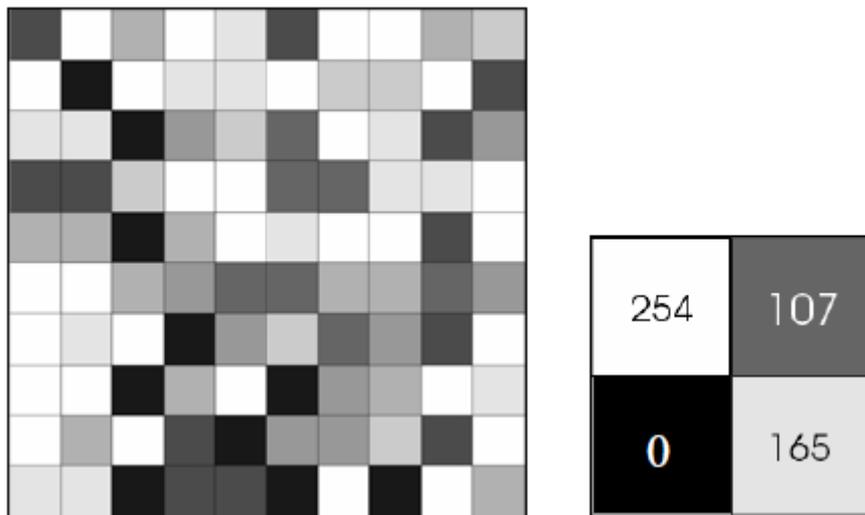


Figure 8 Image en niveau de gris

Le seuillage simple est une des techniques de traitement d'image permettant de mettre en évidence certaines zones de l'image. Cette opération vient du mot "seuil" qui consiste à donner la valeur zéro à tous les pixels ayant un niveau de gris inférieur à une certaine valeur (seuil) et la valeur maximale aux pixels ayant une valeur supérieure. Ainsi on obtiendra une image binaire contenant des pixels noirs et blancs.

Cette opération nous sera utile pour représenter le réseau de capteurs sans fil comme une carte de pixels (image) en 256 niveaux de gris [66].

### 3.2.2 Histogramme

Un histogramme est un graphique statistique permettant de représenter la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse. Par convention un histogramme représente le niveau d'intensité en abscisse en allant du plus foncé (à gauche) au plus clair (à droite). Ainsi, l'histogramme d'une image en 256 niveaux de gris sera représenté par un graphique possédant 256 valeurs en abscisses et le nombre de pixels de l'image en ordonnées [87].

Pour les images en couleur plusieurs histogrammes sont nécessaires. Par exemple pour une image codée en RGB :

- un histogramme représentant la distribution de la luminance,

- trois histogrammes représentant la distribution des valeurs respectives des composantes rouges, bleues et vertes.

### 3.2.3 Détection de contours

La détection de contour est une étape préliminaire à de nombreuses applications d'analyse d'images. Les contours constituent en effet des indices riches, au même titre que les points d'intérêt, pour toute interprétation ultérieure de l'image. Les contours dans une image proviennent des :

- discontinuités de la fonction de refléter (texture, ombre),
- discontinuités de profondeur (bords de l'objet),

Ils sont caractérisés par des discontinuités de la fonction d'intensité dans les images. Le principe de la détection de contours repose donc sur l'étude des dérivées de la fonction d'intensité dans l'image. La difficulté réside dans la présence de bruit dans les images [88, 89].

### 3.2.4 Le gradient d'une image

Le gradient est un vecteur indiquant la façon dont une grandeur physique varie dans l'espace comme le montre l'exemple dans la figure 9 d'un gradient d'une courbe d'isosurface. Dans une image, il est représenté par le vecteur  $\nabla I(x, y)$ :

$$\nabla I(x, y) = \left( \frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)^t$$

où  $I(x, y)$  est la fonction d'intensité.

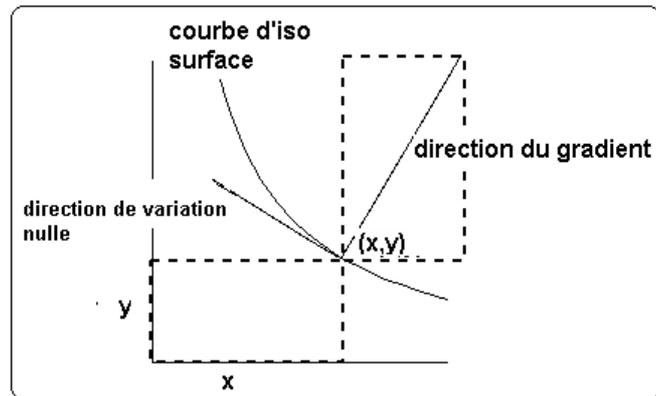


Figure 9 Un gradient d'une courbe d'isosurface

Alors, dans une image le gradient est défini par une direction  $\Phi$  un module  $m$  :

$$m = \sqrt{\left(\frac{\partial I(x, y)}{\partial x}\right)^2 + \left(\frac{\partial I(x, y)}{\partial y}\right)^2}$$

$$\phi = \arctan\left(\frac{\partial I(x, y)}{\partial y} / \frac{\partial I(x, y)}{\partial x}\right)$$

- La direction du gradient maximise la dérivée directionnelle.
- La dérivée de  $I(x, y)$  dans une direction donnée  $d$  s'écrit :

$$\nabla I(x, y) \cdot d$$

### 3.2.5 Les filtres en traitement d'image

Le filtrage consiste à appliquer une transformation (appelée filtre) à tout ou partie d'une image numérique en appliquant un opérateur.

Un filtre est une transformation mathématique permettant, pour chaque pixel de la zone à laquelle il s'applique, de modifier sa valeur en fonction des valeurs des pixels avoisinants, affectées de coefficients.

Filtrer une image c'est lui appliquer une transformation mathématique qui modifie les valeurs de gris de tout ou partie des pixels. Si cette fonction est linéaire au sens algébrique du terme, on parle de filtrage linéaire. Si la fonction prend en compte les valeurs de gris au voisinage de chaque pixel transformé, on parle de fonction de convolution.

Contrairement aux manipulations d'histogrammes, qui sont des opérations ponctuelles sur tous les pixels de l'image, les produits de convolution mettent en jeu l'environnement (voisinage) de chaque pixel.

Un filtre de convolution est défini par:

- Une fenêtre carrée de dimension impaire (3 ou 5) qui est déplacée sur l'image
- Une matrice de coefficients de même dimension (3 ou 5) qui est représentée ci-dessous

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix}$$

Le filtre est représenté par un tableau (matrice), caractérisé par ses dimensions et ses coefficients, dont le centre correspond au pixel concerné. Les coefficients du tableau déterminent les propriétés du filtre..

### 3.2.5.1 Les filtres de convolution

Beaucoup de traitements d'images sont basés sur les produits de convolutions. Un produit de convolution, est un opérateur mathématique qu'on utilise pour multiplier des matrices entre elles. En général, la multiplication se fait entre deux matrices :

1. La matrice image, très grande (par exemple 512 x 512, ce qui représente 262144 pixels)
2. Une matrice plus petite qu'on appelle le noyau parce que c'est le "cœur" de tous les changements qui vont affecter l'image. Le noyau va donc agir sur chacun des pixels, c'est à dire sur chacun des éléments de la matrice "image" (voir figure 10).

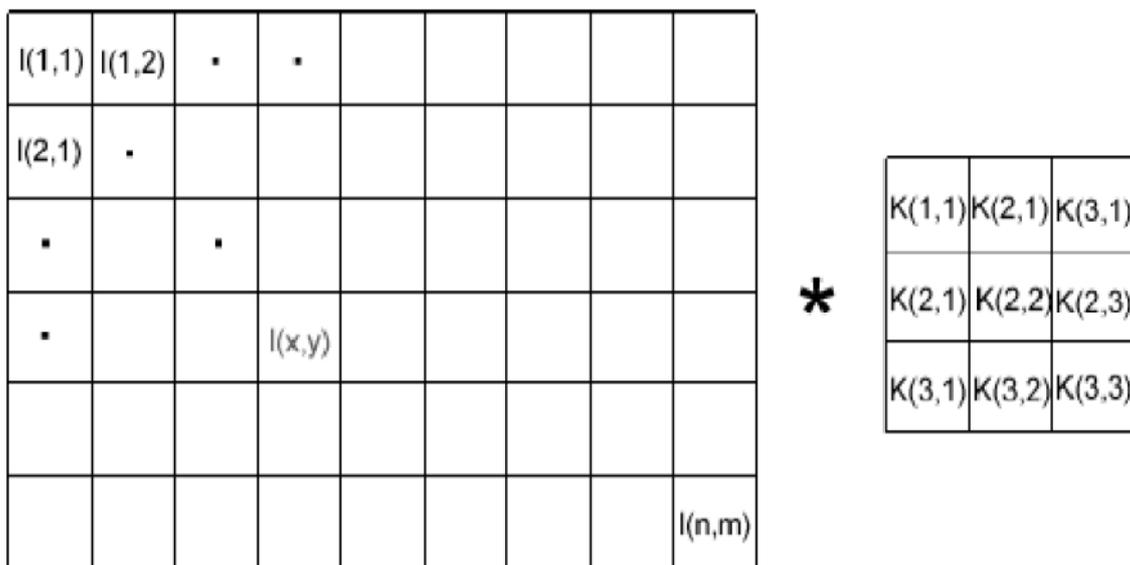


Figure 10 La matrice d'image et le noyau

Appliquer un filtre de convolution consiste à multiplier chacun des pixels de la matrice [i] par le noyau [k]. Pour calculer la valeur d'un pixel I(x, y) de la matrice image, on multiplie sa valeur par celle du pixel central du noyau K(2,2) et on additionne ensuite la valeur des produits des pixels adjacents. Il reste ensuite à diviser le résultat par le nombre d'éléments du noyau, cette dernière opération n'appartient pas au produit de convolution proprement dit, mais elle est nécessaire pour maintenir la dynamique de l'image (différence entre le niveau du pixel le plus élevé et le plus faible) ainsi que sa linéarité [90, 91].

Ainsi, le produit de la matrice image est généralement très grand car il représente l'image initiale (tableau de pixels) par le filtre ce qui donne une matrice correspondant à l'image traitée.

### 3.2.5.2 Exemples des filtres en traitement d'image

#### 3.2.5.2.1 Exemple de filtre passe-bas

La figure 11 montre un exemple d'un filtre de type passe-bas 3x3 :

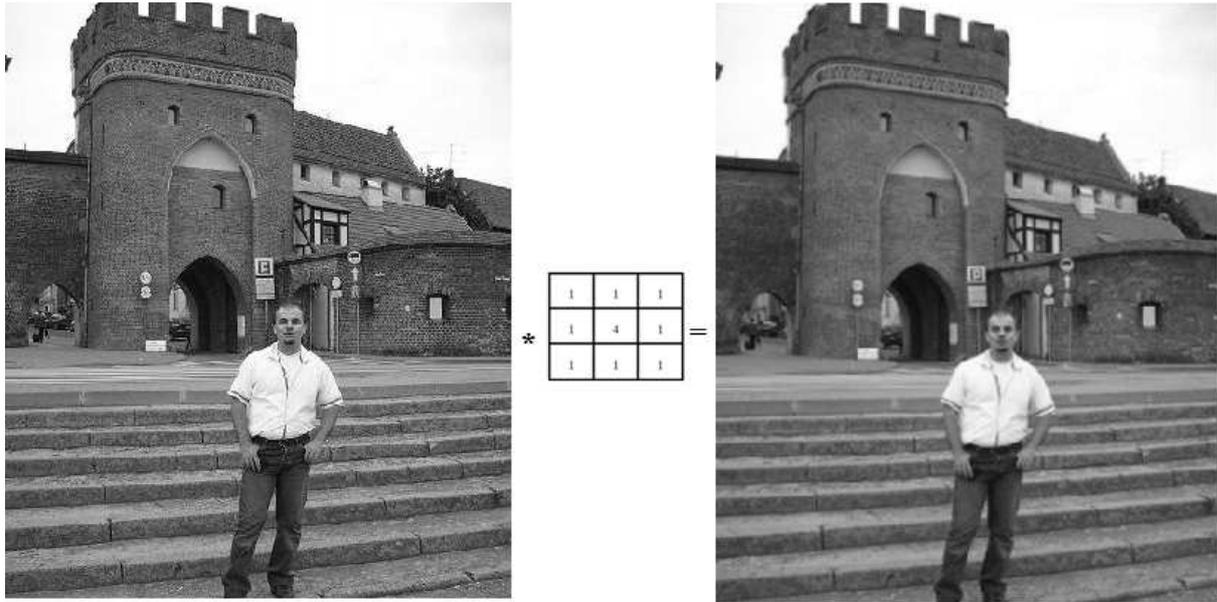


Figure 11 Un exemple de filtre passe-bas

Ce filtre est utilisé pour réduire le bruit granuleux sans perdre les détails dans l'image. Dans l'exemple ci-dessus, en comparant l'image de gauche (image source) et l'image de droite (image filtrée) on remarque le bruit a disparu.

### 3.2.5.2.2 Exemple de filtre moyen

La figure 12 montre exemple d'un filtre moyen qui est simple et facile à mettre en œuvre pour le lissage des images. Ce filtre représente la forme et la taille du voisinage et il facilite le calcul de la moyenne en remplaçant chaque pixel par la valeur moyenne des pixels adjacents et le pixel central. Dans le cas de l'exemple, le nœud central contribue pour la moitié du poids par rapport aux régions voisines.



Figure 12 Un exemple de filtre moyen

Dans l'exemple ci-dessus d'un filtre moyen, on a obtenu une image d'apparence plus lisse que l'image originale sans perdre les détails dans l'image.

Le filtre moyen est le type le plus simple de filtre passe-bas car dans ce filtre tous les coefficients ont des valeurs identiques [101]. Ses caractéristiques sont définies par la largeur, la hauteur et la forme du noyau. Si les changements dans l'image se produisent principalement dans une seule direction, le lissage peut être réglé en changeant la forme du filtre. Lorsque la taille du noyau augmente, le lissage augmente. Le choix de la taille du noyau et la forme est un compromis entre la réduction de bruit et le faible effet flou [92, 93].

### 3.2.5.2.3 Exemple de filtre gradient

La figure 13 suivante montre exemple d'un filtre gradient qui permet de visualiser les variations d'un phénomène, ainsi un dégradé de couleur peut s'appeler un gradient de couleur. Le gradient est la dérivée partielle le long d'une direction particulière, en général l'un des deux axes cartésiens X ou Y de l'image. Il est basé sur un produit de convolution dont l'effet visible est un effet de relief qui permet de visualiser de faibles variations de luminosité [94].

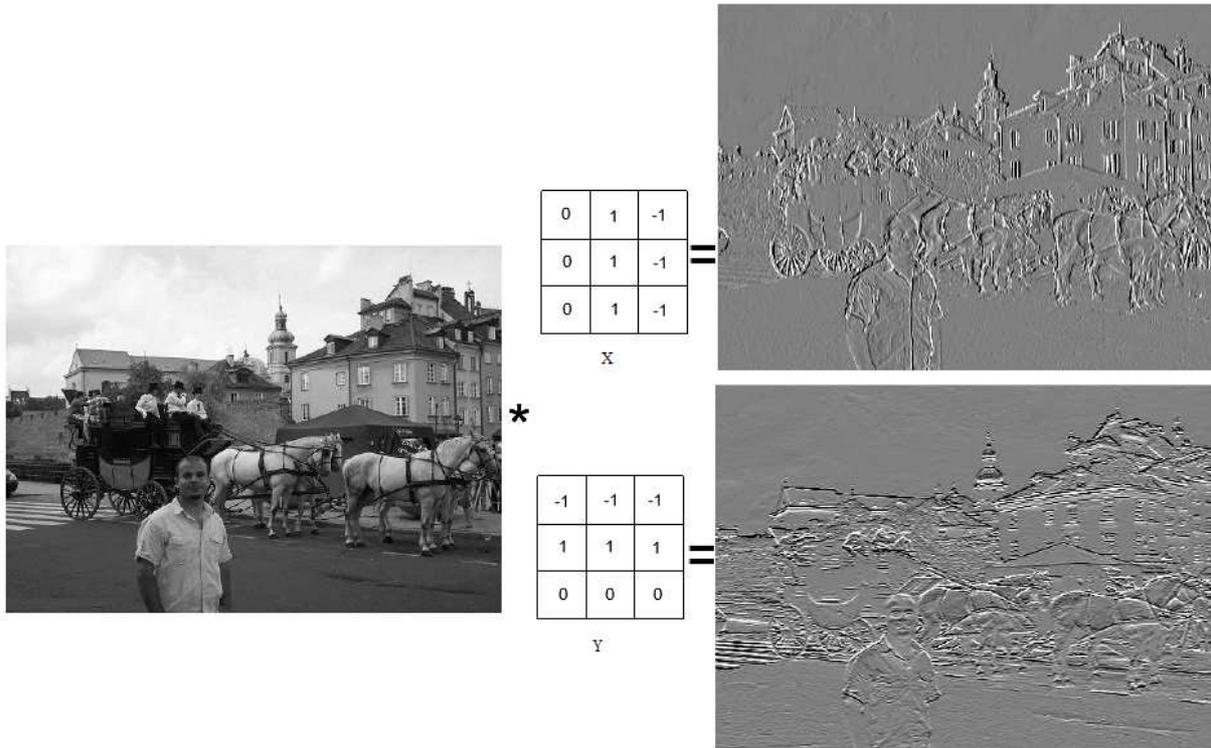


Figure 13 Un exemple de filtre gradient

Dans cet exemple on a appliqué deux filtres gradients directionnels qui montrent dans les deux images de gauche la variation de luminosité dans les directions de l'axe X et l'axe Y.

#### 3.2.5.2.4 Exemple de filtre Sobel

Cet exemple présente un filtre de Sobel qui utilise généralement deux noyaux 3x3, l'un pour l'axe horizontal (X) et l'autre pour l'axe vertical (Y). Chacun des noyaux est en fait un filtre gradient. Ils sont tous les deux combinés pour créer l'image finale [88].

Les deux noyaux utilisés en Sobel :

$$K_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Si nous appliquons ces deux noyaux sur notre image, nous obtenons les résultats montrés dans la figure 14:

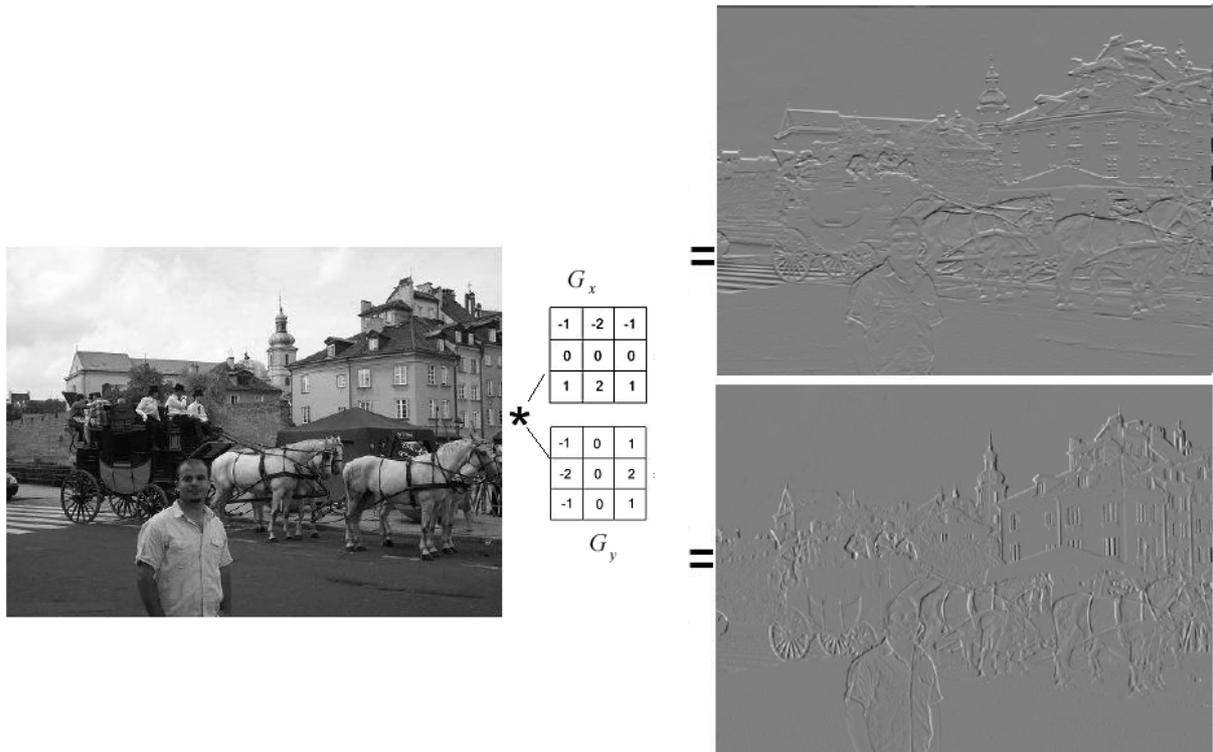


Figure 14 Un exemple de filtre Sobel

Dans les deux images de gauche, le filtre de Sobel a détecté les contours dans les deux directions horizontale et vertical et a indiqué la direction de la plus forte variation du clair au sombre.

## 3.3 Modélisation

### 3.3.1 Modèle du réseau

Dans un réseau de capteurs sans fil, chaque nœud peut surveiller son environnement et réagir si nécessaire en envoyant les informations recueillies à un ou plusieurs points de collecte à l'aide d'une connexion sans fil. Chaque nœud a une capacité énergétique qui doit être utilisée de manière efficace pour maximiser la durée de vie du réseau car la défaillance d'un capteur peut couper les communications dans l'ensemble du réseau. Une fois que le nœud a épuisé son énergie, il est considéré comme défaillant.

Chaque nœud est capable de recevoir et de transmettre des paquets à un nœud particulier choisi par l'administrateur (le cluster-head) qui centralisera les informations. Ces paquets sont transmis sous forme de messages UDP. Ainsi un capteur peut être dans deux états à un moment donné: en transmission ou en réception. La limitation des communications radio implique que les communications d'un canal sont limitées en distance et que les capteurs agissent en mode semi-duplex. Les capteurs sont alimentés par une batterie. Ces ressources sont utilisées pour l'acquisition de données, le calcul et la communication.

Nous considérons un réseau de capteurs tel que:

- Le réseau communique par paquets.
- Les capteurs sont en mode semi-duplex.
- Seuls les capteurs situés à une distance donnée reçoivent les paquets.
- Le temps est discrétisé
- Les paquets sont envoyés à différents intervalles de temps.
- Le capteur est limité en autonomie et en capacité de calcul.

### 3.3.2 Modèle énergétique

La consommation d'énergie est très importante car les capteurs sont généralement déployés dans des zones inaccessibles [77]. Il est difficile, voire impossible, de remplacer les piles si toute l'énergie a été consommée. Ainsi, le niveau d'énergie du capteur a une grande influence sur la durée de vie du réseau. L'énergie consommée par un capteur est due à la capture, au traitement (la commutation de l'énergie et les fuites d'énergie) et à la

communication (l'énergie de transmission et l'énergie de réception) [68, 69]. Ainsi, l'énergie consommée est fonction du temps de transmission, de réception, d'inactivité et du mode veille :

$$E(t + \Delta t) = E(t) - E(\Delta t)$$

$$E(t + \Delta t) = E(t) - E_{\text{tran}} * \Delta t_{\text{tran}} - E_{\text{recv}} * \Delta t_{\text{recv}} - E_{\text{idle}} * \Delta t_{\text{idle}} - E_{\text{sleep}} * \Delta t_{\text{sleep}}$$

$$\Delta t = \Delta t_{\text{tran}} + \Delta t_{\text{recv}} + \Delta t_{\text{idle}} + \Delta t_{\text{sleep}}$$

$E_{\text{tran}}$ ,  $E_{\text{recv}}$  désignent, la consommation d'énergie en mode de transmission et de réception.

$E_{\text{idle}}$ , se réfère à l'énergie dépensée pendant l'inactivité.

$E_{\text{sleep}}$ , représente la dépense d'énergie en mode veille.

$\Delta t_{\text{tran}}$ ,  $\Delta t_{\text{recv}}$ , désignent le temps de la transmission et de la réception d'énergie.

$\Delta t_{\text{idle}}$ , représente le temps d'énergie dépensée pendant l'inactivité.

$\Delta t_{\text{sleep}}$ , désigne le temps d'énergie dépensée en mode veille.

### 3.3.3 Analogie entre les RCSF et le niveau de gris d'une image

Le principe de notre travail est de réaliser un lien entre les réseaux de capteurs sans fil et l'image afin d'utiliser les techniques de traitement d'image pour obtenir des informations concernant la répartition de l'énergie dans le réseau et créer ensuite un protocole de routage dans les RCSF.

Pour atteindre cet objectif, nous représentons la capacité des capteurs du réseau par une image (une grille échelonnée). Le pixel avec des coordonnées (X,Y) représente le capteur situé à (X, Y) avec une valeur de sa capacité énergétique  $C_{x,y}$ . On lie cette capacité à sa luminosité  $0 \leq C_{x,y} \leq 255$ . Tous les capteurs avec une batterie pleine sont représentés par un pixel blanc et les capteurs avec une batterie vide seront représentés par des pixels noirs. De cette façon, on peut regarder le réseau de capteurs comme une image avec des régions noires, blanches et grises.

Après cette représentation et afin de conserver l'énergie des capteurs et trouver le meilleur routage, il suffit de trouver les chemins de la source vers la destination passant par les régions claires de l'image. De plus, en évitant les régions sombres, on évite les trous qui pourraient se retrouver souvent dans les réseaux de capteurs sans fil.

Pour atteindre cet objectif nous avons coupé l'image en sous-ensembles autour d'un nœud central pour déterminer la répartition de l'énergie. Nous avons appelé ces sous-ensembles les secteurs d'énergie.

### 3.3.3.1 Les secteurs d'énergie

Chaque pixel  $P_{x,y}$  dans une image a huit voisins directs comme le montre la figure 15. Le réseau de capteurs est défini comme une carte de pixels et on peut identifier chaque pixel par ses coordonnées X et Y et lui affecter une valeur liée à sa luminosité.

$P_{x-1,y-1}$ b	$P_{x,y-1}$ b	$P_{x+1,y-1}$ b
$P_{x-1,y}$ b	$P_{x,y}$ a	$P_{x+1,y}$ b
$P_{x-1,y+1}$ b	$P_{x,y+1}$ b	$P_{x+1,y+1}$ b

Figure 15 Les huit voisins d'un pixel  $P_{x,y}$

De la même façon, nous pouvons couper le voisinage d'un capteur ou d'un nœud dans un rayon R en secteurs afin de comparer l'énergie de ce nœud par rapport à ses voisins [55, 56] comme le montre la figure 16.

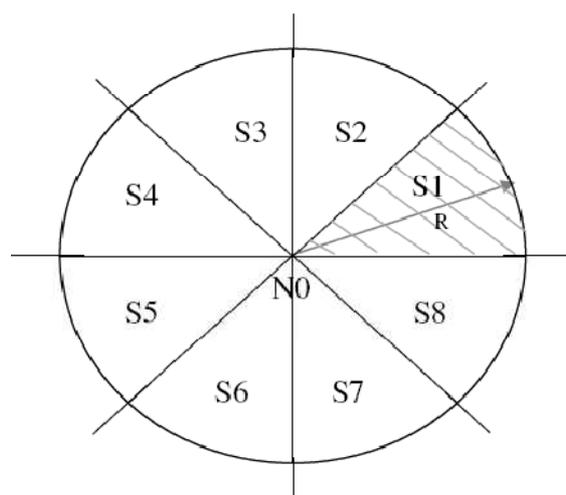


Figure 16 Les secteurs d'énergie

Chacun de ces huit secteurs aura une énergie moyenne qui sera utilisée pour créer la matrice énergétique M.

### 3.3.3.2 La matrice M

On peut utiliser dans les images numériques un tableau de X colonnes et Y lignes. Ce tableau peut réserver une place pour ranger les valeurs énergétiques des nœuds représentées par les pixels de l'image.

M est une matrice  $3 \times 3$  qui représente les valeurs énergétiques des nœuds autour d'un nœud central :

$$M_{x,y} = \begin{pmatrix} s_3 & s_2 & s_1 \\ s_4 & C_{x,y} & s_8 \\ s_5 & s_6 & s_7 \end{pmatrix}$$

où  $C_{x,y}$  est la capacité énergétique du nœud ayant les coordonnées (x,y) et  $S_i$  est la valeur de l'énergie du secteur  $s_i$ .

### 3.3.3.3 Le produit de convolution

Dans notre travail, le produit de convolution est obtenu par la multiplication de deux matrices comportant :

- 1- la matrice  $M_{x,y}$ , qui représente les valeurs de l'énergie des secteurs autour d'un nœud central (matrice énergétique).
- 2- la matrice appelée le noyau ou le filtre, noté K. Ce noyau s'applique successivement à chaque pixel que nous appelons le «pixel initial». Il multiplie la valeur de ce pixel et les valeurs des huit secteurs qui l'entourent par la valeur correspondante du noyau. Puis il ajoute les résultats et le pixel initial est fixé à cette valeur finale [67].

Le produit de convolution de deux matrices K et M d'une taille de  $(n+1) \times (p+1)$  est défini par la formule :

$$K * M = \sum_{i=0}^n \sum_{j=0}^p K_{i,j} \times M_{n-i,p-j}$$

Nos noyaux sont composés d'une matrice carrée K de 3x3 éléments. Pour calculer la valeur d'un pixel (x, y) de la matrice  $M_{x, y}$ , on multiplie sa valeur par le pixel central K(2,2) (voir figure 17) et en additionnant la valeur du produit des pixels adjacents.

K(1,1)	K(1,2)	K(1,3)
K(2,1)	K(2,2)	K(2,3)
K(3,1)	K(3,2)	K(3,3)

Figure 17 La matrice noyau

Dans notre travail nous avons utilisé le filtre Sobel et le filtre moyen.

### 3.3.3.3.1 Le filtre moyen

Pour obtenir les paramètres du routage basé sur l'énergie dans un réseau de capteurs sans fil, nous avons utilisé le filtre de convolution donné par le noyau suivant:

$$K = \begin{pmatrix} 1/12 & 1/12 & 1/12 \\ 1/12 & 4/12 & 1/12 \\ 1/12 & 1/12 & 1/12 \end{pmatrix}$$

Rappelons que le produit de convolution de deux matrices K et M de  $(n+1) \times (p+1)$  est défini par la formule :

$$K * M = \sum_{i=0}^n \sum_{j=0}^p K_{i,j} \times M_{n-i,p-j}$$

où M représente la matrice d'énergie et K le noyau.

Ce filtre moyen fonctionne comme un passe-bas. Il est simple et facile à mettre en œuvre pour le lissage des images. En utilisant ce filtre, le produit de convolution  $K * M$  nous donnera la moyenne de l'énergie autour du nœud central. Autrement dit, il remplace chaque

pixel par la valeur moyenne des pixels adjacents et du pixel central. Dans cette configuration où il y a huit secteurs d'énergie comme le montre la figure 16 dans le paragraphe 3.3.3.1, le nœud central contribue pour la moitié du poids par rapport aux régions voisines. Par conséquent, un nœud d'une grande capacité énergétique sera choisi pour transmettre les paquets.

Ces deux noyaux (Sobel et moyen) seront utilisés dans nos algorithmes que nous présentons dans le prochain chapitre.

### 3.3.3.2 Le filtre de Sobel

Le filtre de Sobel est un opérateur utilisé en traitement d'image pour calculer le gradient et détecter les contours à partir de l'intensité des pixels dans l'image. Par la direction du gradient, on peut connaître la répartition de la densité et la luminosité des pixels. Par conséquent, on arrive à détecter les zones sombres et les zones claires dans cette image.

L'opérateur de Sobel est basé sur le produit de convolution. Il aide, par l'idée de gradient, à trouver les variations à haute luminosité dans l'image en employant une fonction de deux variables pour l'axe horizontal (X) et l'axe vertical (Y) [70, 71].

En remplaçant une image par un réseau de capteurs, nous allons utiliser l'opérateur de Sobel pour détecter la variation d'énergie dans ce réseau à l'aide du gradient d'énergie.

L'algorithme de Sobel utilise deux matrices  $3 \times 3$  :

$$K_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \quad K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

Afin de calculer le gradient d'énergie, nous utilisons la définition suivante :

**Définition : gradient d'énergie**

- Soit  $E$  l'image en niveaux de gris qui représente la capacité énergétique d'un réseau.
- Soit  $M_{i,j}$  les sous-matrices  $3 \times 3$  de  $E$  centré sur  $E_{i,j}$ .

alors le gradient d'énergie  $G_{i,j} = (G_x, G_y)$  dans le capteur  $i, j$  est donné par le produit de convolution suivant :

$$G_x = M_{i,j} * K_x$$

$$G_y = M_{i,j} * K_y$$

Dans l'algorithme de Sobel  $G_{i,j}$  représente le gradient de l'intensité de gris au pixel  $(i, j)$ . Cette intensité est liée à la capacité de la batterie du capteur  $(i, j)$  par définition.

**Remarque:** La norme du gradient est donnée par:

$$\|G\| = \sqrt{G_x^2 + G_y^2}$$

et la direction du gradient est donnée par :

$$\theta = \arctan \left( \frac{G_y}{G_x} \right)$$

Les produits de la multiplication de matrice  $M_i$  par les matrices  $K_x, K_y$  sur les deux axes  $x$  et  $y$  sont:

➤ sur l'axe  $x$  :

$$G_x = M_i * K_x$$

$$= \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * \begin{pmatrix} s_3 & s_2 & s_1 \\ s_4 & C_i & s_8 \\ s_5 & s_6 & s_7 \end{pmatrix}$$

$$= S_7 + 2S_8 + S_1 - S_3 - 2S_4 - S_5$$

Dans ce produit, chaque secteur  $S_i$  a un certain coefficient  $(-2, -1, 0, 1, 2)$ . Le produit n'est autre que le barycentre de tous les secteurs  $S_i$  autour d'un nœud central comme le montre la figure 18:

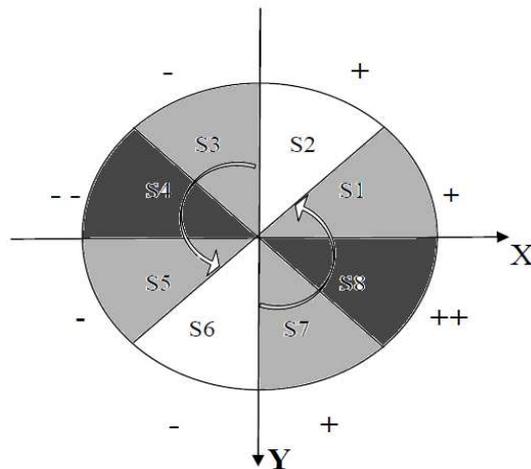


Figure 18 La direction du gradient sur l'axe x

On peut remarquer que la direction du gradient sur l'axe de x est influencée par six valeurs dont trois valeurs positives ( $S_7$ ,  $2S_8$ ,  $S_1$ ) et trois négatives ( $-S_3$ ,  $-2S_4$ ,  $-S_5$ ). Le secteur de l'énergie au centre de chaque côté ( $S_8$  et  $S_4$ ) est le double de l'énergie de ses régions voisines.

➤ sur l'axe y (voir figure 19):

$$\begin{aligned}
 G_y &= M_i * K_y \\
 &= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \begin{pmatrix} S_3 & S_2 & S_1 \\ S_4 & S_0 & S_8 \\ S_5 & S_6 & S_7 \end{pmatrix} \\
 &= S_5 + 2S_6 + S_7 - S_1 - 2S_2 - S_3
 \end{aligned}$$

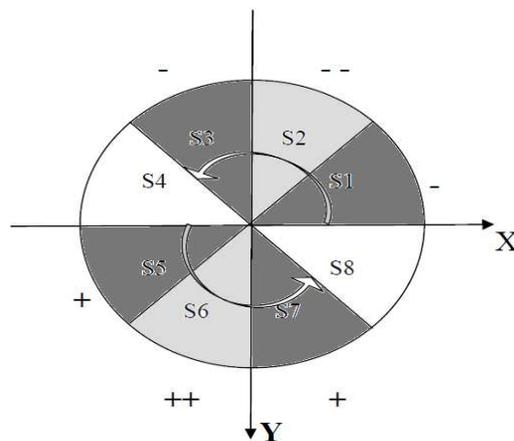


Figure 19 La direction du gradient sur l'axe y

On peut remarquer que la direction du gradient sur l'axe de y est influencée par six valeurs dont trois valeurs positives ( $S_5, 2S_6, S_7$ ) et trois négatives ( $-S_1, -2S_2, -S_3$ ). Le secteur de l'énergie au centre de chaque côté ( $S_6$  et  $S_2$ ) est le double de l'énergie de ses régions voisines.

**Exemple :** soit  $M$  la matrice qui représente les valeurs énergétiques autour du nœud central ayant la valeur énergétique de 4 :

$$M = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \\ 2 & 0 & 1 \end{pmatrix}$$

alors :  $K * M = (2k_{1,3}) + (k_{3,2} + 4k_{3,3}) + (2k_{3,1} + 5k_{2,1} + k_{1,1})$

On peut remarquer que la direction du gradient, qui suit la répartition de l'énergie autour du nœud central, sera influencée principalement par la valeur  $5k_{2,1}$ . Cela dépend bien sûr du noyau utilisé.

Si ce noyau est le filtre de Sobel alors la direction de la route est donnée par le gradient d'énergie  $G_{i,j} = (G_x, G_y)$

$$\begin{cases} G_x = M * K_x \\ G_y = M * K_y \end{cases}$$

- Sur l'axe x :

$$G_x = M_i * K_x = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \\ 2 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

$$G_x = S_7 + 2S_8 + S_1 - S_3 - 2S_4 - S_5$$

$$G_x = 1 + (2 \times 5) + 2 - 0 - (2 \times 0) - 2$$

$$G_x = 11$$

- Sur l'axe y:

$$G_y = M_i * K_y = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 4 & 5 \\ 2 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$

$$G_y = S_5 + 2S_6 + S_7 - S_1 - 2S_2 - S_3$$

$$G_y = 2 + (2 \times 0) + 1 - 2 - (2 \times 1) - 0$$

$$G_y = -1$$

alors la direction du gradient d'énergie est  $G(11, -1)$  comme le montre la figure 20 :

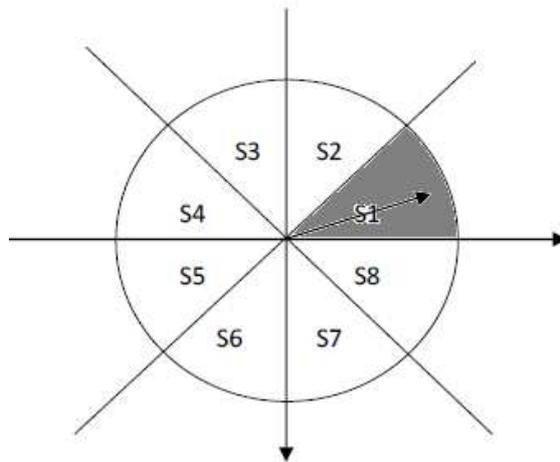


Figure 20 La direction de routage

### 3.3.4 Étude de gradient d'énergie dans les RCSF

Dans le paragraphe 3.3.3.3.2 nous avons proposé un nouveau mécanisme de routage défini par la direction du gradient d'énergie qui est obtenu avec l'algorithme de Sobel.

Nous donnons par la suite une étude mathématique pour vérifier que la direction indiquée par le gradient d'énergie est bien la direction indiquant le capteur qui possède la plus grande capacité énergétique. Notre algorithme de routage choisira la direction définie entre le nœud courant et son voisin qui se rapproche le plus de vecteur gradient d'énergie. Nous cherchons à minimiser l'angle défini par la direction vers le voisin et la direction du gradient d'énergie. Cet angle est considéré comme une distance entre ces deux vecteurs.

L'étude qui va suivre aura pour but de montrer que le fait de minimiser cet angle permet de choisir comme saut suivant (next hop) le voisin qui a le plus d'énergie.

Dans un premier temps nous allons étudier le positionnement du gradient d'énergie par rapport à deux voisins à travers le lemme suivant :

#### Lemme 1

- Soit  $S_i$  les huit secteurs qui représentent le voisinage d'un nœud central  $N_0$  ( $i=1,2,\dots, 8$ ) et  $C_i$  la capacité énergétique de  $S_i$ .
- Soit  $G(x,y)$  le gradient d'énergie de  $N_0$
- Supposons que  $C_1 \neq 0$  et que  $S_1$  a un seul voisin  $N_j$  ( $j \neq 1$ ) avec  $C_j$  comme énergie.

On a :

- 1) pour  $N_2$  : le gradient  $G$  appartient à  $S_2$
- 2) pour  $N_3$  : si  $C_1 > C_3$  alors le gradient  $G$  appartient à  $S_2$   
 - si  $C_1 < C_3$  alors le gradient  $G$  appartient à  $S_3$
- 3) pour  $N_4$  : si  $C_1 > C_4$  (avec  $C_4 = \varepsilon C_1$  et  $0 < \varepsilon < 1$ ) alors :  
 - le gradient  $G$  appartient à  $S_2$  si  $\varepsilon < 1/2$   
 - le gradient  $G$  appartient à  $S_3$  si  $\varepsilon > 1/2$   
 - le gradient  $G$  appartient à la ligne droite  $x=0$  qui sépare les deux secteurs  $S_2$  et  $S_3$  si  $\varepsilon = 1/2$   
 - si  $C_1 < C_4$  alors le gradient  $G$  appartient à  $S_4$
- 4) pour  $N_5$  : si  $C_1 > C_5$  alors  $G$  appartient à la ligne droite qui sépare  $S_1$  et  $S_2$   
 - si  $C_1 < C_5$  alors  $G$  appartient à la ligne droite qui sépare  $S_5$  et  $S_6$
- 5) pour  $N_6$  : si  $C_1 > C_6$  (avec  $C_6 = \varepsilon C_1$  et  $0 < \varepsilon < 1$ ) alors :  
 - le gradient  $G$  appartient à  $S_1$  si  $\varepsilon < 1/2$   
 - le gradient  $G$  appartient à  $S_8$  si  $\varepsilon > 1/2$   
 - le gradient  $G$  appartient à la ligne droite  $y=0$  qui sépare les deux secteurs  $S_1$  et  $S_8$  si  $\varepsilon = 1/2$   
 - si  $C_1 < C_6$  alors le gradient  $G$  appartient à  $S_7$
- 6) pour  $N_7$  : si  $C_1 > C_7$  alors le gradient  $G$  appartient à  $S_1$   
 - si  $C_1 < C_7$  alors le gradient  $G$  appartient à  $S_8$
- 7) Pour  $N_8$  : le gradient  $G$  appartient à  $S_1$

### Démonstration du lemme 1

Comme le montre la figure 21 suivante :

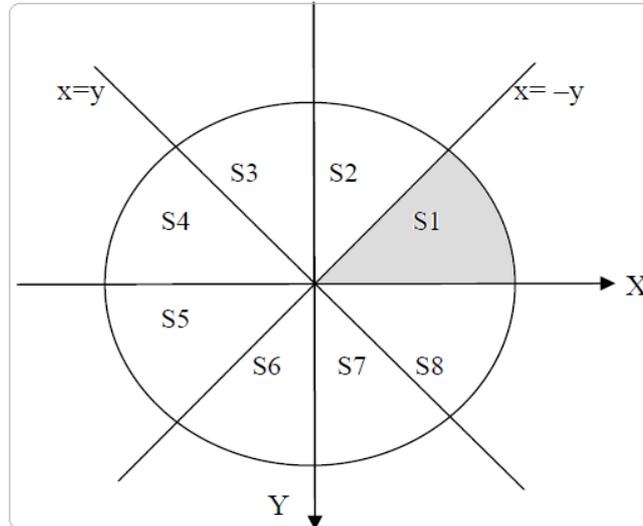


Figure 21 Les secteurs d'énergie

Le secteur  $S_1$  est défini par :

$$\begin{cases} x \geq 0, y \leq 0 \\ x > y \end{cases}$$

Le secteur  $S_2$  est défini par :

$$\begin{cases} x \geq 0, y \leq 0 \\ x < -y \end{cases}$$

Le secteur  $S_3$  est défini par :

$$\begin{cases} x \leq 0, y \leq 0 \\ x > y \end{cases}$$

Le secteur  $S_4$  est défini par :

$$\begin{cases} x \leq 0, y \leq 0 \\ x < y \end{cases}$$

Le secteur  $S_5$  est défini par :

$$\begin{cases} x \leq 0, y \geq 0 \\ x < -y \end{cases}$$

Le secteur  $S_6$  est défini par :

$$\begin{cases} x \leq 0, y \geq 0 \\ x > -y \end{cases}$$

Le secteur  $S_7$  est défini par :

$$\begin{cases} x \geq 0, y \geq 0 \\ x < y \end{cases}$$

Le secteur  $S_8$  est défini par :

$$\begin{cases} x \geq 0, y \geq 0 \\ x > y \end{cases}$$

1) si  $j=2$  :

a) Soit  $C_1 > C_2$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_2 = \varepsilon C_1$

Les coordonnées du gradient sont :

- sur l'axe x :

$$\begin{aligned}
 G_x &= M_i * K_x \\
 &= \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * \begin{pmatrix} C_3 & C_2 & C_1 \\ C_4 & C_0 & C_8 \\ C_5 & C_6 & C_7 \end{pmatrix} \\
 &= C_7 + 2C_8 + C_1 - C_3 - 2C_4 - C_5 \\
 &= C_1
 \end{aligned}$$

- sur l'axe y :

$$\begin{aligned}
 G_y &= M_i * K_y \\
 &= \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \begin{pmatrix} C_3 & C_2 & C_1 \\ C_4 & C_0 & C_8 \\ C_5 & C_6 & C_7 \end{pmatrix} \\
 &= C_5 + 2C_6 + C_7 - C_1 - 2C_2 - C_3 \\
 &= -C_1 - 2C_2 \\
 &= -C_1 - 2 \varepsilon C_1
 \end{aligned}$$

On a  $G(x,y) = G(C_1, -C_1 - 2 \varepsilon C_1)$

$$\rightarrow G(x,y) = G(C_1, -(2 \varepsilon + 1) C_1)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x < -y$  car  $C_1 < (2 \varepsilon + 1) C_1$

Cela prouve que  $G \in S_2$

- b) soit  $C_1 < C_2$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_2$

$$G(x,y) = G(C_1, 2C_2 - C_1)$$

$$G(x,y) = G(\varepsilon C_2, -\varepsilon C_2 - 2C_2)$$

$$G(x,y) = G(\varepsilon C_2, -(2 + \varepsilon) C_2)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x < -y$  car  $\varepsilon C_2 < (2 + \varepsilon) C_2$

cela prouve également que  $G \in S_2$

- 2) si  $j=3$  ceci implique  $G(x,y) = G(C_1 - C_3, -C_1 - C_3)$

- a) soit  $C_1 > C_3$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_3 = \varepsilon C_1$  alors :

$$G(x,y) = G(C_1 - \varepsilon C_1, -C_1 - \varepsilon C_1)$$

$$G(x,y) = G((1 - \varepsilon) C_1, -(1 + \varepsilon) C_1)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x < -y$  car  $(1 - \varepsilon) < (1 + \varepsilon)$

cela prouve que  $G \in S_2$

b) Soit  $C_1 < C_3$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_3$

$$G(x,y) = G(\varepsilon C_3 - C_3, -\varepsilon C_3 - C_3)$$

$$G(x,y) = G((\varepsilon - 1) C_3, -(\varepsilon + 1) C_3)$$

on remarque que  $x < 0$ ,  $y < 0$  et  $x > y$

cela prouve que  $G \in S_3$

3) si  $j=4$  ceci implique  $G(x,y) = G(C_1 - 2C_4, -C_1)$

a) Soit  $C_1 > C_4$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_4 = \varepsilon C_1$  alors :

$$G(x,y) = G(C_1 - 2\varepsilon C_1, -C_1)$$

$$G(x,y) = G((1 - 2\varepsilon) C_1, -C_1)$$

on remarque que  $y < 0$

➤ si  $\varepsilon < 1/2$  alors  $x > 0$  et  $x < -y$  cela prouve que  $G \in S_2$

➤ si  $\varepsilon > 1/2$  alors  $x < 0$  et  $x > y$  cela prouve que  $G \in S_3$

➤ si  $\varepsilon = 1/2$  alors  $x = 0$  cela prouve que  $G$  appartient à coté positif de la ligne droite  $x=0$  qui sépare les deux secteurs  $S_2$  et  $S_3$ .

b) Soit  $C_1 < C_4$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_4$

$$G(x,y) = G(\varepsilon C_4 - 2C_4, -\varepsilon C_4)$$

$$G(x,y) = G((\varepsilon - 2)C_4, -\varepsilon C_4)$$

on remarque que  $x < 0$ ,  $y < 0$  et  $x < y$

cela prouve que  $G \in S_4$

4) si  $j=5$  ceci implique  $G(x,y) = G(C_1 - C_5, C_5 - C_1)$

a) Soit  $C_1 > C_5$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_5 = \varepsilon C_1$  alors :

$$G(x,y) = G(C_1 - \varepsilon C_1, \varepsilon C_1 - C_1)$$

$$G(x,y) = G((1 - \varepsilon) C_1, (\varepsilon - 1)C_1)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x = -y$  car  $1 - \varepsilon = -(\varepsilon - 1)$

cela prouve que  $G$  appartient à la ligne droite qui sépare les deux secteurs  $S_1$  et  $S_2$ .

b) Soit  $C_1 < C_5$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_5$

$$G(x,y) = G(\varepsilon C_5 - C_5, C_5 - \varepsilon C_5)$$

$$G(x,y) = G((\varepsilon - 1) C_5, (1 - \varepsilon) C_5)$$

on remarque que  $x < 0$ ,  $y > 0$  et  $x = -y$  car  $\varepsilon - 1 = -(1 - \varepsilon)$

cela prouve que  $G$  appartient à la ligne droite qui sépare les deux secteurs  $S_5$  et  $S_6$ .

5) si  $j=6$  ceci implique  $G(x,y) = G(C_1, 2C_6 - C_1)$

a) Soit  $C_1 > C_6$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_6 = \varepsilon C_1$  alors :

$$G(x,y) = G(C_1, 2\varepsilon C_1 - C_1)$$

$$G(x,y) = G(C_1, (2\varepsilon - 1) C_1)$$

on remarque que  $x > 0$ ,

➤ si  $\varepsilon < 1/2$  alors  $y < 0$  et  $x > y$  cela prouve que  $G \in S_1$

➤ si  $\varepsilon > 1/2$  alors  $y > 0$  et  $x > y$  cela prouve que  $G \in S_8$

➤ si  $\varepsilon = 1/2$  alors  $y = 0$  cela prouve que  $G$  appartient à coté positif de la ligne droite  $y=0$  qui sépare les deux secteurs  $S_1$  et  $S_8$ .

b) Soit  $C_1 < C_6$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_6$

$$G(x,y) = G(\varepsilon C_6, 2C_6 - \varepsilon C_6)$$

$$G(x,y) = G(\varepsilon C_6, (2 - \varepsilon) C_6)$$

on remarque que  $x > 0$ ,  $y > 0$  et  $x < y$

cela prouve que  $G \in S_7$

6) si  $j=7$  ceci implique  $G(x,y) = G(C_1 + C_7, C_7 - C_1)$

a) Soit  $C_1 > C_7$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_7 = \varepsilon C_1$  alors :

$$G(x,y) = G(C_1 + \varepsilon C_1, \varepsilon C_1 - C_1)$$

$$G(x,y) = G((1 + \varepsilon) C_1, (\varepsilon - 1) C_1)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x > y$

cela prouve que  $G \in S_1$

b) Soit  $C_1 < C_7$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_7$

$$G(x,y) = G(\varepsilon C_7 + C_7, C_7 - \varepsilon C_7)$$

$$G(x,y) = G((\varepsilon + 1) C_7, (1 - \varepsilon) C_7)$$

on remarque que  $x > 0$ ,  $y > 0$  et  $x > y$

cela prouve que  $G \in S_8$

7) si  $j=8$  ceci implique  $G(x,y) = G(C_1, 2C_6 - C_1)$

a) Soit  $C_1 > C_8$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_8 = \varepsilon C_1$  alors :

$$G(x,y) = G(C_1 + 2\varepsilon C_1, -C_1)$$

$$G(x,y) = G((1 + 2\varepsilon) C_1, -C_1)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x > y$

cela prouve que  $G \in S_1$

b) Soit  $C_1 > C_8$  alors  $\exists 0 < \varepsilon < 1$  tel que  $C_1 = \varepsilon C_8$  alors :

$$G(x,y) = G(2C_8 + \varepsilon C_8, -\varepsilon C_8)$$

$$G(x,y) = G((2 + \varepsilon) C_8, -\varepsilon C_8)$$

on remarque que  $x > 0$ ,  $y < 0$  et  $x > y$

cela prouve que  $G \in S_1$

Ce lemme permet dans certains cas de conclure que le gradient d'énergie indique la direction vers le voisin qui a la plus grande capacité énergétique.

**Corollaire du lemme1**

Soit  $N_1, N_i$  deux nœuds tel que  $N_1 \in S_1, N_i \in S_i$  avec  $C_i$  comme capacité énergétique alors :

a) Si  $C_1 > C_i$  (avec  $C_i = \varepsilon C_1$  et  $0 < \varepsilon < 1$ ) alors on a  $(\vec{G}, \vec{N}_1) < (\vec{G}, \vec{N}_i)$  dans les cas suivants :

1)  $i = 4$  et  $\varepsilon < 1/2$

2)  $i = 5$

3)  $i = 6$

4)  $i = 7$

5)  $i = 8$

b) Si  $C_1 < C_i$  (avec  $C_1 = \varepsilon C_i$  et  $0 < \varepsilon < 1$ ) alors on a  $(\vec{G}, \vec{N}_1) > (\vec{G}, \vec{N}_i)$  dans les cas suivants :

6)  $i = 2$

7)  $i = 3$

8)  $i = 4$

9)  $i = 5$

10)  $i = 6$

c) Le gradient est plus proche d'un secteur faible en énergie dans les cas suivants :

1) pour  $i = 2$  et  $C_1 > C_2$

2) pour  $i = 4$  et  $C_1 > C_4$  et  $\varepsilon > 1/2$

3) pour  $i = 8$  et  $C_1 < C_8$

d) Le gradient indique un secteur intermédiaire dans les cas suivants :

1) pour  $i = 3$  et  $C_1 > C_3$

2) pour  $i = 4$  et  $C_1 > C_4$  et  $\varepsilon = 1/2$

3) pour  $i = 7$  et  $C_1 < C_7$

Le corollaire précédent nous montre qu'il existe des cas où la direction du gradient d'énergie n'est pas forcément celle du voisin doté de la plus grande capacité énergétique.

Remarquons que le gradient d'énergie se positionne uniquement en fonction des capacités énergétiques de deux voisins du capteur courant. Le positionnement des voisins n'est pas connu exactement. Effectivement la seule information dont on dispose est que le voisin  $N_i$  appartient au secteur  $S_i$ .

Nous adoptons une approche probabiliste en supposant que les coordonnées du nœud  $N_i$  sont données par des variables aléatoires de la loi uniforme sur les secteurs  $S_i$ .

Le lemme suivant montre que, dans le cas où  $C_1 > C_2$ , la probabilité pour que l'angle  $(\vec{G}, \vec{N}_1)$  soit inférieur à l'angle  $(\vec{G}, \vec{N}_2)$  est majoré par 1/2.

**Lemme2**

Soit  $N_1 \in S_1, N_2 \in S_2$  tel que  $C_1 > C_2$

Soit  $\alpha_1 = (\vec{X}, \vec{N}_1), \alpha_2 = (\vec{X}, \vec{N}_2), \beta = (\vec{X}, \vec{G})$  alors :

- $P((\vec{G}, \vec{N}_1) > (\vec{G}, \vec{N}_2)) = 0$  si  $(\vec{G}, \vec{N}_2) < \beta$

- $P((\vec{G}, \vec{N}_1) > (\vec{G}, \vec{N}_2)) = \frac{32}{\pi^2} \left(\beta - \frac{3\pi}{8}\right)^2$  avec  $\beta \in \left[\frac{\pi}{4}, \frac{3\pi}{8}\right]$  si  $(\vec{G}, \vec{N}_2) > \beta$
- $P((\vec{G}, \vec{N}_1) > (\vec{G}, \vec{N}_2)) = 0$  si  $\beta > \frac{3\pi}{8}$

### Démonstration du lemme2

Lorsque  $C_1 > C_2$  alors  $G \in S_2$  (voir lemme1)

Comme le montre la figure 22 on a :

$$(\vec{G}, \vec{N}_1) = \beta - \alpha_1$$

$$(\vec{G}, \vec{N}_2) = \beta - \alpha_2$$

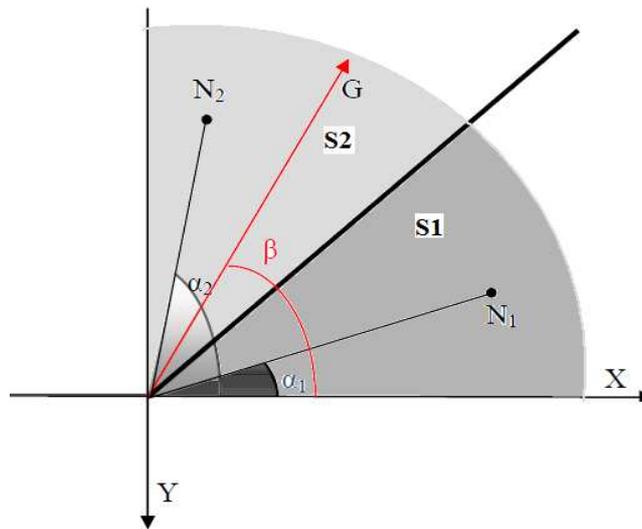


Figure 22 La direction du gradient d'énergie

1) si  $\beta - \alpha_2 > 0$  alors  $N_2$  est toujours plus proche de  $G \Rightarrow P(\beta - \alpha_1 < \beta - \alpha_2) = 0$

2) si  $\beta - \alpha_2 < 0 \Rightarrow$

$$P(\beta - \alpha_1 < \alpha_2 - \beta) = P(2\beta < \alpha_1 + \alpha_2)$$

$\alpha_1$  est une variable aléatoire uniforme :  $\alpha_1 \in \left[0, \frac{\pi}{4}\right]$

$\alpha_2$  est une variable aléatoire uniforme :  $\alpha_2 \in \left[\frac{\pi}{4}, \frac{\pi}{2}\right]$

D'après la loi uniforme on a la densité de probabilité de  $\alpha_1$  et de  $\alpha_2$  :

$$f_{\alpha_1}(x) = \begin{cases} \frac{4}{\pi} & \text{si } x \in \left[0, \frac{\pi}{4}\right] \\ 0 & \text{sinon} \end{cases}$$

$$f_{\alpha_2}(x) = \begin{cases} \frac{4}{\pi} & \text{si } x \in \left[ \frac{\pi}{4}, \frac{\pi}{2} \right] \\ 0 & \text{sinon} \end{cases}$$

Les variables aléatoires  $\alpha_1$  et  $\alpha_2$  sont indépendants

On a alors une nouvelle variable aléatoire  $\alpha_1 + \alpha_2$  :

$$f_{\alpha_1 + \alpha_2}(x) = f_{\alpha_1}(x) * f_{\alpha_2}(x) = \int_{-\infty}^{+\infty} f_{\alpha_1}(y) f_{\alpha_2}(x - y) dy$$

Comme  $f_{\alpha_1}(y) = 0$  si  $y \notin \left[ 0, \frac{\pi}{4} \right]$  on a :

$$f_{\alpha_1}(x) * f_{\alpha_2}(x) = \int_0^{\frac{\pi}{4}} f_{\alpha_1}(y) f_{\alpha_2}(x - y) dy = \frac{4}{\pi} \int_0^{\frac{\pi}{4}} f_{\alpha_2}(x - y) dy$$

On fait le changement de variable :

$$u = x - y$$

On a alors :

$$f_{\alpha_1 + \alpha_2}(x) = \frac{4}{\pi} \int_x^{x - \frac{\pi}{4}} f_{\alpha_2}(u) (-du) = \frac{4}{\pi} \int_{x - \frac{\pi}{4}}^x f_{\alpha_2}(u) du$$

Comme  $f_{\alpha_2}(u) \neq 0$  si  $u \in \left[ \frac{\pi}{4}, \frac{\pi}{2} \right]$

L'intégral  $f_{\alpha_1 + \alpha_2}(x)$  est non nul si  $I = \left[ \frac{\pi}{4}, \frac{\pi}{2} \right] \cap \left[ x - \frac{\pi}{4}, x \right] \neq \emptyset$

On a  $I = \emptyset$  si  $x < \frac{\pi}{4}$

$$I = \emptyset \text{ si } x > \frac{3\pi}{4}$$

$$I = \left[ \frac{\pi}{4}, x \right] \text{ si } x \in \left[ \frac{\pi}{4}, \frac{\pi}{2} \right]$$

$$I = \left[ x - \frac{\pi}{4}, \frac{\pi}{2} \right] \text{ si } x \in \left[ \frac{\pi}{2}, \frac{3\pi}{4} \right]$$

Ceci implique :

$$f_{\alpha_1 + \alpha_2}(x) = \begin{cases} 0 & \text{si } x < \frac{\pi}{4} \text{ ou } x > \frac{3\pi}{4} \\ \frac{4}{\pi} \int_{\frac{\pi}{4}}^x \frac{4}{\pi} du & \text{si } x \in \left[ \frac{\pi}{4}, \frac{\pi}{2} \right] \\ \frac{4}{\pi} \int_{x-\frac{\pi}{4}}^{\frac{\pi}{2}} \frac{4}{\pi} du & \text{si } x \in \left[ \frac{\pi}{2}, \frac{3\pi}{4} \right] \end{cases}$$

$$f_{\alpha_1 + \alpha_2}(x) = \begin{cases} 0 & \text{si } x \in ]-\infty, \frac{\pi}{4}] \cup \left[ \frac{3\pi}{4}, +\infty[ \\ \frac{16}{\pi^2} \left( x - \frac{\pi}{4} \right) & \text{si } x \in \left[ \frac{\pi}{4}, \frac{\pi}{2} \right] \\ \frac{16}{\pi^2} \left( \frac{3\pi}{4} - x \right) & \text{si } x \in \left[ \frac{\pi}{2}, \frac{3\pi}{4} \right] \end{cases}$$

Nous voulons calculer  $P(2\beta < \alpha_1 + \alpha_2) = 1 - P(\alpha_1 + \alpha_2 < 2\beta)$

$$P(\alpha_1 + \alpha_2 < 2\beta) = \int_{-\infty}^{2\beta} f_{\alpha_1 + \alpha_2}(x) dx$$

Rappelons que :

$G(x, y) = G(C_1, -C_1 - 2\varepsilon C_1)$  avec  $C_2 = \varepsilon C_1$  et  $\varepsilon \in [0, 1]$

Si  $\varepsilon = 0$  alors  $G(x, y) = G(C_1, -C_1) \Rightarrow \beta = \frac{\pi}{4}$

Si  $\varepsilon = 1$  alors  $G(x, y) = G(C_1, -3C_1) \Rightarrow \beta = \text{Arctan}(-3) + 2\pi \approx \frac{8\pi}{5}$

$$\Rightarrow \beta \in \left[ \frac{\pi}{4}, \frac{8\pi}{5} \right[ \Rightarrow 2\beta \in \left[ \frac{\pi}{2}, \frac{16\pi}{5} \right[$$

Ce qui nous donne :

$$P(\alpha_1 + \alpha_2 < 2\beta) = \int_{\frac{\pi}{4}}^{2\beta} f_{\alpha_1 + \alpha_2}(x) dx$$

si  $2\beta \in \left[ \frac{\pi}{4}, \frac{\pi}{2} \right]$  alors  $\beta \in \left[ \frac{\pi}{16}, \frac{\pi}{4} \right]$  qui signifie qu'il y a une contradiction car  $\beta > \frac{\pi}{4}$

alors  $\beta \in \left[ \frac{\pi}{2}, \frac{3\pi}{4} \right]$

On a :

$$P(\alpha_1 + \alpha_2 < 2\beta) = \int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \frac{16}{\pi^2} \left( x - \frac{\pi}{4} \right) dx + \int_{\frac{\pi}{2}}^{2\beta} \frac{16}{\pi^2} \left( \frac{3\pi}{4} - x \right) dx$$

$$\begin{aligned}
 P(\alpha_1 + \alpha_2 < 2\beta) &= \frac{16}{\pi^2} \left[ \frac{x^2}{2} - \frac{\pi}{4}x \right] + \frac{16}{\pi^2} \left[ \frac{3\pi}{4}x - \frac{x^2}{2} \right] \\
 &= \frac{16}{\pi^2} \left( \frac{\pi^2}{8} - \frac{\pi^2}{8} - \frac{\pi^2}{32} + \frac{\pi^2}{16} \right) + \frac{16}{\pi^2} \left( \frac{3\pi}{4} * 2\beta - 2\beta^2 - \frac{3\pi^2}{8} + \frac{\pi^2}{8} \right) \\
 &= \frac{16}{\pi^2} \left( \frac{\pi^2}{32} \right) + \frac{16}{\pi^2} \left( \frac{3\pi}{2}\beta - 2\beta^2 - \frac{\pi^2}{4} \right) \\
 &= \frac{1}{2} + \frac{24}{\pi}\beta - \frac{32}{\pi^2}\beta^2 - 4 \\
 &= \frac{24}{\pi}\beta - \frac{32}{\pi^2}\beta^2 - \frac{9}{2}
 \end{aligned}$$

Avec  $\beta \in \left[ \frac{\pi}{4}, \frac{3\pi}{8} \right]$  on a donc :

$$\begin{aligned}
 P(2\beta < \alpha_1 + \alpha_2) &= 1 - P(\alpha_1 + \alpha_2 < 2\beta) \\
 &= \frac{32}{\pi^2}\beta^2 - \frac{24}{\pi}\beta + \frac{9}{2} \\
 &= \frac{32}{\pi^2} \left( \beta - \frac{3\pi}{8} \right)^2
 \end{aligned}$$

### Corollaire du lemme2

- ❖ Si  $\varepsilon \geq \frac{\sqrt{2}}{2}$  alors la probabilité pour que G soit plus proche de  $N_1$  que de  $N_2$  est de 0.
- ❖ Si  $\varepsilon = 0$  alors  $\tan \beta = 1 \Rightarrow \beta = \frac{\pi}{4}$  alors la probabilité pour que G soit plus proche de  $N_1$  que de  $N_2$  est de  $\frac{1}{2}$  comme le montre la figure 23 :

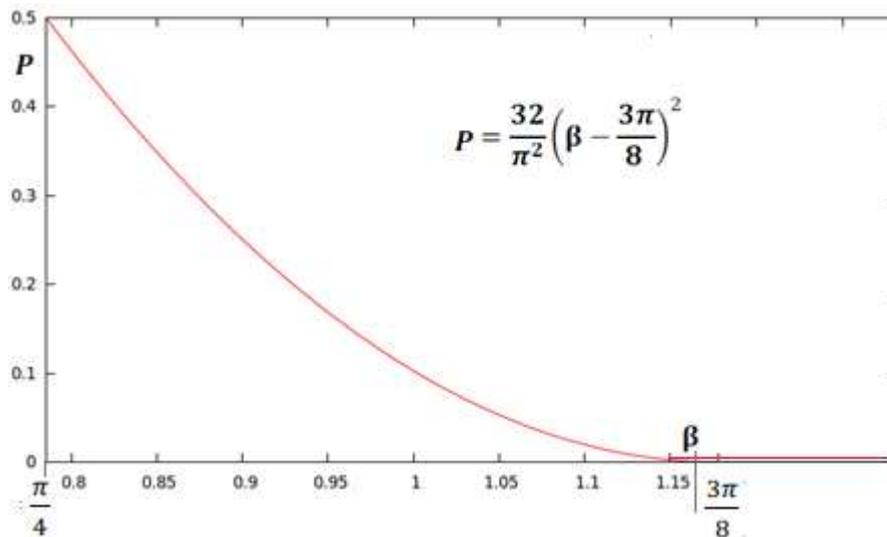


Figure 23 La fonction de probabilité pour le gradient d'énergie

### Remarque 1:

- ❖ Si  $\beta = \frac{\pi}{4} \Rightarrow P((\vec{X}, \vec{N}_1) > (\vec{X}, \vec{N}_2)) = \frac{1}{2}$

❖ Si  $\beta = \frac{3\pi}{8} \Rightarrow P((\vec{X}, \vec{N}_1) > (\vec{X}, \vec{N}_2)) = 0$

La probabilité pour que G soit plus proche de  $N_1$  est de  $\frac{1}{2}$  si  $\beta = \frac{\pi}{4}$

En utilisant cet algorithme il y a donc une chance sur deux pour que l'envoi des paquets se passe vers le nœud faible en énergie.

**Remarque 2:**

Si la direction de l'axe Y est vers le haut (Y↑) comme le montre la figure 24 :

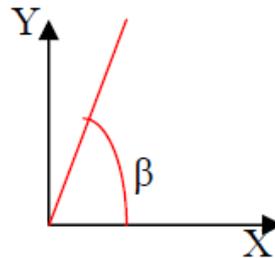


Figure 24 Le gradient d'énergie avec Y↑

On a alors  $G(x,y) = G(C_1, C_1 + 2 \varepsilon C_1)$

$\tan \beta = 1 + 2 \varepsilon$

Si  $\beta = \frac{3\pi}{8} \Rightarrow \tan \beta = \tan \frac{3\pi}{8} = 1 + \sqrt{2}$

$$\Rightarrow \varepsilon = \frac{\sqrt{2}}{2} < 1$$

Donc si  $\varepsilon > \frac{\sqrt{2}}{2}$  la probabilité est de 0.

### 3.4 Conclusion

Dans ce chapitre, nous avons ajouté un nouveau domaine d'application de l'image portant sur les réseaux de capteurs sans fil. Ainsi nous avons présenté quelques outils de traitement d'images tels que l'histogramme, la détection de contours et les filtres de convolution puis nous avons proposé une analogie entre un réseau de capteurs et le niveau de gris d'une image en regardant les capteurs du réseau comme une carte de pixels noirs, blancs et gris.

Nous avons défini également une matrice d'énergie et un gradient d'énergie obtenu par le filtre de Sobel. De plus nous avons fait une étude mathématique pour montrer que le vecteur du gradient d'énergie indique généralement la direction du voisin ayant la plus grande capacité énergétique.

La suite de notre travail sera de donner les algorithmes de routage qui utilisent cette propriété.

# Chapitre 4

## La gestion de l'énergie dans les RCSFs par les filtres de convolution

### Sommaire

---

4.1	Introduction .....	74
4.2	L'algorithme de routage basé sur le gradient d'énergie .....	74
4.2.1	L'algorithme de Sobel en RCSF .....	75
4.2.1.1	Les paramètres de l'algorithme .....	75
4.2.1.2	Les étapes de l'algorithme de routage par Sobel .....	75
4.2.1.2.1	Calculer le gradient d'énergie .....	75
4.2.1.2.2	L'implémentation de l'algorithme de traitement d'image dans le protocole AODV.....	76
4.2.2	Simulation de l'algorithme de routage basé sur Sobel .....	77
4.2.3	Conclusion de l'algorithme de routage basé sur Sobel .....	80
4.3	L'algorithme de routage basé sur le filtre moyen .....	81
4.3.1	Les paramètres de l'algorithme .....	82
4.3.2	Les étapes de l'algorithme de routage .....	82
4.3.3	Simulation de l'algorithme de routage basé sur le filtre moyen .....	83
4.3.3.1	OMNeT++ .....	83
4.3.3.2	Les simulations de l'algorithme par OMNeT++ .....	84
4.3.4	Conclusion de l'algorithme de routage basé sur le filtre moyen .....	89
4.4	Algorithme hybride.....	89
4.4.1	Introduction .....	89
4.4.2	Les secteurs de l'énergie et la matrice associée .....	90

4.4.3	L'algorithme de routage basé sur les deux filtres (moyen et Sobel) .....	91
4.4.3.1	L'algorithme hybride .....	91
4.4.3.2	Le routage des paquets .....	92
4.4.4	Simulation de l'algorithme hybride avec quatre nœuds .....	94
4.4.5	Simulation de l'algorithme hybride avec dix nœuds .....	105
4.4.6	Conclusion de l'algorithme hybride .....	114
4.5	Conclusion .....	115

## 4.1 Introduction

Le principal facteur pour la durée de vie d'un réseau de capteurs est l'énergie. Chaque nœud doit être conçu pour gérer ses ressources énergétiques afin de maximiser la durée de vie totale du réseau.

Ces dernières décennies, plusieurs protocoles de routage efficace en énergie ont été proposés pour les réseaux de capteurs. L'objectif principal de la plupart de ces protocoles est de trouver des routes qui sont économes en énergie, de sorte que la durée de vie du réseau soit maximisée. Pour atteindre cet objectif, de nombreux protocoles ont été développés en utilisant différentes stratégies pour obtenir des chemins. Certains de ces protocoles utilisent le clustering, d'autres utilisent les fonctions, les équations, les arbres de routage optimal, les techniques de multi routes ou des mécanismes pour éviter les trous.

L'algorithme 'Maximum lifetime energy routing' [58] permet de réduire la consommation d'énergie en utilisant les liens qui ont le coût le plus bas. Ainsi, cet algorithme calcule le coût de tous les liens à l'aide d'une fonction définie par l'énergie résiduelle du nœud. En d'autres termes, il s'agit de trouver la meilleure fonction des liens en terme de coût afin de trouver le meilleur chemin.

[59] présente une méthode à l'aide d'équations différentielles analogue aux équations de Maxwell pour établir la communication dans les régions ayant le plus d'énergie résiduelle et en évitant les régions avec des trous où il y a moins d'énergie résiduelle [61]. De même, les auteurs de [60] ont le même objectif en utilisant l'ellipse pour éviter les trous.

Dans ce chapitre nous avons proposé trois algorithmes de routage basés sur le produit de convolution utilisé en traitement d'image.

## 4.2 L'algorithme de routage basé sur le gradient d'énergie

Cette partie présente un nouvel algorithme de routage adaptatif basé sur l'algorithme de traitement d'image qui utilise le gradient pour choisir la route la plus économe en terme d'énergie [68].

A l'aide des techniques de traitement d'image et en employant l'algorithme de Sobel, le capteur peut calculer le gradient de l'énergie entre lui-même et ses voisins. Par conséquent, ce gradient donne la direction de transmission de paquets de la source vers la destination en évitant les régions pauvres en énergie.

### 4.2.1 L'algorithme de Sobel en RCSF

Pour chaque capteur de  $N_{i,j}$ , il existe un gradient d'énergie  $G_{i,j}$ . Ce gradient sera utilisé dans l'algorithme de routage proposé afin de trouver la route ayant la capacité de batterie la plus élevée.

#### 4.2.1.1 Les paramètres de l'algorithme

Voici les paramètres de l'algorithme basé sur le filtre Sobel

- ❖  $E$  : l'image en niveaux de gris.
- ❖  $X_i, Y_i$  : les positions du capteur  $N_i$ .
- ❖  $C_i$  : Les capacités des batteries pour le nœud  $N_i$ .
- ❖  $M$  : la matrice d'énergie.
- ❖  $K_x, K_y$  : les matrices de Sobel.
- ❖  $G_i$  : le gradient d'énergie de  $N_i$ .

#### 4.2.1.2 Les étapes de l'algorithme de routage par Sobel

Notre algorithme se déroulera en utilisant l'algorithme de Sobel à travers les étapes suivantes :

##### 4.2.1.2.1 Calculer le gradient d'énergie

1. Récupérer les capacités des batteries  $C_i$  et les positions des voisins  $N_i$ .
2. Créer une  $M$  matrice  $3 \times 3$ , où  $C$  représente la capacité de la batterie du capteur  $N_0$ . Puis, on divise le voisinage de  $N_0$  en huit secteurs comme le montre la figure 25. Pour chaque secteur, le capteur calcule la capacité moyenne  $M_i$  pour compléter la matrice :

$$M = \begin{pmatrix} M_1 & M_2 & M_3 \\ M_8 & C & M_4 \\ M_7 & M_6 & M_5 \end{pmatrix}$$

3. Pour calculer le gradient d'énergie, nous utilisons ces produits de convolution suivant :

$$\begin{cases} G_x = M * K_x \\ G_y = M * K_y \end{cases}$$

qui calculent le gradient de l'intensité de l'image de chaque pixel. Ainsi, selon l'opérateur Sobel, nous pouvons utiliser ces produits pour obtenir le gradient de l'énergie.

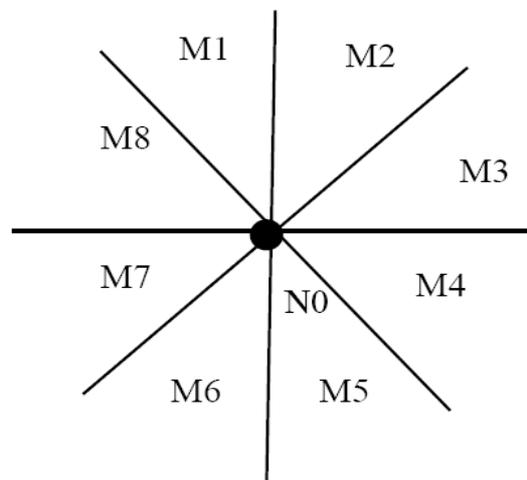


Figure 25 Les secteurs d'énergie autour de N0

#### 4.2.1.2.2 L'implémentation de l'algorithme de traitement d'image dans le protocole

##### AODV

Nous supposons que chaque nœud  $N(x, y)$  calcule le gradient d'énergie  $G_{x,y}$  par la méthode décrite dans le paragraphe 3.3.3.3.2.

Dans cet algorithme nous utilisons le protocole AODV où la source diffuse une requête RREQ à tous les voisins et il attend une requête RREP de la destination.

Quand un paquet RREP est reçu d'un nœud  $N(x', y')$ , le cosinus de l'angle  $(V(x', y'), G(x, y))$  est calculé par le produit scalaire:

$$\cos(\vec{V}_{x',y'}, \vec{G}_{x,y}) = \frac{\vec{V}_{x',y'} \times \vec{G}_{x,y}}{\|\vec{V}_{x',y'}\| \times \|\vec{G}_{x,y}\|}$$

Où  $V(x', y')$  est le vecteur défini par les nœuds  $N(x, y)$ ,  $N(x', y')$  comme le montre la figure 26.

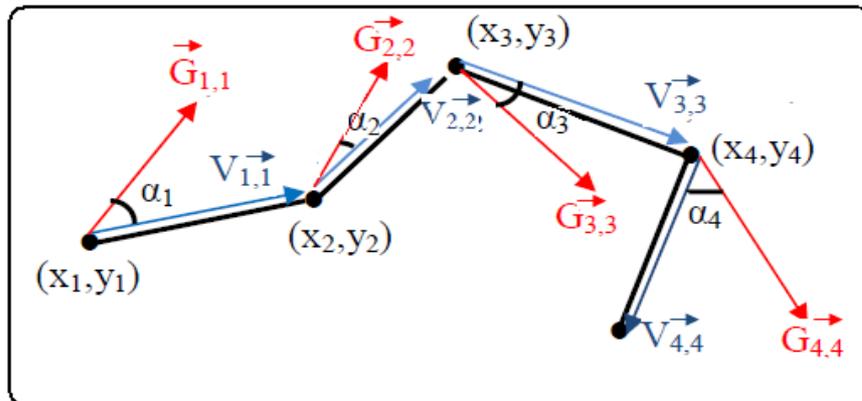


Figure 26 Le routage par la direction du gradient

Avec RREQ, la valeur absolue minimale (non-zéro) de cosinus pour la route est conservée. Lorsque le nœud source reçoit le message RREQ, ce message contient le cosinus minimal pour le chemin. Cette valeur représente l'angle le plus grand entre les deux vecteurs. Le nœud source va choisir le chemin contenu dans RREQ avec le cosinus maximal.

Nous avons maximisé la durée de vie de l'ensemble du réseau en choisissant de router les paquets dans une direction croissante en énergie. Afin de réduire l'énergie d'un nœud, nous essayons de limiter les calculs et les communications avec seulement huit voisins en utilisant l'algorithme de Sobel qui calcule le gradient de l'énergie.

#### 4.2.2 Simulation de l'algorithme de routage basé sur Sobel

Nous donnons un exemple pour montrer l'efficacité de cet algorithme. Nous prenons la topologie dans la figure 27 qui représente un réseau de cinq nœuds: N0 est la source et N3 est la destination :

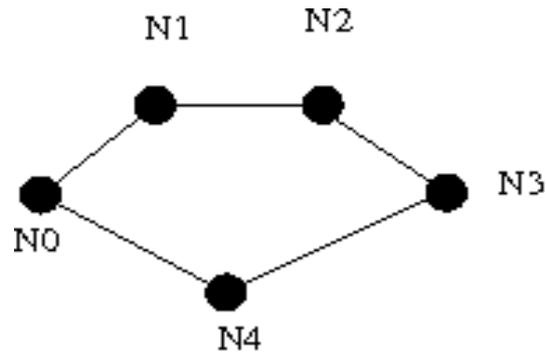


Figure 27 La topologie du réseau de notre algorithme par Sobel

Il existe deux chemins possibles entre la source et la destination:

$P_1 = (N_0, N_1, N_2, N_3)$

$P_2 = (N_0, N_4, N_3)$

Nous présentons deux exemples: dans le premier tous les nœuds ont une énergie égale à 10, dans le deuxième, la valeur énergétique de N4 est égal à 5.

$P_1 = (N_0, N_1, N_2, N_3)$     example1     $E(N_0) = E(N_1) = E(N_2) = E(N_3) = E(N_4) = 10$   
 $P_2 = (N_0, N_4, N_3)$     example2     $\left\{ \begin{array}{l} E(N_0) = E(N_1) = E(N_2) = E(N_3) = 10 \\ E(N_4) = 5 \end{array} \right.$

Les valeurs de gradient, calculées avec la méthode de Sobel:

• Dans l'exemple 1:

	$G_0$	$G_1$	$G_2$	$G_3$	$G_4$
x	20	10	-10	-20	0
Y	0	10	10	0	-20

• Dans l'exemple 2 :

	$G_0$	$G_1$	$G_2$	$G_3$	$G_4$
x	15	10	-10	-15	0
Y	-5	10	10	-5	-20

Dans le premier exemple, les vecteurs  $V_i$ ,  $G_i$  selon les deux routes sont :

$$\begin{array}{l}
 \text{P1} \left\{ \begin{array}{l}
 \text{N2} \quad V = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \quad g = \begin{pmatrix} -10 \\ 10 \end{pmatrix} \\
 \text{N1} \quad V = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad g = \begin{pmatrix} 10 \\ 10 \end{pmatrix} \\
 \text{N0} \quad V = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad g = \begin{pmatrix} 20 \\ 0 \end{pmatrix}
 \end{array} \right. \\
 \\
 \text{P2} \left\{ \begin{array}{l}
 \text{N4} \quad V = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad g = \begin{pmatrix} 0 \\ -20 \end{pmatrix} \\
 \text{N0} \quad V = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad g = \begin{pmatrix} 20 \\ 0 \end{pmatrix}
 \end{array} \right.
 \end{array}$$

- Dans le deuxième exemple:

$$\begin{array}{l}
 \text{P1} \left\{ \begin{array}{l}
 \text{N2} \quad V = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \quad g = \begin{pmatrix} -10 \\ 10 \end{pmatrix} \\
 \text{N1} \quad V = \begin{pmatrix} -1 \\ 0 \end{pmatrix} \quad g = \begin{pmatrix} 10 \\ 10 \end{pmatrix} \\
 \text{N0} \quad V = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad g = \begin{pmatrix} 15 \\ -5 \end{pmatrix}
 \end{array} \right. \\
 \\
 \text{P2} \left\{ \begin{array}{l}
 \text{N4} \quad V = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad g = \begin{pmatrix} 0 \\ -20 \end{pmatrix} \\
 \text{N0} \quad V = \begin{pmatrix} -1 \\ -1 \end{pmatrix} \quad g = \begin{pmatrix} 15 \\ -5 \end{pmatrix}
 \end{array} \right.
 \end{array}$$

Les angles utilisés dans cet algorithme sont déterminés par deux vecteurs : le vecteur qui suit la direction où la variation d'énergie est importante et le vecteur contenu dans RREQ. L'utilité de l'utilisation de ces angles est de choisir parmi les différentes routes possibles allant vers la destination celles qui sont les plus proches du chemin contenu dans RREQ. Autrement dit, nous cherchons les angles les plus petits.

Les cosinus du produit scalaire de deux vecteurs sont les suivants:

- Dans l'exemple 1:

P1	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>
CCS	0	-0,7	-0,7

P2	N <sub>4</sub>	N <sub>0</sub>
CCS	-0,7	-0,7

En P1, la valeur absolue minimale (non-zéro) de cosinus est: 0,7

En P2, la valeur absolue minimale (non-zéro) de cosinus est: 0,7

Le maximum de ces deux valeurs est: 0,7.

Dans cet exemple, nous avons choisi la route P1 avec la valeur absolue maximale de cosinus.

- Dans l'exemple 2 :

P1	N <sub>2</sub>	N <sub>1</sub>	N <sub>0</sub>
cos	0	-0,7	-0,8

P2	N <sub>4</sub>	N <sub>0</sub>
cos	-0,7	-0,4

En P1, la valeur absolue minimale (non-zéro) de cosinus est: 0,7

En P2, la valeur absolue minimale (non-zéro) de cosinus est: 0,4

Le maximum de ces deux valeurs est: 0,7.

Dans cet exemple, nous avons choisi la route P1 avec la valeur absolue maximale de cosinus.

Par conséquent, dans les deux cas, le nœud source N0 va choisir la route P1 parce que le nœud source choisit la route avec le cosinus maximal, ce qui signifie, le chemin avec une variation d'énergie maximale.

Dans cet algorithme les routes sont sélectionnées en passant par les nœuds ayant des capacités énergétiques plus importantes que leurs voisins. En revanche, le protocole de routage AODV va choisir la route P2 car il est fondé sur le principe de vecteurs de distance (le nombre des sauts entre l'émetteur et le récepteur). De plus, la capacité énergétique n'appartient pas à sa table de routage qui contient l'adresse de la destination, le nœud suivant et la distance en nombre de nœuds nécessaires pour atteindre la destination.

### 4.2.3 Conclusion de l'algorithme de routage basé sur Sobel

Le protocole de routage AODV ne possède pas le paramètre énergie dans sa table de routage. Donc, il n'assure pas un meilleur chemin entre la source et la destination en terme de préservation d'énergie.

Dans ce travail, nous avons adapté le protocole de routage AODV aux réseaux de capteurs sans fil. Rappelons que les capteurs sont alimentés par batterie. Dans cette

configuration, le chemin consommant le moins d'énergie est utilisé par nos modifications d'AODV.

Nous avons proposé dans cet algorithme un nouveau mécanisme de routage défini par la direction du meilleur gradient d'énergie du réseau. Ce gradient est obtenu avec l'algorithme de Sobel de traitement de l'image. Nous avons représenté la capacité énergétique du réseau comme une image en niveaux de gris.

### **4.3 L'algorithme de routage basé sur le filtre moyen**

Dans ce chapitre, nous introduisons un nouvel algorithme de routage centralisé, dans lequel nous utilisons le clustering et les matrices de convolution pour sélectionner le chemin le plus efficace en terme d'énergie.

Le clustering est une méthode répandue pour partitionner le réseau en un certain nombre de grappes, plus homogène selon une métrique spécifique ou une combinaison de paramètres, pour former une topologie virtuelle. Les clusters sont généralement identifiés par un nœud particulier appelé cluster-head. Cela permet la coordination entre les membres du cluster. Le cluster est sélectionné pour jouer un rôle très particulier avec une métrique ou une combinaison de paramètres.

Dans cet algorithme, le cluster-head est choisi pour avoir la meilleure capacité en terme de calcul et de ressources énergétiques. Nous avons utilisé le produit des matrices de convolution inspiré par les filtres de convolution qui sont utilisés en traitement d'images. Ce produit est calculé sur le cluster-head [29].

L'algorithme par filtre moyen utilise le clustering et le traitement d'image pour trouver le meilleur chemin pour envoyer des paquets de la source vers la destination. L'envoi de l'énergie des capteurs s'effectue par des communications UDP car il présente l'avantage d'une vitesse supérieure et d'un coût réduit par rapport au TCP. Dans notre algorithme, le choix de cluster-head CH se fait par l'administrateur [78].

Les capteurs recevant un taux d'énergie d'un bon niveau, supérieur au seuil de l'énergie (ET), vont envoyer des paquets UDP et participer au routage.

### 4.3.1 Les paramètres de l'algorithme

Voici les paramètres de notre algorithme basé sur le filtre moyen :

- $X_i, Y_i$  : la position du capteur  $N_i$  pour  $i$  donné.
- CH : cluster-Head.
- $BC_i$  : les capacités des batteries pour les nœuds  $i$ .
- ET : le seuil d'énergie, si la batterie du capteur est inférieure à cette valeur, la transmission des paquets UDP ne se fera pas.
- SEID (Sending Energy Information Delay) : le délai entre deux paquets d'information.
- T : le temps d'envoi d'un paquet d'énergie.
- R : le rayon pour les secteurs de l'énergie.
- K : la matrice noyau.
- $M_i$  : la matrice d'énergie de  $N_i$ .
- $ERP_i$  (Energy Routing Parameter) : le paramètre du routage d'énergie :  $ERP_i = K * M_i$

### 4.3.2 Les étapes de l'algorithme de routage

Nous allons présenter les différentes étapes de l'algorithme de routage par filtre moyen :

#### A. L'envoi d'informations sur l'énergie

Si la capacité  $BC_i$  change et

Si ( $T > SEID$ )

Alors on envoie ( $BC_i$  et  $X_i, Y_i$ ) par UDP à CH.

#### B. Le processus de calcul de convolution

Nous présentons ici la partie de l'algorithme effectuée par le cluster-head :

- 1- Le CH récupère les informations d'énergie de  $N_i$
- 2- Il met à jour la table d'énergie qui contient :
  - Les adresses IP de  $N_i$
  - La position  $X_i, Y_i$
  - La capacité  $BC_i$
- 3- Le CH calcule :
  - La matrice d'énergie moyenne  $M_i$  de  $N_i$

- Le produit de convolution  $ERP_i = K * M_i$

Où  $M_i$  est une matrice  $3 \times 3$ ,  $BC_i$  sont les capacités des batteries de  $N_i$  et de ses huit voisins situés en 8 secteurs.

4- Le CH envoie  $ERP_i$  du capteur  $N_i$

### C. Le routage des paquets et le processus du routage IP.

Cette partie de l'algorithme se déroule dans chaque capteur

Chaque capteur  $N_i$  récupère  $ERP_i$  sur UDP

Si ( $ERP_i < ET$ ) et si le paquet à envoyer n'est pas un paquet de contrôle

alors  $N_i$  supprime le paquet.

Si ( $ERP_i > ET$ ) alors la route est créée en utilisant AODV

### 4.3.3 Simulation de l'algorithme de routage basé sur le filtre moyen

Dans un premier temps, nous donnons un aperçu du simulateur OMNeT++ que nous avons utilisé et qui a été conçu principalement pour les réseaux de communication [71, 79]. Ensuite nous montrons les résultats de nos simulations qui prouvent la performance de cet algorithme.

#### 4.3.3.1 OMNeT++

OMNeT++ est une application développée par Andras Varga, chercheur à l'université de Budapest [80]. Il possède une architecture générique de sorte qu'il peut être utilisé pour la modélisation des réseaux câblés, les communications sans fil, la modélisation de protocoles, la modélisation des multiprocesseurs de systèmes distribués, l'évaluation des aspects de la performance des systèmes logiciels complexes, la validation d'architectures matérielles, etc....

Il est totalement paramétrable programmable et modulaire. Il fournit des outils, des modèles et des infrastructures qui aident à réaliser une simulation théorique. Les modèles sont assemblés à partir des composants réutilisables appelés *modules*. Ces modules peuvent être combinés de différentes manières.

Les simulations par OMNeT++ peuvent être exécutées sous différentes interfaces d'utilisateur : interfaces graphiques, interfaces d'animation et les interfaces utilisateurs de

commande en ligne qui sont les meilleurs. De plus, il est disponible pour les systèmes d'exploitation les plus courants (Linux, Mac OS / X, Windows).

Nous avons utilisé ce simulateur parce qu'il est conçu principalement pour les réseaux de communication et parce qu'il contient les paquets INETMANET qu'on a utilisé pour nos simulations.

#### 4.3.3.2 Les simulations de l'algorithme par OMNeT++

Notre algorithme a été implanté sur OMNeT ++ pour un réseau de quatre nœuds où N1 est la source et N0 est la destination comme le montre la figure 28. Nous avons simulé avec les mêmes paramètres que le dispositif Nano WiReach [81] qui utilise le protocole MAC IEEE 802.11. Sa consommation courante est de 250 mA en transmission Tx, 190 mA en réception Rx, 8 mA en veille et 8 mA en mode inactif. Nous utilisons UDP pour envoyer des messages de contrôle et AODV comme un protocole de routage.

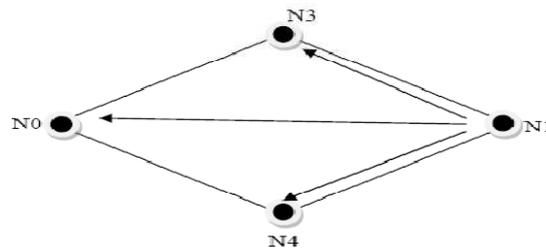


Figure 28 La topologie du réseau de notre algorithme par filtre moyen

Les paquets UDP sont envoyés de N1 à N0 comme une rafale (Burst) d'une fréquence de 0,01s et la taille des paquets est de 512 octets. Les résultats des simulations ont été obtenus à partir d'un ensemble de dix simulations. Les tables 2, 3, 4 et 5 montrent la moyenne de ces dix simulations.

Ces tables présentent les résultats de nos simulations avec un délai de simulation de 1500s, une période d'envoi des informations d'énergie (SEID) respectivement de 0.1s, 0.5s, 1s et 1.5s, et un seuil d'énergie respectivement de : 0, 25, 50, 75 et 100. Les valeurs initiales de l'énergie donnée à N0, N1, N3, et N4 dans nos simulations sont respectivement de 400, 200, 20 et 100. Les deux premières tables indiquent les résultats des simulations pour le nombre de paquets envoyés.

ET	0.1 s				0.5 s			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$1.43 e^{+4}$	$5.25 e^{+4}$	$3.41 e^{+4}$	$3.28 e^{+4}$	$2.95 e^{+3}$	$4.88 e^{+4}$	$2.47 e^{+4}$	$2.69 e^{+4}$
25	$1.43 e^{+4}$	$5.25 e^{+4}$	$9.75 e^{+3}$	$3.22 e^{+4}$	$2.95 e^{+3}$	$4.88 e^{+4}$	$1.99 e^{+3}$	$2.85 e^{+4}$
50	$1.43 e^{+4}$	$5.25 e^{+4}$	$1.09 e^{+4}$	$8.38 e^{+3}$	$2.96 e^{+3}$	$4.88 e^{+4}$	$2.10 e^{+3}$	$2.02 e^{+3}$
75	$1.62 e^{+3}$	572.000	$1.20 e^{+3}$	$1.17 e^{+3}$	$1.07 e^{+3}$	392.000	691.000	905.000
100	$1.60 e^{+3}$	571.000	$1.17 e^{+3}$	$1.18 e^{+3}$	$1.03 e^{+3}$	377.000	664.000	871.000

Table 2. Le nombre de paquets envoyés avec un SEID de 0.1s et de 0.5s

ET	1 s				1.5 s			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$1.51 e^{+3}$	$4.83 e^{+4}$	$2.56 e^{+4}$	$2.37 e^{+4}$	$1.03 e^{+3}$	$4.82 e^{+4}$	$2.47 e^{+4}$	$2.40 e^{+4}$
25	$1.51 e^{+3}$	$4.83 e^{+4}$	$1.08 e^{+3}$	$2.45 e^{+4}$	$1.03 e^{+3}$	$4.82 e^{+4}$	891.000	$1.95 e^{+4}$
50	$1.51 e^{+3}$	$4.83 e^{+4}$	$1.06 e^{+3}$	$1.11 e^{+3}$	$1.03 e^{+3}$	$4.82 e^{+4}$	764.000	780.000
75	237.000	93.000	203.000	158.000	85.000	36.000	68000	70.000
100	237.000	93.000	203.000	158.000	85.000	36.000	68000	70.000

Table 3. Le nombre de paquets envoyés avec un SEID de 1s et de 1.5s

Lorsque le seuil d'énergie est inférieur à 75, on remarque que le nombre de paquets envoyés ne change pas pour le nœud source et le nœud destination quelque soit la valeur de SEID. Le nombre de paquets envoyés confirme que c'est N1 qui diffuse les paquets UDP et que N0 envoie uniquement des paquets de contrôle comme le montre la figure 29.

Lorsque le seuil d'énergie égal à 0, les paquets sont transmis par N3 et N4, mais avec un seuil d'énergie de 25, les paquets sont transmis principalement par le nœud N4.

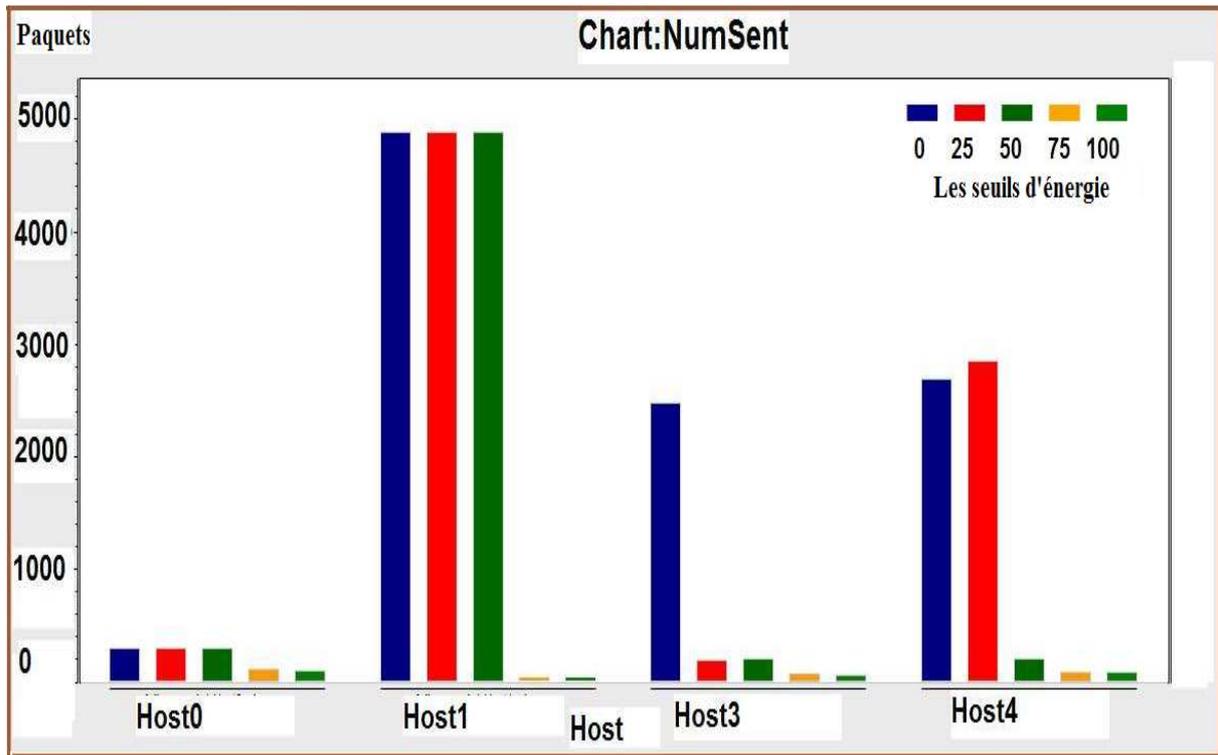


Figure 29 Le nombre de paquets envoyés avec un délai de 0.5s

En revanche, à un seuil d'énergie supérieur à 75, le nombre de paquets envoyés ne change pas pour tous les nœuds quelque soit la valeur de SEID (voir tables 2 et 3).

La figure 30 suivante représente le pourcentage de l'énergie restante dans les quatre nœuds avec un SEID de 0,5s.

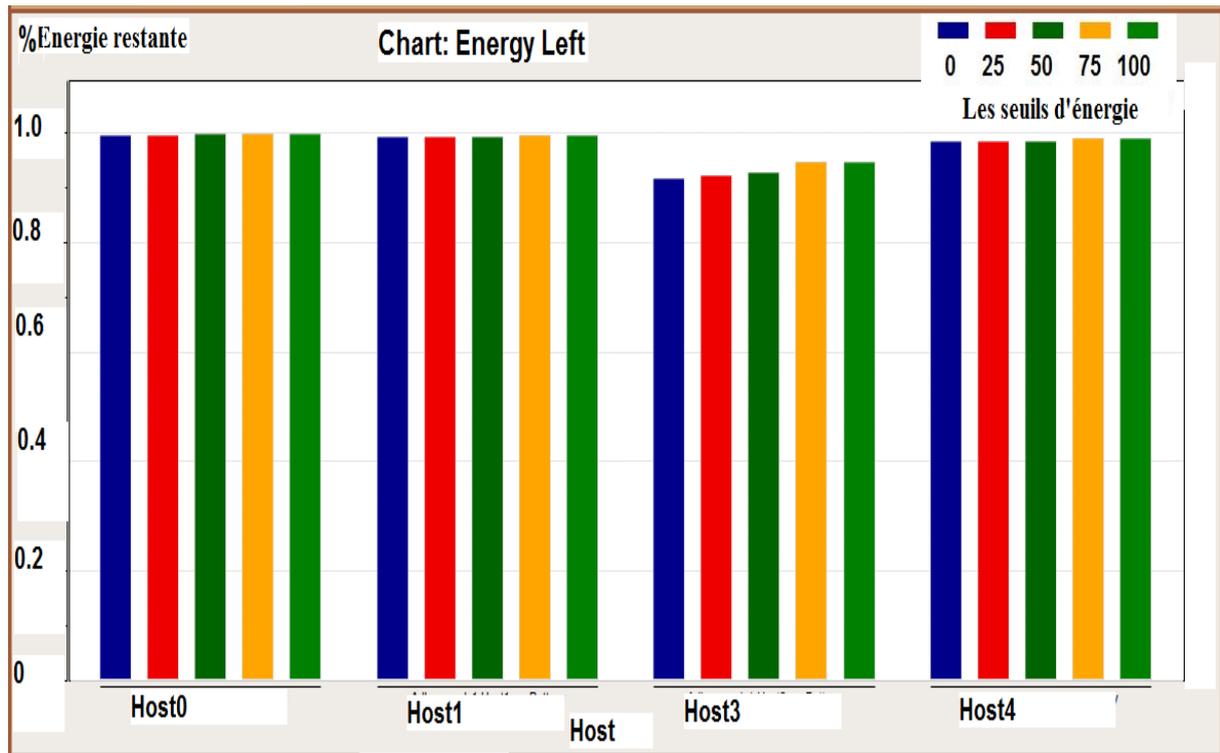


Figure 30 L'énergie restante avec un délai de 0.5s

Lorsque le seuil d'énergie est supérieur ou égal à 75, le nombre de paquets transmis par N3 et N4 arrive à zéro. Cela montre que l'énergie restante autour des nœuds N3 et N4 est inférieure au seuil.

Les deux tables 4 et 5 indiquent les pourcentages de l'énergie restante dans les quatre nœuds par rapport aux valeurs initiales de leurs capacités énergétiques :

ET	0.1 s				0.5 s			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	9950	9910	9000	9800	9960	9920	9150	9830
25	9960	9920	9070	9810	9960	9930	9220	9840
50	9960	9920	9130	9830	9970	9930	9280	9860
75	9970	9940	9420	9880	9970	9950	9450	9890
100	9970	9940	9420	9880	9970	9950	9470	9890

Table 4. L'énergie restante avec un SEID de 0.1s et de 0.5s

ET	1 s				1.5 s			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	9960	9920	9170	9830	9960	9920	9180	9840
25	9970	9930	9240	9850	9960	9930	9260	9850
50	9970	9930	9300	9860	9970	9930	9300	9850
75	9990	9980	9830	9960	9990	9990	9900	9980
100	9990	9980	9830	9960	9990	9990	9900	9980

Table 5. L'énergie restante avec un SEID de 1s et de 1.5s

En analysant les tables on constate qu'avec un SEID de 0.1s, le nœud N3 a utilisé 10% de son énergie, mais avec un SIED de 0.5s et un seuil d'énergie de 0, il a utilisé seulement 8,5% de l'énergie. Cela montre que, lorsque le SEID est égal à 0.1s, beaucoup d'énergie est utilisée pour diffuser des paquets de contrôle d'énergie. Si nous augmentons le SEID jusqu'à 1s ou 1.5s, la consommation d'énergie ne change pas. Par conséquent, un SEID de 0.5s semble être un bon compromis. Nous concentrons notre analyse sur cette valeur.

Avec un seuil d'énergie de 0, le nœud N3 compte 8,5% de l'énergie utilisée, mais à un seuil d'énergie de 25, il a utilisé 7,8% de l'énergie et nous avons économisé 0,7% de l'énergie de N3. Cela montre que l'algorithme fonctionne. À un seuil d'énergie de 50, l'énergie économisée est égale à 1,3%, mais le nombre de paquets transmis par N4 diminue de façon brusque. Si nous continuons à augmenter le seuil d'énergie, le nœud arrête de faire passer les paquets. Dans cette situation, plus d'énergie sera économisée, mais la transmission sera interrompue.

Enfin, il semble que pour cette topologie les meilleurs paramètres sont obtenus avec un SEID de 0.5s et un seuil d'énergie de 25.

Par conséquent, ces résultats montrent la bonne performance de notre algorithme, car nous avons prolongé la durée de vie du nœud N3 en économisant 7% de sa capacité énergétique.

### 4.3.4 Conclusion de l'algorithme de routage basé sur le filtre moyen

Nous avons présenté un nouvel algorithme de routage en utilisant les matrices de convolution et le clustering afin d'obtenir un routage efficace en terme d'énergie. Nous avons étendu le simulateur réseau OMNET ++ pour s'adapter et pouvoir simuler notre algorithme. Nous avons implémenté cet algorithme dans OMNET ++ pour simuler un réseau de capteurs composé de quatre nœuds pour évaluer la performance énergétique efficace.

Les résultats des simulations ont montré que nous avons diminué avec succès la consommation d'énergie du réseau. Nous avons trouvé le bon délai d'envoi des informations concernant l'énergie (qui est égal à 0,5s).

## 4.4 Algorithme hybride

### 4.4.1 Introduction

Plusieurs solutions au niveau de la sous-couche MAC ont été proposés [62, 63, 64]. Cette couche permet aux nœuds d'avoir des phases de sommeil. Deux topologies sont utilisées par le protocole 802.15.4 [103, 104], la topologie en étoile [103, 104] et la topologie peer to peer [104]. Dans la topologie en étoile, les communications sont établies directement entre le nœud central (coordinateur) et les capteurs. Le coordinateur est le nœud qui initie et gère les communications réseau. La topologie peer-to-peer permet à chaque nœud du réseau de communiquer avec un autre nœud.

S-MAC (Sensor MAC) [31] est un autre protocole au niveau de la couche MAC qui est similaire à 802.11. Il utilise l'accès au médium CSMA / CA, RTS / CTS (Request-To-Send, Clear-To-Send) qui évite les collisions et les problèmes du nœud caché [82]. Ce protocole établit un mécanisme de distribution d'état de veille avec chaque nœud afin de réduire la consommation d'énergie et pour prolonger sa durée de vie. Chaque nœud doit coordonner et échanger des informations avec ses voisins pour choisir son propre cycle "veille / active".

Dans le contexte de mobilité et d'économie de ressources énergétiques de nœuds, l'auto-configuration est un problème important dans les réseaux de capteurs sans fil. Pour économiser les batteries des nœuds, il est nécessaire d'éteindre leurs émetteurs radio autant

que possible. Par contre, le problème de la synchronisation entre les nœuds par rapport à la distribution de périodes de veille apparaît. Une couche MAC est une solution qui permet aux nœuds d'avoir des phases de sommeil, sans perturber la communication. Plusieurs solutions au niveau de couche MAC ont été proposées pour économiser l'énergie. Le rôle des protocoles MAC est d'organiser l'accès au médium entre les nœuds qui souhaitent communiquer entre eux. Donc, il permet la réalisation d'une bonne coordination entre les nœuds, mais aussi la minimisation de la dissipation de leurs énergies.

Dans cette partie, nous présentons un algorithme hybride pour améliorer et compléter nos deux premiers algorithmes basés sur le filtre Sobel et le filtre moyen [68, 69]. Comme dans les deux premiers algorithmes, nous présentons le réseau de capteurs sans fil comme une image avec une échelle de gris pour obtenir un routage d'énergie efficace. Les pixels de l'image représentent les nœuds du RCSF et nous avons coupé l'image en sous-ensembles ou en zones autour d'un nœud central pour déterminer la répartition de l'énergie. Dans cet algorithme, nous avons utilisé des matrices et des produits de convolution avec deux filtres. Ces deux filtres sont utilisés dans les techniques de traitement d'image. Le premier (le filtre moyen) est efficace dans l'identification des nœuds qui vont participer au système de routage. Le second (le filtre de Sobel) sert à réaliser le routage des paquets à travers les routes qui passent par les zones claires (par les nœuds qui ont la plus grande capacité).

#### **4.4.2 Les secteurs de l'énergie et la matrice associée**

Nous procédons de la même manière que dans nos deux précédents algorithmes en coupant le voisinage d'un nœud en huit secteurs afin de comparer l'énergie de ce nœud par rapport à ses voisins comme le montre la figure 16 (paragraphe 3.3.3.1).

#### **4.4.3 L'algorithme de routage basé sur les deux filtres (moyen et Sobel)**

La défaillance d'un capteur peut avoir de graves conséquences sur la vie totale du réseau, il est donc nécessaire de n'impliquer que les nœuds qui ont une capacité énergétique supérieure à un certain seuil ET (Energy Threshold). Les capteurs qui ne participent pas au

routage vont se mettre en veille pendant une période  $T_{\text{sleep}}$ . Puis ils se réveillent pour obtenir des informations de la part de CH.

#### 4.4.3.1 L'algorithme hybride

Cet algorithme hybride se déroulera en quatre phases pour trouver la route la plus économique en énergie de la source vers la destination :

##### 1. L'envoi d'informations sur l'énergie

Si la capacité  $BC_i$  du nœud  $N_i$  change et

Si ( $T > SEID$ ) et le nœud n'est pas en mode sommeil :

Alors envoyer ( $BC_i$  et  $X_i, Y_i$ ) par UDP à CH.

##### 2. Les calculs de convolution

Voici les différentes opérations effectuées au niveau de cluster-head

1) Le CH récupère les informations d'énergie des nœuds  $N_i$

2) Il met à jour la table d'énergie qui contient :

- Les adresses IP des  $N_i$
- Les positions  $X_i, Y_i$
- La capacité des batteries  $BC_i$

3) Le CH calcule :

- La matrice d'énergie moyenne  $M_i$  de  $N_i$
- Le produit de convolution  $ERP_i = K * M_i$
- Le gradient  $G_{x_i}, G_{y_i}$

4) Le CH envoie  $ERP_i$  au nœud  $N_i$  via UDP.

##### 3. Le routage des paquets et le processus du routage IP.

Chaque capteur effectue les opérations suivantes :

Chaque capteur récupère  $ERP_i, G_{x_i}, G_{y_i}$  sur UDP

Si ( $ERP_i < ET$ )

Alors le capteur est en mode de sommeil durant  $T_{\text{sleep}}$

Et si le paquet envoyé n'est pas un paquet de contrôle

Alors le capteur reste en mode veille

4. Capteur reste en mode veille pendant le temps  $T_{sleep}$ , puis il demande une nouvelle valeur  $K * M_i$

Si  $(K * M_i > ET)$

Alors il met fin au mode veille

#### 4.4.3.2 Le routage des paquets

5.  $N_i$  calcule le vecteur  $V_i = (K_x * M_i - X_i, K_y * M_i - Y_i)$  dans le capteur  $N_i (X_i, Y_i)$

6.  $N_i$  calcule  $\cos \alpha_i = \cos (V_i, G_i)$  où  $G_i$  est le gradient

$$\cos \alpha_i = \text{COS} (V_i, G_i) = \frac{\vec{V}_i \times \vec{G}_i}{\|\vec{V}_i\| \times \|\vec{G}_i\|}$$

7. Le nœud  $N_i$  compare la valeur  $\cos \alpha_i$  à celle contenue dans le paquet du routage AODV RREQ. Si la valeur est plus petite, il remplace la valeur dans le paquet RREQ retransmis.
8. Si le paquet est la destination, il choisi parmi les paquets RREQ reçu la valeur  $\cos \alpha_i$  maximal.
9. La route est crée saut par saut par AODV avec la valeur  $\cos \alpha_i$  maximal.

Les figures 31 et 32 montrent les deux organigrammes qui représentent notre algorithme :

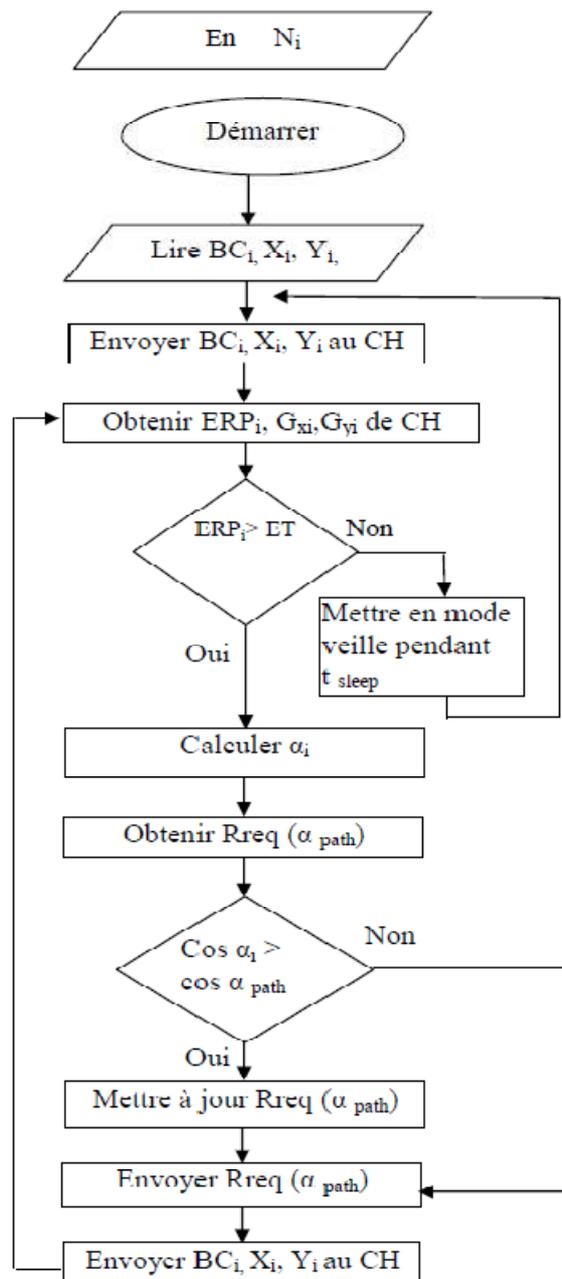


Figure 31 Les étapes de l'algorithme au niveau de Ni

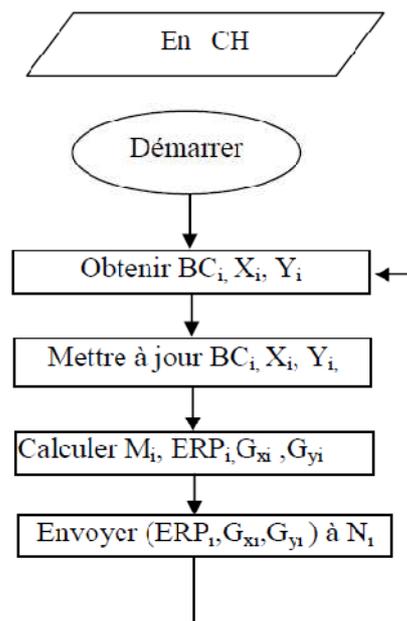


Figure 32 Les étapes de l'algorithme au niveau de CH

Par conséquent, nous pouvons résumer les trois phases précédentes de cet algorithme comme :

- 1 : obtenir les informations énergétiques de chaque nœud par le cluster-head.
- 2 : calculer la direction du gradient indiquant la route recherchée à travers laquelle passent les paquets et l'envoyer en retour.
- 3 : sélectionner la route

#### 4.4.4 Simulation de l'algorithme hybride

Notre travail est effectué dans un réseau de capteurs qui possède la topologie montrée dans la figure 28 (paragraphe 4.3.3.2) où N1 est la source, N3, N4 sont les passerelles et N0 est la destination.

Les résultats obtenus représentent la moyenne de vingt simulations effectuées par le simulateur OMNET ++ qui a été modifié par l'implémentation de deux noyaux (le filtre Sobel et le filtre moyen). Nous avons utilisé le dispositif Nano WiReach [81] comme un modèle. La capacité de consommation est de 250mA en émission, 190 mA en réception (typique), 8 mA en mode veille et 8 mA en mode inactif.

Nous avons modifié le protocole de routage AODV parce qu'il ne possède pas la capacité énergétique dans sa table de routage. Les paquets de contrôle ont une taille de 512 octets et sont envoyés en utilisant le protocole UDP. Le nœud source N1 envoie des paquets en rafale avec une fréquence de 0.01s avec un délai de veille de 0s, 50s, 100s et 150s. Les valeurs initiales de l'énergie donnée à N0, N1, N3 et N4 dans nos simulations sont respectivement de 40mA, 20mA, 2mA et 10mA. Le nœud N0 est le cluster-head dans notre configuration. Le seuil d'énergie a respectivement les valeurs de 0mA, 1mA, 2mA, 3mA, 4mA et 5mA. Le temps total de la simulation est de 1000 secondes.

Dans les tables suivantes 6, 7 et 8 nous voyons qu'il y a quelques paquets envoyés par le nœud N0. Comme N0 est la destination, il n'émet pas de paquets de contrôle. Dans la table 8 où le seuil d'énergie est de 4 et 5, nous constatons que le nombre de paquets qui font la route entre la source et la destination par les deux passerelles (N3 et N4) tend vers zéro lorsque le seuil d'énergie est supérieur ou égal à 4. En d'autres termes, la communication est interrompue si le seuil d'énergie est supérieur ou égal à 4.

délai en mode veille	Seuil (0)				Seuil (1)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$2.41 e^{+4}$	$1.06 e^{+5}$	$3.32 e^{+4}$	$9.72 e^{+4}$	$2.45 e^{+4}$	$1.06 e^{+5}$	$2.54 e^{+4}$	$9.33 e^{+4}$
50	$1.92 e^{+4}$	$7.09 e^{+4}$	$3.63 e^{+4}$	$6.21 e^{+4}$	$1.52 e^{+4}$	$6.66 e^{+4}$	$1.99 e^{+4}$	$5.81 e^{+4}$
100	$1.93 e^{+4}$	$7.95 e^{+4}$	$3.88 e^{+4}$	$6.00 e^{+4}$	$1.88 e^{+4}$	$8.53 e^{+4}$	$1.72 e^{+4}$	$8.48 e^{+4}$
150	$1.91 e^{+4}$	$7.87 e^{+4}$	$3.63 e^{+4}$	$6.17 e^{+4}$	$2.13 e^{+4}$	$1.06 e^{+5}$	$1.80 e^{+4}$	$1.07 e^{+5}$

Table 6. Le nombre de paquets envoyés avec un seuil d'énergie de 0 et de 1

délai en mode veille	Seuil (2)				Seuil (3)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$2.12 e^{+4}$	$8.50 e^{+4}$	$1.22 e^{+4}$	$6.02 e^{+4}$	$1.82 e^{+4}$	$6.75 e^{+4}$	$1.25 e^{+4}$	$2.36 e^{+4}$
50	$2.13 e^{+4}$	$1.06 e^{+5}$	$2.82 e^{+3}$	$1.19 e^{+5}$	$7.21 e^{+3}$	$5.11 e^{+4}$	$1.25 e^{+3}$	$3.74 e^{+4}$
100	$2.08 e^{+4}$	$1.06 e^{+5}$	$1.88 e^{+3}$	$1.21 e^{+5}$	$6.85 e^{+3}$	$4.53 e^{+4}$	603.000	$4.01 e^{+4}$
150	$1.88 e^{+4}$	$9.85 e^{+4}$	$1.03 e^{+3}$	$1.13 e^{+5}$	$6.65 e^{+3}$	$4.25 e^{+4}$	366.000	$4.00 e^{+4}$

Table 7. Le nombre de paquets envoyés avec un seuil d'énergie de 2 et de 3

délai en mode veille	Seuil (4)				Seuil (5)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	$2.21 e^{+4}$	$8.72 e^{+4}$	$1.35 e^{+4}$	$1.68 e^{+4}$	$2.56 e^{+4}$	$1.06 e^{+5}$	$1.41 e^{+4}$	$2.13 e^{+4}$
50	325.000	$2.49 e^{+4}$	306.000	41.000	321.000	$2.49 e^{+4}$	305.000	42.000
100	110.000	$1.29 e^{+4}$	23.000	98.000	110.000	$1.30 e^{+4}$	24.000	97.000
150	92.500	$1.03 e^{+4}$	83.000	19.000	92.500	$1.03 e^{+4}$	82.000	18.000

Table 8. Le nombre de paquets envoyés avec un ET de 4 et de 5

Nous mettrons l'accent sur le routage avec un seuil d'énergie inférieur à 4. Avec un seuil d'énergie entre 2 et 3, l'énergie a été préservée parce que le nombre de paquets envoyés par N4 baisse comme le montrent les figures 33 et 34.

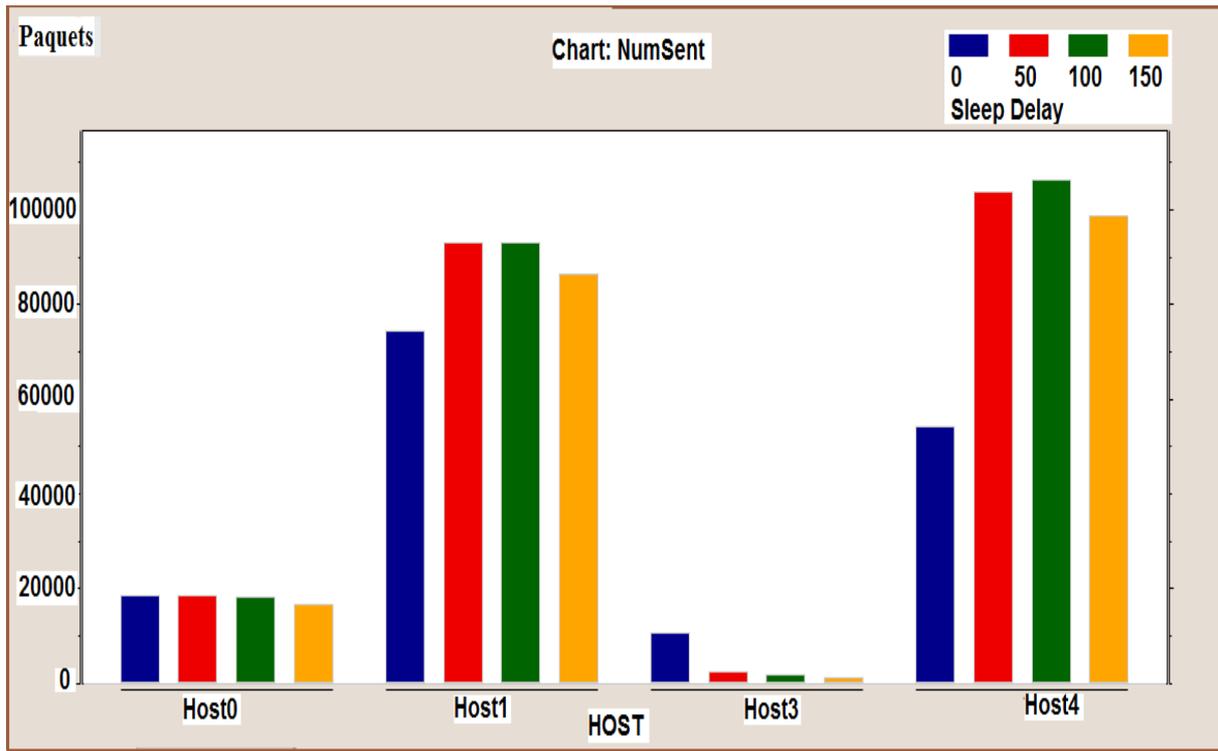


Figure 33 Le nombre de paquets envoyés avec un seuil d'énergie de 2

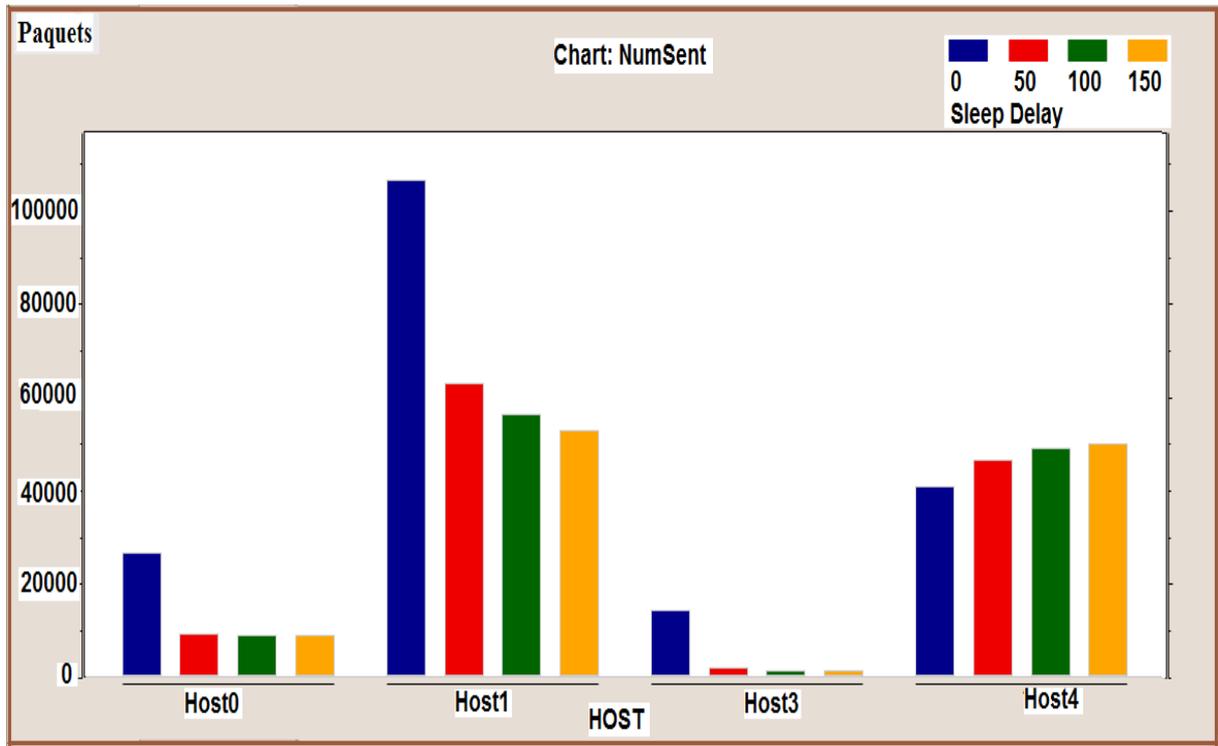


Figure 34 Le nombre de paquets envoyés avec un seuil d'énergie de 3

Les tables suivantes représentent le pourcentage d'énergie restante dans les nœuds du réseau. Nous constatons que la valeur énergétique restante en N3 est zéro parce que sa valeur

initiale est faible et le temps de la simulation total est 1000s. Par conséquent, nous allons nous concentrer sur l'autre passerelle, le nœud N4.

délai en mode veille	Seuil (0)				Seuil (1)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	0.909	0.826	0	0.603	0.911	0.830	0	0.609
50	0.870	0.694	0	0.307	0.884	0.666	0	0.301
100	0.867	0.736	0	0.358	0.868	0.799	0	0.443
150	0.866	0.734	0	0.353	0.912	0.830	0	0.609

Table 9. L'énergie restante avec un seuil d'énergie de 0 et de 1

délai en mode veille	Seuil (2)				Seuil (3)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	0.879	0.761	0	0.426	0.839	0.771	0	0.342
50	0.912	0.831	0	0.608	0.933	0.865	0	0.724
100	0.912	0.831	0	0.608	0.936	0.867	0	0.722
150	0.907	0.800	0	0.550	0.936	0.868	0	0.723

Table 10. L'énergie restante avec un seuil d'énergie de 2 et de 3

Dans les tables ci-dessus 9 et 10, nous constatons qu'avec un  $T_{\text{sleep}}$  égal à 0s et un seuil d'énergie entre 0 et 3, le nœud N4 perd son énergie. Par contre, il conserve son énergie si le seuil d'énergie est supérieur ou égal à 4, car il n'envoie pas de paquets (sauf les paquets de contrôle) comme le montre la table 11.

délai en mode veille	Seuil (4)				Seuil (5)			
	Host0	Host1	Host3	Host4	Host0	Host1	Host3	Host4
0	0.890	0.771	0	0.503	0.922	0.839	0	0.698
50	0.944	0.881	0	0.779	0.944	0.811	0	0.779
100	0.948	0.885	0	0.778	0.948	0.885	0	0.778
150	0.949	0.886	0	0.778	0.949	0.886	0	0.778

Table 11. L'énergie restante avec un seuil d'énergie de 4 et de 5

Avec un  $T_{\text{sleep}}$  de 50s et un seuil d'énergie de 1, le nœud N4 a utilisé 70% de son énergie. Avec un seuil d'énergie de 2, il en a seulement utilisé 40%, mais avec un seuil d'énergie de 3 ou 4 ou 5, il a utilisé 24% de sa capacité. Cela signifie, nous avons économisé 30% d'énergie avec la combinaison ( $T_{\text{sleep}} = 50\text{s}$ ,  $ET = 2$ ) comme le montrent les figures 35 et 36.

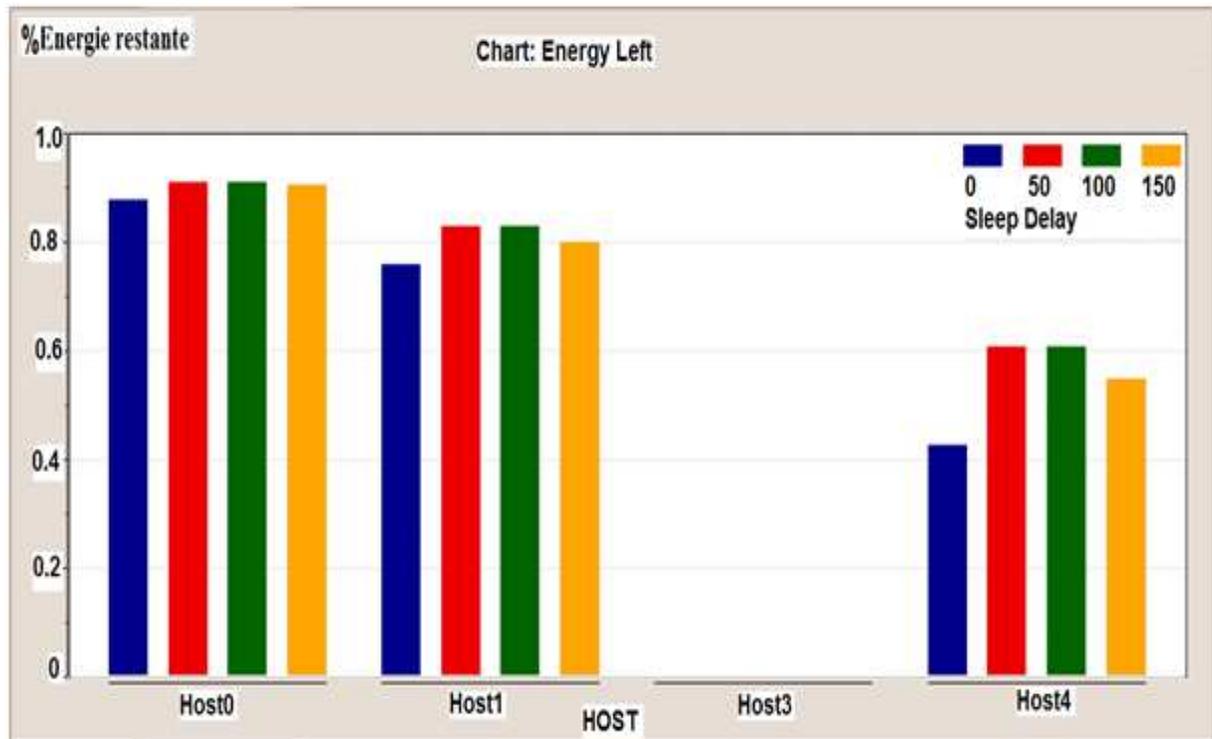


Figure 35 L'énergie restante avec un seuil d'énergie de 2

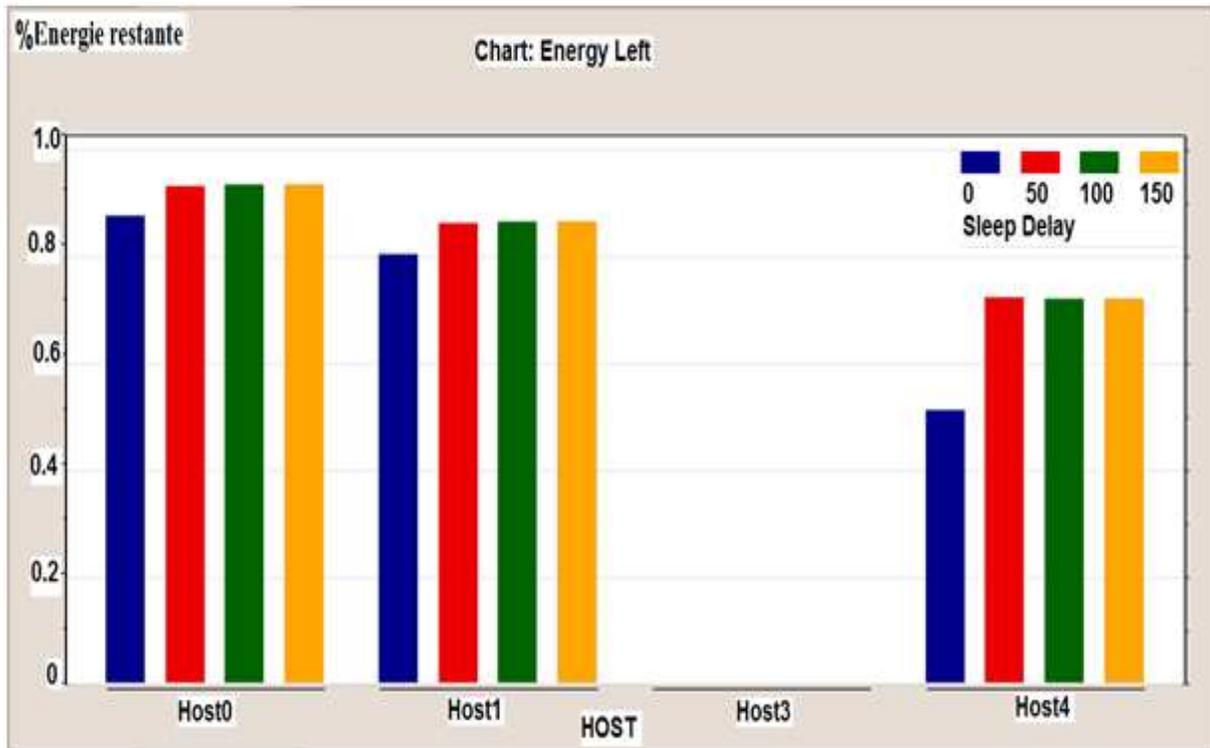


Figure 36 L'énergie restante avec un seuil d'énergie de 3

Avec un  $T_{\text{sleep}}$  de 100s et un seuil d'énergie de 1, le nœud N4 a consommé 55% de son énergie. Avec un seuil d'énergie de 2, il en a seulement utilisé 40% et avec un seuil d'énergie de 3 ou 4 ou 5, il a utilisé 24% de sa capacité. Cela signifie, nous avons économisé 15% d'énergie avec la combinaison  $T_{\text{sleep}} = 100\text{s}$ ,  $ET = 2$ .

Si nous augmentons le  $T_{\text{sleep}}$  jusqu'à 150s avec un seuil d'énergie de 0, le nœud N4 a utilisé 64% de son énergie. Avec un seuil d'énergie de 2, il a seulement utilisé 40% et avec un seuil d'énergie de 3 ou 4 ou 5, il a utilisé 24% de sa capacité. Mais le nombre de paquets transmis par N4 diminue de façon spectaculaire. Si nous continuons à augmenter le seuil d'énergie, le nœud s'arrête pour relayer les paquets, comme l'illustre la figure 37.

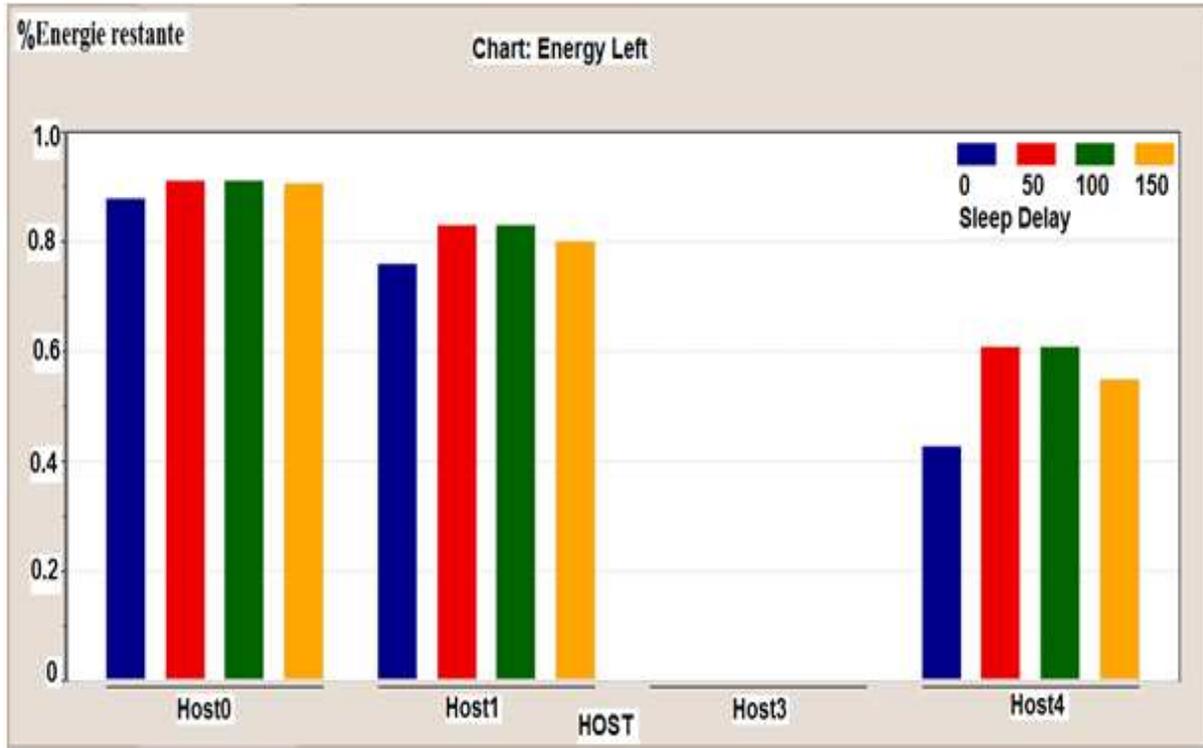


Figure 37 L'énergie restante avec un seuil d'énergie de 0

En ce qui concerne la consommation d'énergie au fil du temps, il est clair que si on augmente le temps de sommeil, nous économisons plus d'énergie comme le représentent les figures 38 et 39, qui indiquent la variation d'énergie du nœud N3 durant la période de la simulation de 1000s.

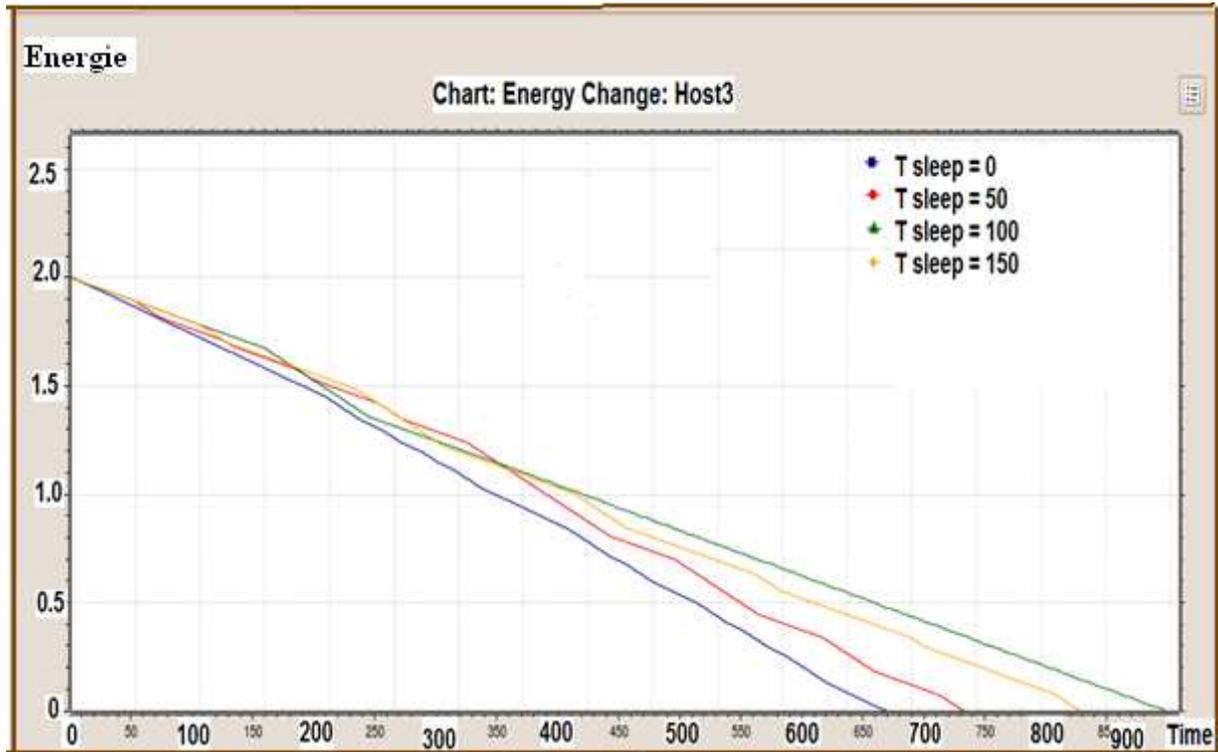


Figure 38 La consommation d'énergie de N3 avec un seuil d'énergie de 2

Avec un seuil d'énergie de 2mA et un temps de simulation de 1000s, on remarque que le nœud N3 cesse d'émettre car il est épuisé après:

- 672 secondes quand  $T_{\text{sleep}} = 0\text{s}$
- 739 secondes quand  $T_{\text{sleep}} = 50\text{s}$
- 882 secondes quand  $T_{\text{sleep}} = 100\text{s}$
- 823 secondes quand  $T_{\text{sleep}} = 150\text{s}$

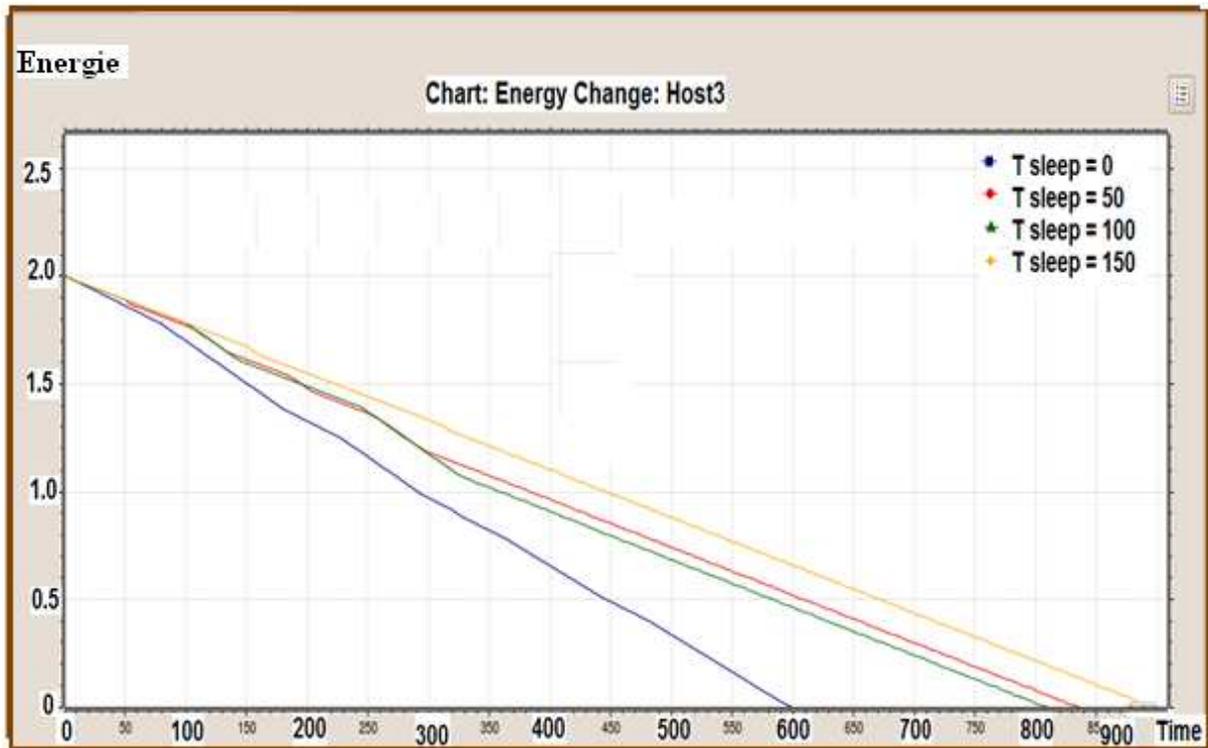


Figure 39 La consommation d'énergie de N3 avec un seuil d'énergie de 3

Avec un seuil d'énergie de 3 mA, le nœud N3 est épuisé après :

- 598 secondes quand  $T_{\text{sleep}} = 0\text{s}$
- 836 secondes quand  $T_{\text{sleep}} = 50\text{s}$
- 810 secondes lorsque  $T_{\text{sleep}} = 100\text{s}$
- 897 secondes quand  $T_{\text{sleep}} = 150\text{s}$

Notre principal objectif est de maximiser la durée de vie du réseau. Les deux figures 40 et 41 montrent la variation d'énergie du nœud N4 pendant le temps de simulation de 1000 secondes. Nous constatons que si nous augmentons le temps de sommeil jusqu'à 150s et si nous augmentons le seuil d'énergie de 0 à 3 on obtient de meilleurs résultats.

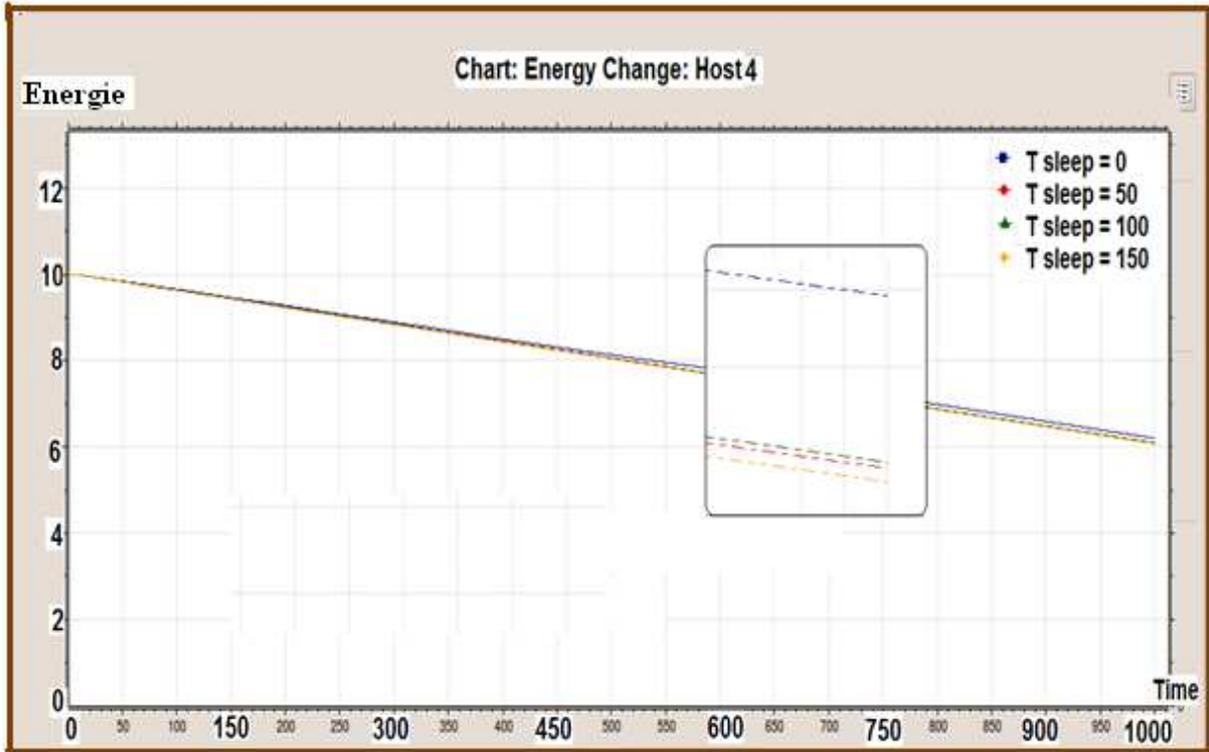


Figure 40 La consommation d'énergie de N4 avec un seuil d'énergie de 2

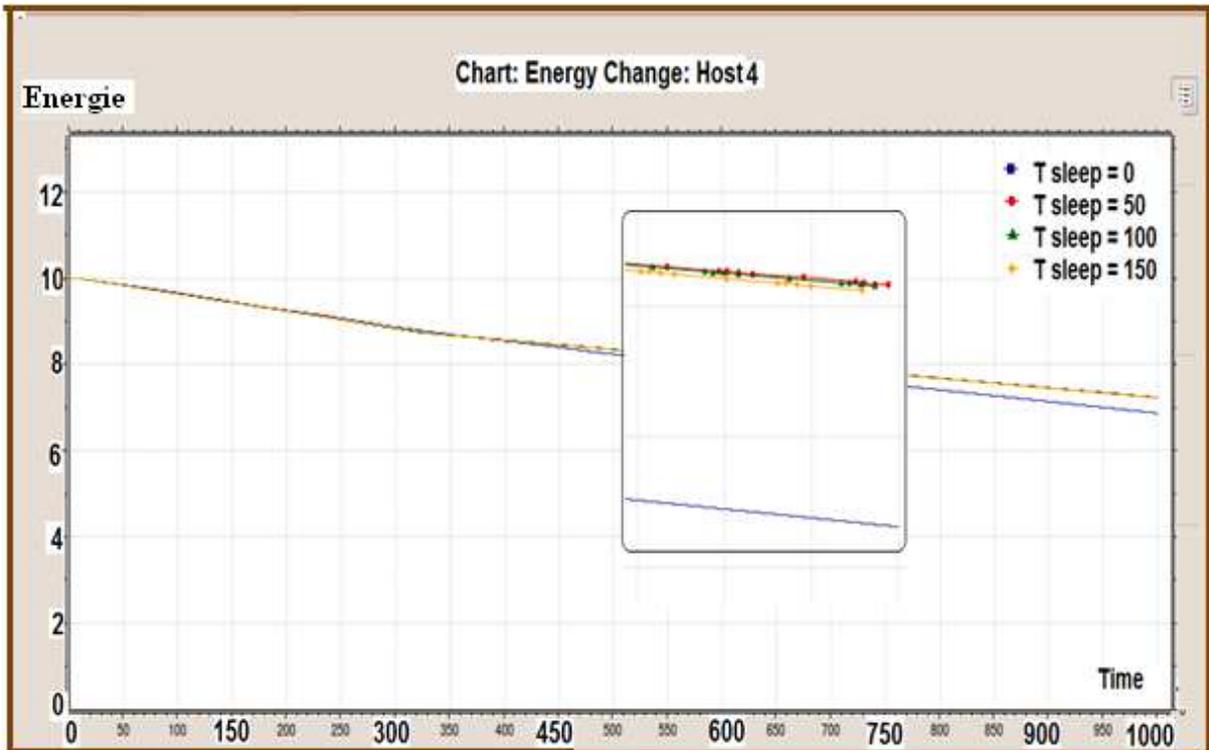


Figure 41 La consommation d'énergie de N4 avec un seuil d'énergie de 3

Par conséquent, nous avons constaté que les meilleurs résultats sont obtenus avec un  $T_{\text{sleep}}$  de 50s et un seuil d'énergie situé entre 1 et 2 car nous avons économisé 30% de l'énergie.

Ces résultats montrent la bonne performance de notre algorithme, car nous avons réussi à prolonger la durée de vie du nœud N4.

#### 4.4.5 Simulation de l'algorithme hybride avec dix nœuds

Nous présentons une analyse détaillée pour mieux évoluer la performance de notre algorithme. L'évaluation est faite par dix simulations avec le simulateur OMNeT++ de dix nœuds capteurs dont  $N_1$  est la source et  $N_0$  est la destination comme le montre la topologie représentée par la figure 42 :

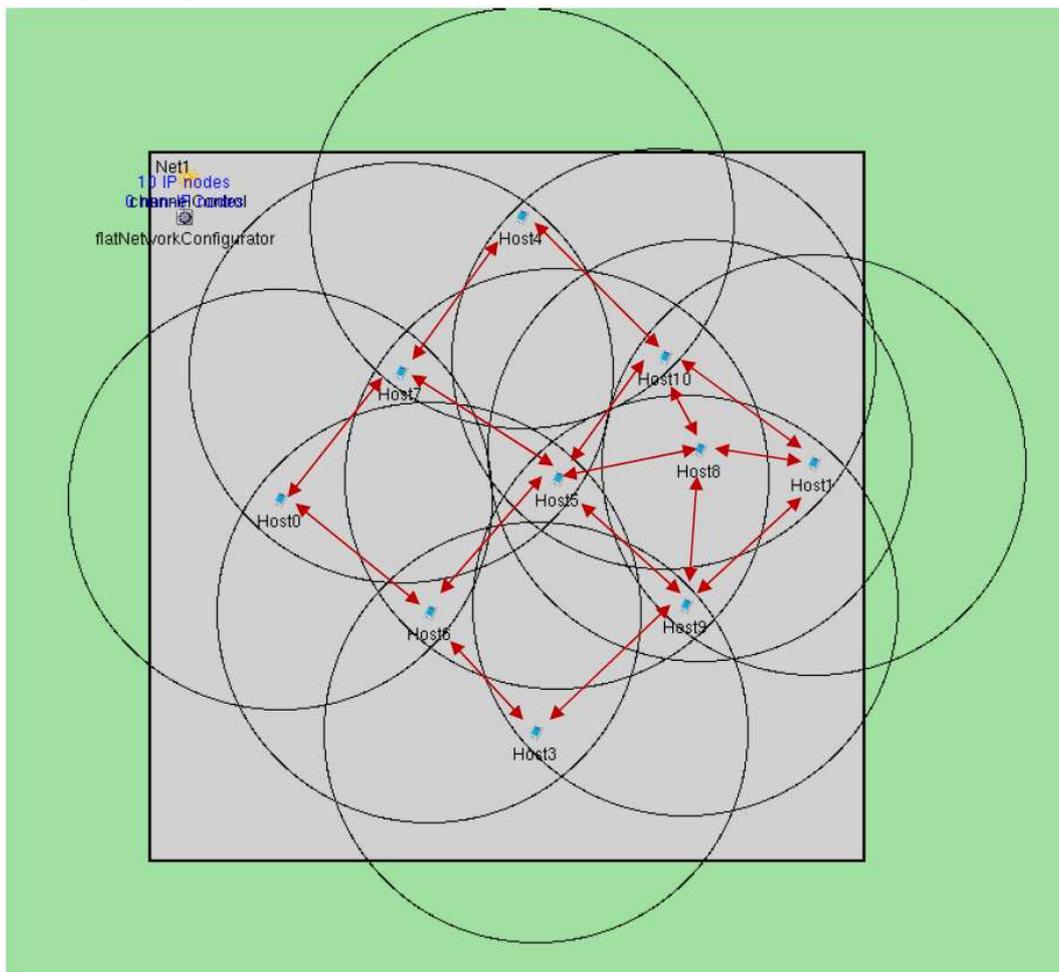


Figure 42 La topologie de notre réseau

Les capteurs sont déployés sur une surface de 1000m \* 1000m et chaque capteur a un rayon de communication de 300m avec une puissance d'envoi de 90 mW.

La table 12 suivante représente les positions et les capacités énergétiques des nœuds en mA:

	N <sub>0</sub>	N <sub>1</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>
N <sub>x</sub>	180	930	530	520	570	330	350	770	750	720
N <sub>y</sub>	490	440	840	90	420	670	340	220	640	290
Energie	50	30	30	2	10	40	2	10	30	1

Table 12. Les positions et les capacités énergétiques des capteurs

A l'aide de notre analogie ce réseau peut être représenté par une image en niveau de gris où chaque capteur a une intensité variante de 0 à 50 comme le montre la figure 43 :

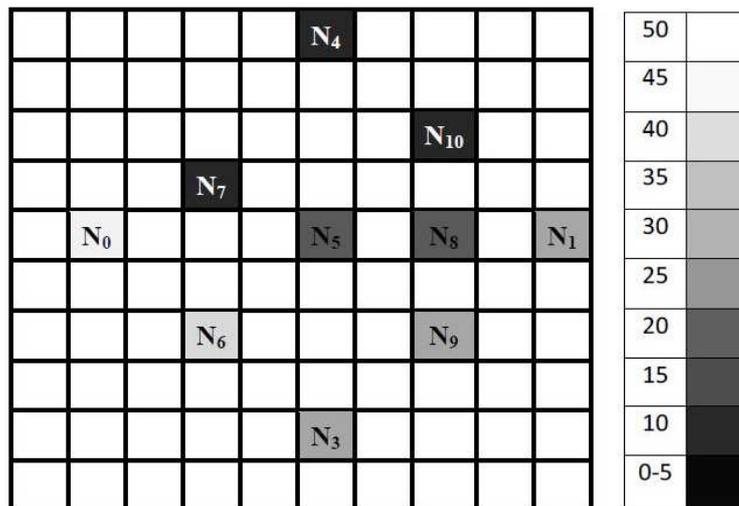


Figure 43 La représentation du réseau en niveau de gris

En appliquant l'algorithme de Sobel sur notre réseau on obtient les gradients  $G_x$  et  $G_y$  donnés par la table 13 suivante :

	N <sub>0</sub>	N <sub>1</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>
G <sub>x</sub>	42	-51	-10	-1	9	-10	-38	20	-10,	18
G <sub>y</sub>	38	29	-70	3	67	-30	58	58	30	58

Table 13. Les gradients d'énergie

La figure 44 suivante montre les directions des gradients  $G$  ( $G_x$ ,  $G_y$ ) dans chaque capteur de l'image en niveau de gris :

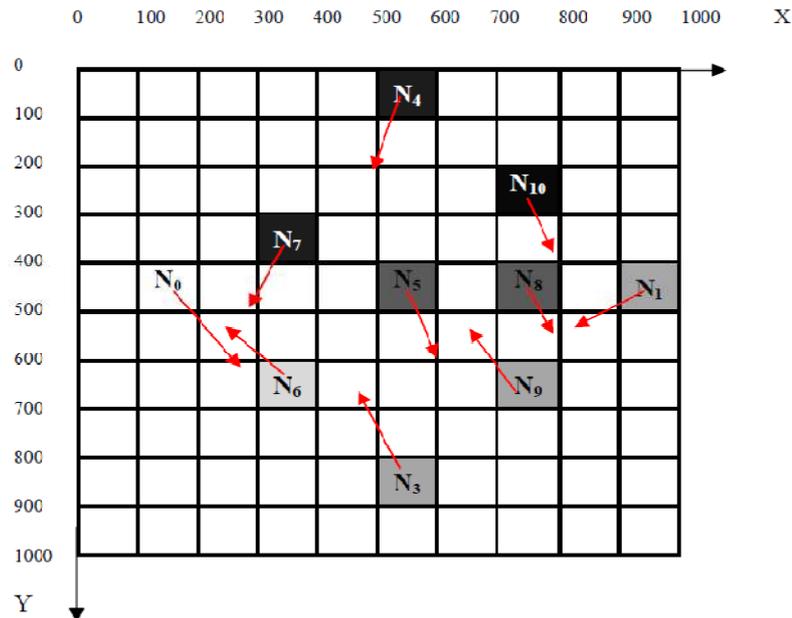


Figure 44 Le réseau en niveau de gris avec les directions des gradients

Les résultats des simulations sont obtenus à partir d'un ensemble de dix simulations avec un temps de simulation total de 1000s.

Nous allons nous concentrer sur les deux critères : le seuil d'énergie (ET) et le temps qu'un nœud reste en mode veille ( $T_{\text{sleep}}$ ).

La figure 45 représente le nombre de paquets envoyés avec des seuils d'énergie respectivement de : 0 mA, 5 mA, 10 mA et 15 mA et un  $T_{\text{sleep}}$  de 0.1s :

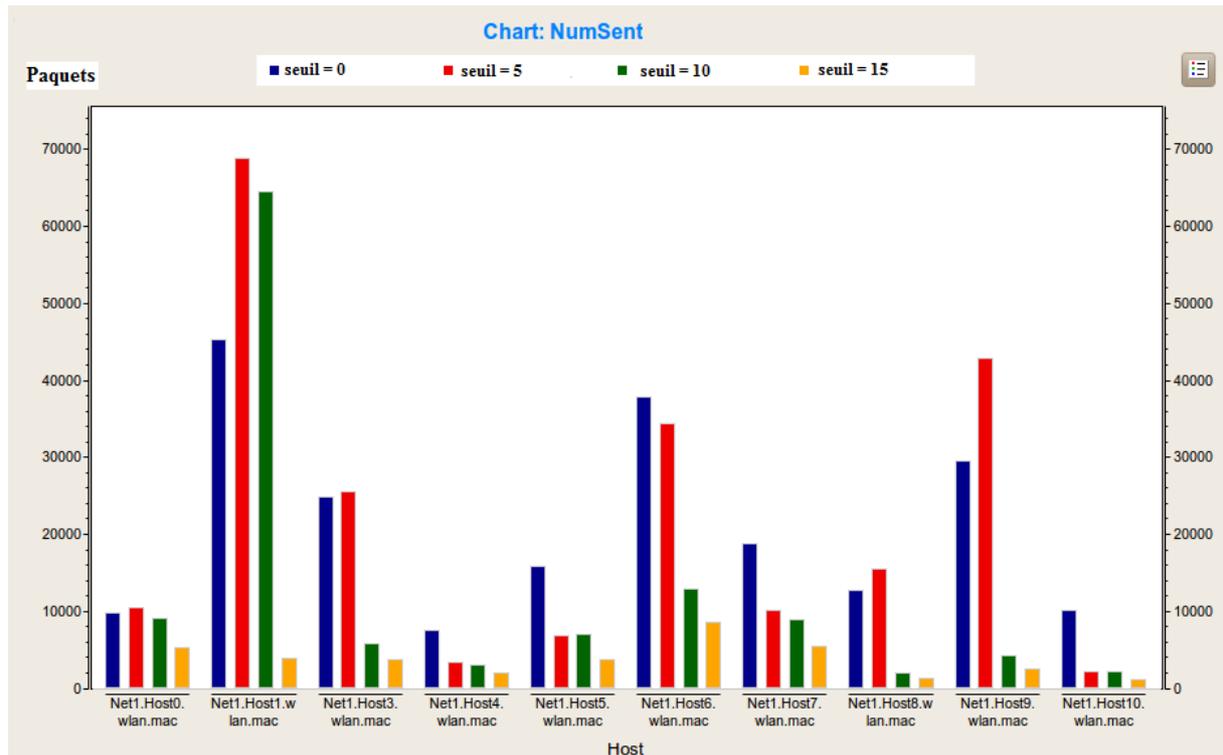


Figure 45 Le nombre de paquets envoyés avec un  $T_{sleep}$  de 0.1s

Dans cette figure nous constatons que les paquets sont envoyés principalement par les nœuds  $N_1$ ,  $N_3$ ,  $N_6$  et  $N_9$  qui ont les capacités énergétiques supérieures aux autres nœuds. Autrement dit, si on regarde la topologie de notre réseau dans la figure 42 on constate que la route va passer par les nœuds passerelles  $N_3$ ,  $N_6$  et  $N_9$  de la source  $N_1$  vers la destination  $N_0$ . Le nœud  $N_0$  a une grande capacité énergétique mais il a envoyé peu de paquets (des paquets de contrôle seulement) et c'est normal parce qu'il est le nœud de destination. Nous constatons également que si le seuil d'énergie est supérieur ou égal à 15mA le nombre de paquets émis par tous les nœuds tend vers zéro ce qui signifie que la communication est alors interrompue. Le nombre de paquets envoyés confirme que c'est  $N_1$  qui diffuse les paquets UDP.

La figure 46 suivante représente le nombre de paquets envoyés avec un  $T_{sleep}$  égal à 5 :

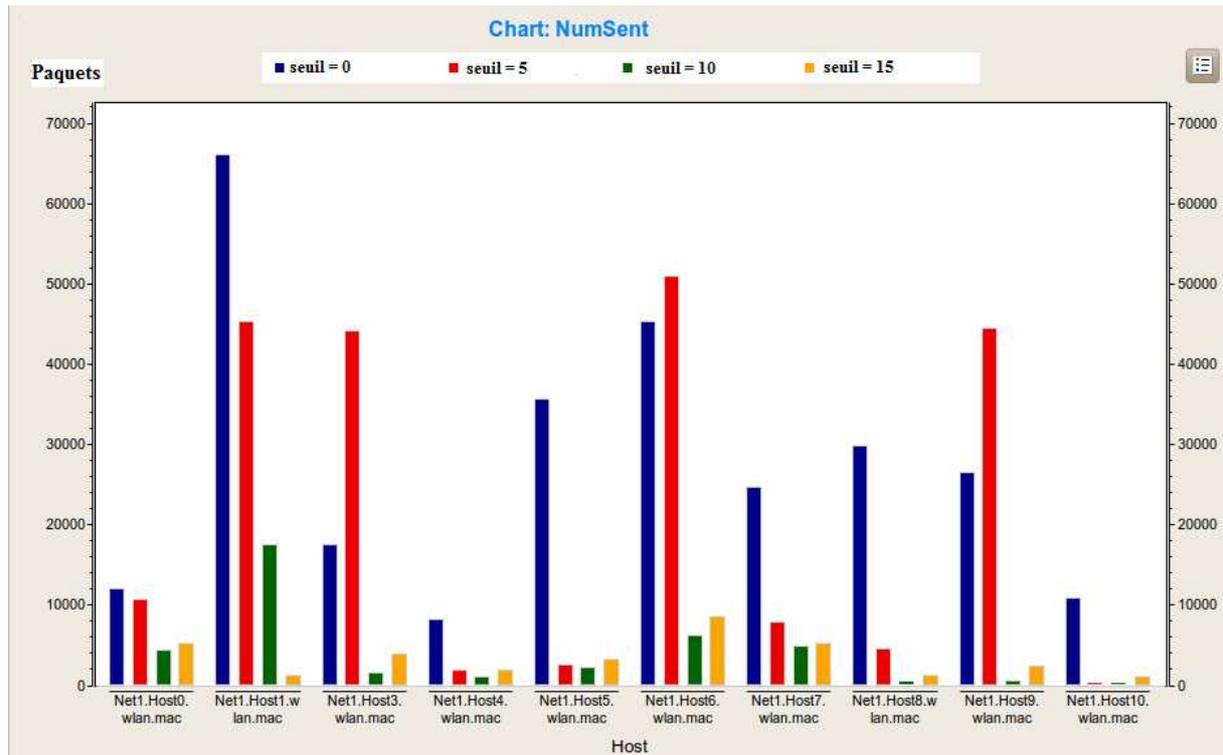


Figure 46 Le nombre de paquets envoyés avec un  $T_{\text{sleep}}$  de 5s

On remarque également dans cette figure que les paquets sont envoyés essentiellement pas les nœuds ayant la plus grande capacité énergétique. En revanche, le nombre de paquets diminue brusquement lorsque le seuil d'énergie est égal ou supérieur à 10mA comme le montre également la figure 47 avec un  $T_{\text{sleep}}$  égale à 10 et un seuil d'énergie entre 5mA et 15mA parce que le seuil est assez élevé et le temps total de simulation est de 1000s seulement.

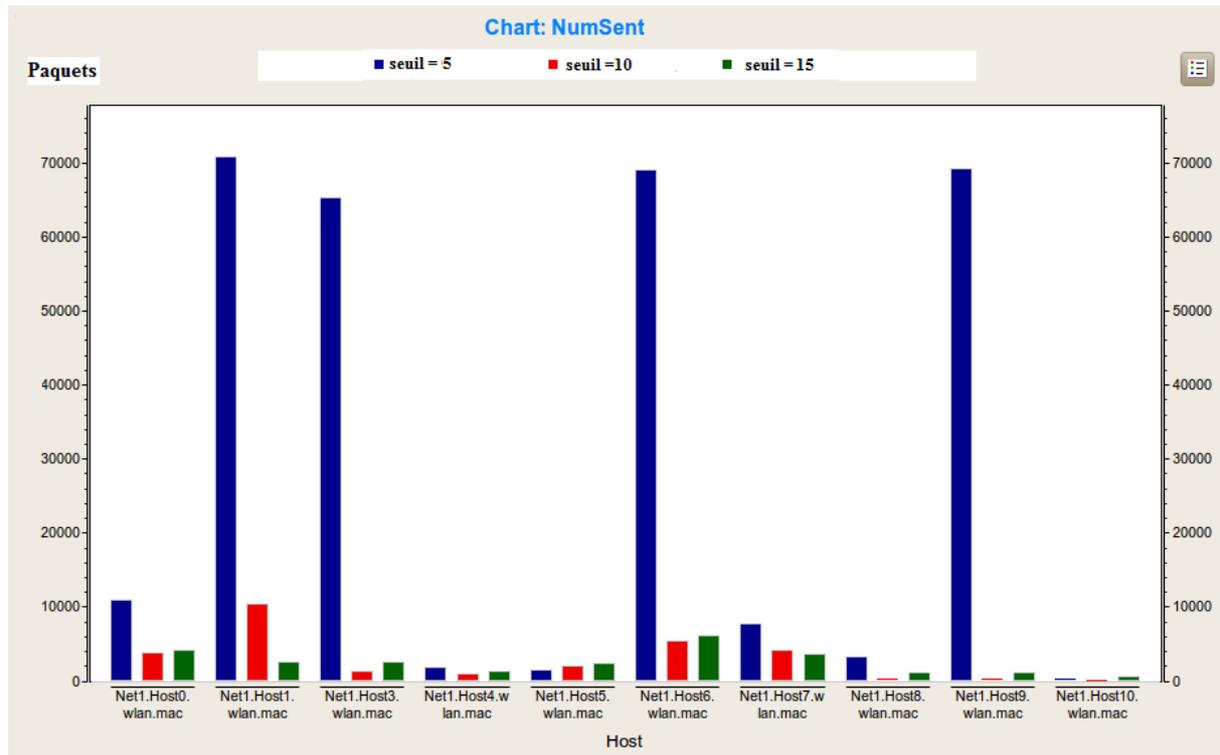


Figure 47 Le nombre de paquets envoyés avec un  $T_{sleep}$  de 10

Dans les trois figures précédentes nous avons remarqué que les paquets sont transmis par les nœuds ayant le plus d'énergie et cela signifie l'algorithme fonctionne. Par contre si nous augmentons le seuil d'énergie au-delà de 5mA avec un  $T_{sleep}$  entre 5s et 10s les nœuds arrêtent de transmettre les paquets à cause des valeurs assez élevées de  $T_{sleep}$  et d'ET.

Dans les trois figures suivantes avec leurs tables associées nous analysons l'énergie consommée par les capteurs avec les mêmes seuils d'énergie et les mêmes périodes en mode veille avec un temps total de simulation de 1000s.

La figure 48 avec la table 14 montrent l'énergie consommée avec un  $T_{sleep}$  de 0,1s :

Nœuds		N <sub>0</sub>	N <sub>1</sub>	N <sub>3</sub>	N <sub>4</sub>	N <sub>5</sub>	N <sub>6</sub>	N <sub>7</sub>	N <sub>8</sub>	N <sub>9</sub>	N <sub>10</sub>
Energie initiale		50	30	30	2	10	40	2	10	30	1
Energie consommée Tsleap= 0,1	Seuil = 0	2.9	8	3	1.7	8.4	4.2	2	10	9.7	1
	Seuil = 5	2.8	5.7	3	1.7	5.3	3.8	2	6.5	6.4	1
	Seuil = 10	2.5	3	2.5	1.6	5.5	3.7	2	5.3	4.2	1
	Seuil = 15	2.5	2.3	2.5	1.6	5.2	3.6	2	5.2	4.3	1

Table 14. L'énergie consommée avec un Tsleep de 0,1s

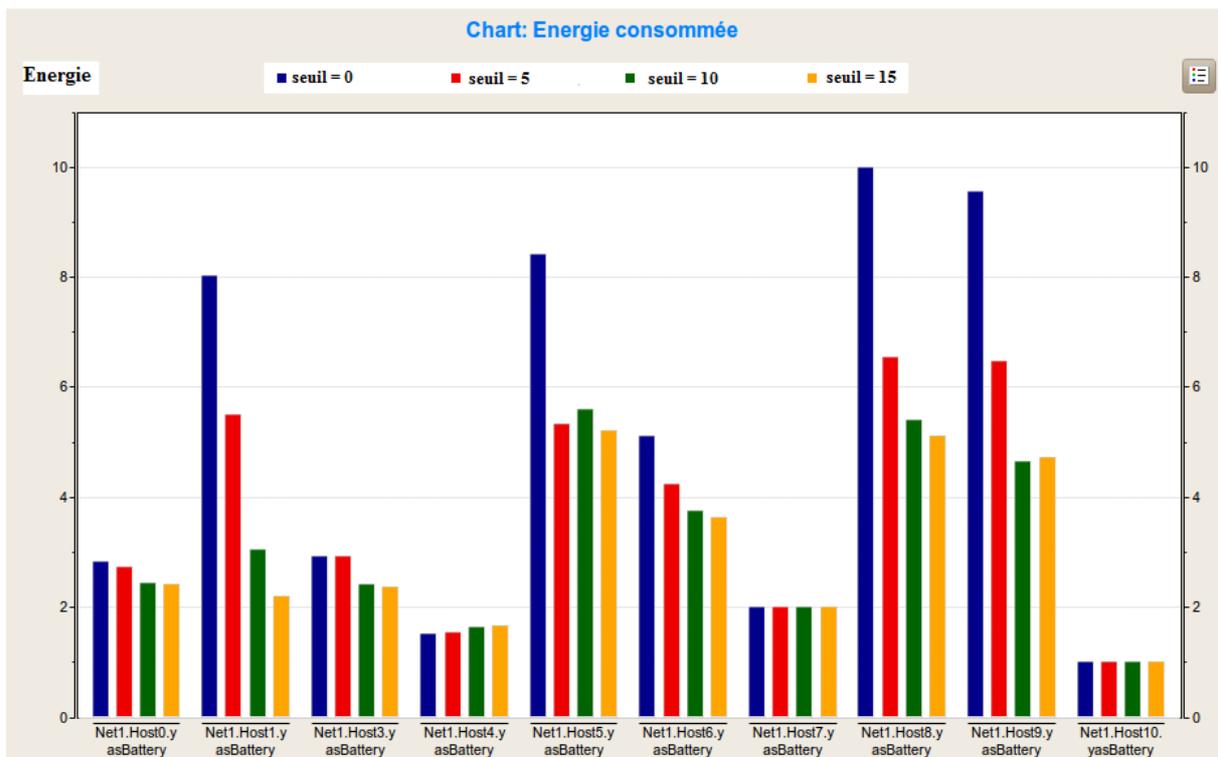


Figure 48 L'énergie consommée avec un Tsleep de 0,1s

Avec un  $T_{\text{sleep}}$  égal à 0,1s et un seuil d'énergie de 0mA, on constate que les nœuds N<sub>4</sub>, N<sub>5</sub>, N<sub>7</sub>, N<sub>8</sub> et N<sub>10</sub> qui sont faibles en énergie ont utilisé respectivement 85%, 84%, 100%, 100% et 100% de leurs capacités énergétiques. Par contre, si nous augmentons le seuil d'énergie jusqu'au 5mA nous les nœuds N<sub>5</sub> et N<sub>8</sub> ont consommés respectivement 53% et 65% seulement de leurs capacités totales. Cela signifie, nous avons économisé respectivement 32% et 35% d'énergie pour les nœuds N<sub>5</sub> et N<sub>8</sub> avec la combinaison ( $T_{\text{sleep}} = 0,1\text{s}$ ,  $ET = 5\text{mA}$ ). On remarque dans la même table qu'on a économisé également avec la même combinaison 11%

d'énergie pour le nœud  $N_9$  et 0.2% d'énergie pour  $N_6$ . Si nous continuons à augmenter le seuil jusqu'au 10mA ou 15mA nous économisons un peu d'énergie mais l'énergie consommée ne change pas dans tous les nœuds qui signifie qu'il n'y a plus de paquets qui passent dans le réseau.

Si nous augmentons le  $T_{sleep}$  jusqu'à 5s comme le montre la table 15 et la figure 49 :

Nœuds		$N_0$	$N_1$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	$N_{10}$
Energie initiale		50	30	30	2	10	40	2	10	30	1
Energie consommée $T_{sleep}=5$	Seuil = 0	3	6.5	2.7	1.6	4.9	3.2	2	6.7	6.2	1
	Seuil = 5	2.8	7.7	3.1	1.6	5.4	4	2	8.4	8.1	1
	Seuil = 10	2.4	2.4	2.4	1.9	2.5	2.5	2	2.3	2.3	1
	Seuil = 15	2.3	2.1	2.4	1.6	5	3.5	2	4.9	4.9	1

Table 15. L'énergie consommée avec un  $T_{sleep}$  de 5s

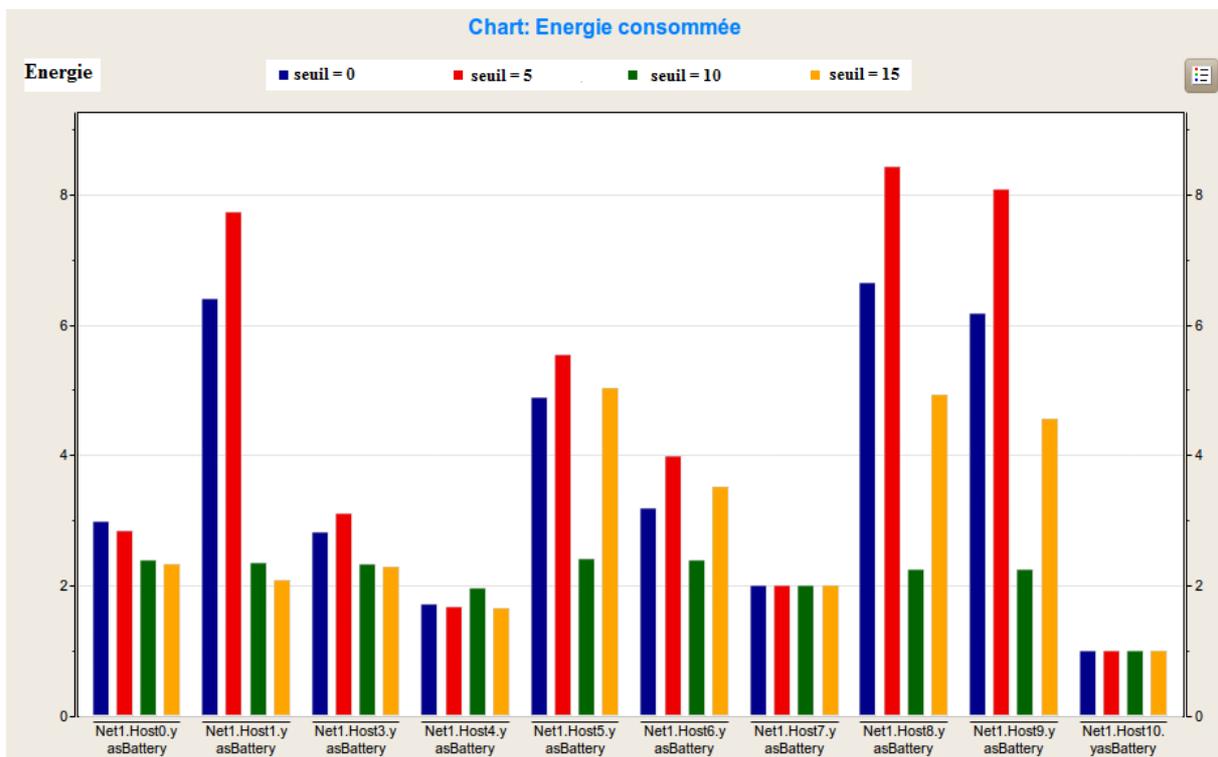


Figure 49 L'énergie consommée avec un  $T_{sleep}$  de 5s

Avec un seuil d'énergie de 0mA, les nœuds qui sont faibles en énergie  $N_4$ ,  $N_5$ ,  $N_7$ ,  $N_8$  et  $N_{10}$  ont utilisé respectivement 80%, 49%, 100%, 67% et 100% de leurs capacités énergétiques

et si le seuil d'énergie est égal à 5mA alors la consommation d'énergie augmente dans tous les nœuds et les nœuds  $N_5$  et  $N_8$  consomment respectivement 54% et 84% de leurs capacités. Donc, nous économisons avec un seuil de 0mA 5% d'énergie en  $N_5$  et 17% d'énergie en  $N_8$ . Si le seuil d'énergie est égal ou supérieur à 10 la consommation d'énergie ne change pas et les nœuds arrêtent de transmettre les paquets. Par conséquent, si le  $T_{sleep}$  est égal à 5s on obtient les meilleurs résultats lorsque le seuil d'énergie est égal à 0.

La table 16 et la figure 50 montrent les résultats de nos simulations avec un  $T_{sleep}$  est égal à 10s :

Nœuds		$N_0$	$N_1$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	$N_{10}$
Energie initiale		50	30	30	2	10	40	2	10	30	1
Energie consommée	Seuil = 5	3	5.4	3.5	1.8	3.8	3.8	2	5.6	5.9	1
	Seuil = 10	2.4	2.4	2.4	1.9	2.5	2.4	2	2.3	2.4	1
	Seuil = 15	1.8	1.6	1.8	1.2	4.6	3	1.6	4.5	4.2	1

Table 16. L'énergie consommée avec un  $T_{sleep}$  de 10s

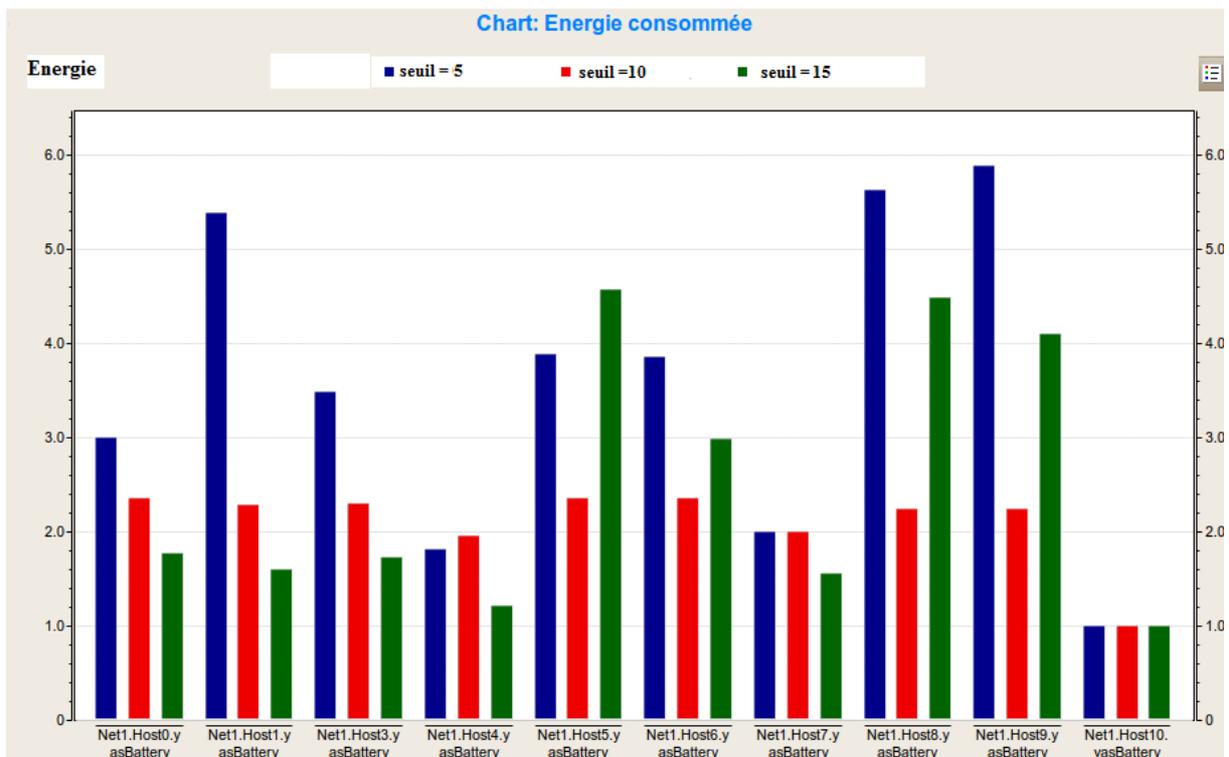


Figure 50 L'énergie consommée avec un  $T_{sleep}$  de 5s

Dans la figure précédente nous pouvons constater qu'avec un  $T_{\text{sleep}}$  de 10 et un seuil d'énergie supérieur à 5mA l'énergie consommée par tous les nœuds ne change pas comme le montre également la figure 6 où le nombre de paquets envoyés avec un  $T_{\text{sleep}}$  de 10 diminue brusquement Lorsque le seuil d'énergie est supérieur à 10.

Par conséquent, nous avons obtenu de bons résultats avec les combinaisons :

- 1-  $T_{\text{sleep}} = 0,1\text{s}$ ,  $ET = 5\text{mA}$
- 2-  $T_{\text{sleep}} = 5\text{s}$ ,  $ET = 0\text{mA}$
- 3-  $T_{\text{sleep}} = 10\text{s}$ ,  $ET = 5\text{mA}$

Mais les meilleurs ont été obtenus avec  $T_{\text{sleep}} = 0,1\text{s}$  et  $ET = 5\text{mA}$  car nous avons prolongé la durée de vie des nœuds  $N_5$  et  $N_8$  en économisant respectivement 32% et 35% de leurs capacité énergétique.

#### 4.4.6 Conclusion de l'algorithme hybride

Dans cet algorithme, tous les calculs sont faits par le CH qui a la puissance du processeur, les ressources mémoire et la capacité d'énergie plus importante. De plus, afin d'économiser l'énergie, les nœuds qui ne participent pas au routage sont instantanément mis en veille pendant une période déterminée en attendant leur réveil pour participer au routage ou se mettre à nouveau en veille.

Nous avons développé certains composants du simulateur OMNET ++ pour qu'il s'adapte à notre algorithme.

Dans un réseau de quatre nœuds nous avons conservé 30% de l'énergie de  $N_4$ . De plus, avec un réseau de dix nœuds, nous avons prolongé la durée de vie des nœuds  $N_5$  et  $N_8$  en économisant respectivement 32% et 35% de leurs capacités énergétiques.

### 4.5 Conclusion

Dans ce chapitre, nous avons exploité certaines techniques de traitement d'image pour proposer de nouvelles approches dans le monde de réseaux de capteurs dans fil. Nous avons traité en particulier le problème de la consommation d'énergie par trois algorithmes dont le

premier est basé sur l'algorithme de Sobel afin de trouver la direction du gradient qui indique la route à suivre pour faire passer les paquets de la source vers la destination tout en conservant l'énergie. Ensuite nous avons exploité une autre technique de traitement d'image pour créer un deuxième algorithme qui est basé sur le produit de convolution de deux matrices dont la première est la matrice représentant les valeurs énergétiques de huit secteurs autour d'un nœud central. La deuxième matrice est le filtre moyen qui calcule une valeur d'énergie moyenne d'une région. La dernière partie de ce chapitre consiste à réaliser un algorithme hybride ayant pour objectif de trouver le chemin le plus économe en énergie.

Les résultats montrent la bonne performance de l'algorithme hybride par rapport aux deux algorithmes précédents qui sont basés sur le filtre moyen et le filtre de Sobel car il intègre les deux filtres (Sobel et moyen). De plus, avec l'algorithme hybride, nous avons prolongé la durée de vie d'un nœud en économisant 30% de son énergie par rapport au 7% obtenu avec l'algorithme basé sur le filtre moyen.

## **Chapitre 5**

### **Conclusion et Perspectives**

## **4.1 Conclusion**

L'utilisation des batteries par les capteurs est une contrainte critique dans les réseaux de capteurs. La deuxième contrainte est environnementale : les capteurs sont parfois déployés sans surveillance et en grand nombre, de sorte qu'il est difficile de changer ou de recharger leurs batteries. Pour cela, les algorithmes et les protocoles de communication pour les réseaux de capteurs doivent minimiser la consommation d'énergie. Mais le taux de consommation d'énergie des capteurs reste très variable selon les protocoles utilisés.

Dans cette thèse, nous nous sommes intéressés au problème de l'énergie et du routage dans les réseaux de capteurs sans fil. Nous avons proposé des solutions permettant d'offrir une meilleure prise en compte des ressources énergétiques du réseau.

Nous avons présenté un état de l'art sur les réseaux de capteurs et leurs principaux protocoles de routage et nous avons présenté les protocoles basés sur plusieurs approches : le clustering, la position géographique des nœuds, la couche MAC, des mécanismes pour éviter les trous, etc.... Parmi ces protocoles, nous nous sommes intéressés plus précisément à ceux qui traitent du problème de la consommation d'énergie.

Dans le troisième chapitre nous avons fait une analogie entre le traitement d'images et les réseaux de capteurs dans la quelle on considère le réseau comme une image en niveaux de gris. Les capteurs sont les pixels et la capacité énergétique représente l'intensité lumineuse. Les zones sombres de l'image sont pauvres en énergie et les zones claires sont les capteurs ayant des batteries pleines.

Après cette représentation d'un réseau de capteurs, il nous restait à appliquer les techniques de traitement d'images afin de proposer un nouveau protocole de routage efficace en terme de conservation d'énergie. Les routes sélectionnées doivent passer par les régions claires de l'image. Autrement dit, elles doivent passer par les capteurs qui ont plus d'énergie. Pour déterminer la répartition de l'énergie dans le voisinage d'un nœud, nous avons coupé ce voisinage en huit secteurs autour de ce nœud. La valeur énergétique de chacun de ces huit secteurs fait une partie d'une matrice centrée sur la valeur énergétique du nœud central. Cette matrice, qu'on a appelée matrice d'énergie, est utilisée dans le produit de convolution avec

une matrice appelée le noyau. Le produit de convolution, représenté par la multiplication de deux précédentes matrices, est à la base de nos trois algorithmes suivants :

- 1- L'utilisation de l'algorithme de Sobel pour créer un routage efficace en terme de conservation d'énergie dans les réseaux de capteurs. Nous avons exploité l'idée de gradient, utilisé par Sobel, pour détecter la variation d'énergie dans le réseau de capteurs en employant une fonction de deux variables pour l'axe horizontal (X) et l'axe vertical (Y). La route est choisie parmi celles données par AODV en privilégiant celle qui a la direction du meilleur gradient d'énergie du réseau.
- 2- L'utilisation du produit de convolution pour créer un routage efficace en terme de conservation d'énergie dans les réseaux de capteurs. Cet algorithme est basé sur le clustering et le filtre moyen qui donne la moyenne de l'énergie autour du nœud central.
- 3- L'algorithme de routage basé sur les deux filtres (moyen et Sobel). Nous avons utilisé cet algorithme hybride pour trouver la route la plus économe en énergie en deux étapes : la direction du meilleur gradient d'énergie du réseau et la mise en sommeil de nœuds pauvres en énergie.

Enfin, ces études ont été validées à l'aide du simulateur de réseaux OMNeT++ que nous avons modifié pour qu'il s'adapte à nos algorithmes. Les résultats des simulations ont bien illustré, pour les trois algorithmes, que les protocoles proposés améliorent sensiblement les performances du routage et garantissent un prolongement de la durée de vie du réseau.

## **4.2 Perspectives**

Les réseaux de capteurs constituent un domaine de recherche très vaste. Ils ont de nombreuses perspectives d'application dans des domaines très variés. Il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans des conditions réelles.

Nous proposons de développer des algorithmes en utilisant d'autres filtres de convolution ou d'autres techniques de traitement d'images pour des projets dans les réseaux de capteurs sans fil.

Nous proposons également de développer notre approche en remplaçant le protocole de routage AODV qui n'assure pas l'utilisation de la meilleure route existant entre la source et la destination.

Nous proposons aussi d'utiliser uniquement des méthodes de traitement d'image pour construire des routes à travers le réseau. Dans cette perspective, la détection de contour dans l'image d'énergie permettra de construire des routes évitant les trous au niveau du réseau.

Afin d'être adaptés à un environnement réel, nos algorithmes peuvent être toujours améliorés en introduisant la mobilité des nœuds et en calculant le coût de sélection et la maintenance des chemins.

Il serait aussi intéressant également de proposer un nouvel algorithme de routage spécial pour déterminer et dessiner les frontières d'un trou dans un réseau de capteurs. Ainsi un autre algorithme pour éviter le bouclage.

Enfin, les performances de nos algorithmes ont été prouvées à l'aide du simulateur OMNeT++ et il serait préférable de réaliser ces études dans des conditions réelles de manière à comparer les performances réelles par rapport à celles effectuées par des simulations.

## Table des figures

1. Les composants d'un nœud capteur
2. Exemple de réseaux de capteurs
3. Les techniques de conservation d'énergie
4. Les deux requêtes RREQ et RREP en AODV
5. Les principaux protocoles de routages dans les RCSF
6. Topologie hiérarchique
7. Le routage data-centric
8. Image en niveau de gris
9. Un gradient d'une courbe d'isosurface
10. La matrice d'image et un noyau
11. Un exemple de filtre passe-bas
12. Un exemple de filtre moyen
13. Un exemple de filtre gradient
14. Un exemple de filtre Sobel
15. Les huit voisins d'un pixel  $P_{x,y}$
16. Les secteurs d'énergie
17. La matrice noyau
18. La direction du gradient sur l'axe x
19. La direction du gradient sur l'axe y
20. La direction de routage
21. Les secteurs d'énergie
22. La direction du gradient d'énergie
23. La fonction de probabilité pour le gradient d'énergie
24. Le gradient d'énergie avec  $Y \uparrow$

25. Les secteurs d'énergie autour de N0
26. Le routage par la direction du gradient
27. La topologie du réseau de notre algorithme par Sobel
28. La topologie du réseau de notre algorithme par filtre moyen
29. Le nombre de paquets envoyés avec un délai de 0.5s
30. L'énergie restante avec un délai de 0.5s
31. Les étapes de l'algorithme au niveau de Ni
32. Les étapes de l'algorithme au niveau de CH
33. Le nombre de paquets envoyés avec un seuil d'énergie de 2
34. Le nombre de paquets envoyés avec un seuil d'énergie de 3
35. L'énergie restante avec un seuil d'énergie de 2
36. L'énergie restante avec un seuil d'énergie de 3
37. L'énergie restante avec un seuil d'énergie de 0
38. La consommation d'énergie de N3 avec un seuil d'énergie de 2
39. La consommation d'énergie de N3 avec un seuil d'énergie de 3
40. La consommation d'énergie de N4 avec un seuil d'énergie de 2
41. La consommation d'énergie de N4 avec un seuil d'énergie de 3
42. La topologie d'un réseau de dix nœuds
43. La représentation du réseau en niveau de gris
44. Le réseau en niveau de gris avec les directions des gradients
45. Le nombre de paquets envoyés avec un Tsleep de 0.1s
46. Le nombre de paquets envoyés avec un Tsleep de 5s
47. Le nombre de paquets envoyés avec un Tsleep de 10
48. L'énergie consommée avec un Tsleep de 0,1s
49. L'énergie consommée avec un Tsleep de 5s
50. L'énergie consommée avec un Tsleep de 5s

## Liste des tables

1. Classification et comparaison des protocoles de routages dans les réseaux de capteurs
2. Le nombre de paquets envoyés avec un SEID de 0.1 et de 0.5s
3. Le nombre de paquets envoyés avec un SEID de 1 et de 1.5s
4. Le reste d'énergie avec un SEID de 0.1 et de 0.5s
5. Le reste d'énergie avec un SEID de 1 et de 1.5s
6. Le nombre de paquets envoyés avec un ET de 0 et de 1
7. Le nombre de paquets envoyés avec un ET de 2 et de 3
8. Le nombre de paquets envoyés avec un ET de 4 et de 5
9. Le reste d'énergie avec un ET de 0 et de 1
10. Le reste d'énergie avec un ET de 2 et de 3
11. Le reste d'énergie avec un ET de 4 et de 5
12. Les positions et les capacités énergétiques des capteurs
13. Les gradients d'énergie
14. L'énergie consommée avec un  $T_{\text{sleep}}$  de 0,1s
15. L'énergie consommée avec un  $T_{\text{sleep}}$  de 5s
16. L'énergie consommée avec un  $T_{\text{sleep}}$  de 10s



## Liste des acronymes

ADC : Analog-Digital Converter

AODV: Ad hoc On-demand Distance Vector

APTEEN: Adaptive Threshold sensitive Energy Efficient sensor Network protocol

ACQUIRE: ACtive QUery forwarding In sensoR nEtworks

BC: Battery Capacity

CADR: Constrained Anisotropic Diffusion Routing

CH: Cluster-Head

CPU: Central Processing Unit

CSIP: Collaborative Signal and Information Processing

CSMA: Carrier Sense Multiple Access

CSMA/CA: Carrier Sense Multiple Access with Collision Avoidance

CTS: Clear To Send

DVS: Dynamic Voltage Scaling

DSDV: Destination Sequence Distance Vector

EARCBSN: Energy-Aware Routing in Cluster-Based Sensor Networks

EARLEAHSN: Energy Aware Routing for Low Energy Ad Hoc Sensor Networks

ERP: Energy Routing Parameter

ET: Energy Threshold

FSR: Fisheye State Routing

GAF: Geographic Adaptive Fidelity

GEAR: Geographic and Energy Aware Routing

GBR: Gradient-Based Routing

GPS: Global Position System

HTTP: Hyper Text Transfer Protocol

Idle: Ecoute non actif

IEEE: Institute of Electrical Electronic Engineers

IP: Internet Protocol

LAN: Local Area Network

LEACH: Low Energy Adaptive Clustering Hierarchical

MAC: Medium Access Control

MANET: Mobile Ad hoc Networks

MECN: Minimum Energy Communication Network

NS: Network Simulator

OLSR: Optimized Link State Routing Protocol

PEGASIS: Power-Efficient GATHERing in Sensor Information Systems

QoS: Quality of Service

R: Range

RCSF: Réseau de Capteurs Sans Fil

RGB: Red, Green, Blue

RREP: Route Reply

RREQ: Route Request

RTS: Request to Send

Rx: Reception

SAR: Sequential Assignment Routing

SEID: Sending Energy Information Delay

S-MAC: Sensor MAC

SN: Sequence Number

SPIN: Sensor Protocols for Information via Negotiation

TCP: Transmission Control Protocol

TEEN: Threshold sensitive Energy Efficient sensor Network protocol

Tx: Transmission

UDP: User Datagram Protocol

WLAN: Wireless LAN

WSNs: Wireless Sensor Networks

ZB: ZigBee

## REFERENCES

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci , “A Survey on Sensor Networks”. IEEE Communications Magazine, August 2002.
- [2] B. Jung and G. Sukhatme, “Multi-target tracking using a mobile sensor network”. In Proceedings of the IEEE International Conference on Robotics and Automation, May 2002.
- [3] C. Chong and S. Kumar, “Sensor networks: Evolution, opportunities, and challenges” Proceedings of IEEE, August 2003.
- [4] E. M. Petriu, N. D. Georganas, D. C. Petriu, D. Makrakis, and V. Z. Groza, “Sensor-based information appliances”. IEEE Information and Measurement Magazine, December 2000.
- [5] B. Sibbald, “Use computerized systems to cut adverse drug events”. Canadian Medical Association Journal, June 2001.
- [6] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, “Impact of Radio Irregularity on Wireless Sensor Networks”. MobiSys 2004.
- [7] ALERT Systems. <http://www.alertsystems.org/>.
- [8] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron, “Monitoring behavior in home using a smart fall sensor”. IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology, October 2000.
- [9] J. Lester Hill, “System Architecture for Wireless Sensor Networks”. University of California, Berkeley, 2003.
- [10] V. Handziski, A. Kopke, H. Karl, and A. Wolisz, “A common wireless sensor network architecture”. Technische Universität Berlin, July 2003, pp.10-17.
- [11] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, “Fault Tolerance in Wireless Ad hoc Sensor Networks”. Proceedings of IEEE Sensors 2002, June 2002.

- [12] L. Paradis and Q. Han, “A Survey of Fault Management in Wireless Sensor Networks”. Plenum Press New York, NY, USA, 2007.
- [13] F. Nekoogar, F. Dowla, and A. Spiridon, “Self organization of wireless sensor networks using ultra-wideband radios”. Atlanta, GA, United States, September 2004.
- [14] M. Beigl, C. Decker, A. Krohn, T. Riedel, and T. Zimmer, “Low Cost Sensor Networks at Scale”. 7th International Conference on Ubiquitous Computing Demonstration Proceedings(ubicomp’05), 2005.
- [15] C.Y. Chong and S.P. Kumar, “Sensor networks: Evolution, opportunities, and challenges”. Proceedings of the IEEE, vol. 91, n.8, 2003, pp. 1247-1256.
- [16] T. Zhao, W. D. Cai, and Y. J. Li, “A Sensor Network Topology Inference Algorithm, computational Intelligence and Security”. 2007 International Conference on. January 2008.
- [17] V. Handziski, J. Polastre, J. H. Hauer, C. Sharp, A. Wolisz, and D. Cullery, “Flexible Hardware Abstraction for Wireless Sensor Networks”. In Proceedings of the Second European Workshop on Wireless Sensor Networks (EWSN '05), February 2005.
- [18] Jaap C. Haartsen, “The Bluetooth radio system”. IEEE Personal Communications Magazine, February 2000, pp. 28-36.
- [19] Practel, Inc. ZigBee, “Technology for Wireless Sensor Networks”. April 2006.
- [20] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, “Wireless sensor networks for habitat monitoring”. Atlanta, Georgia, USA, 2002.
- [21] I. Teixeira, J. F. de Rezende, A. de Castro, and A. C. P. Pedroza, “Wireless Sensor Network: Improving the Network Energy Consumption”. in XXI Symposium Brazilian Telecommunications, SBT’04, Belem, Brazil, September 2004.
- [22] E. Souto, R. Gomes, D. Sadok and J. Kelner, “Sampling Energy Consumption in Wireless Sensor Networks”. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06), June 2006.
- [23] M. Ali and S. K. Ravula, “Real-time support and energy efficiency in wireless sensor networks”. Technical report, IDE0805, January 2008.

- [24] K. Akkaya, and M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks”. *Journal of Ad Hoc Networks*, Vol. 3, No. 3, May 2005, pp. 325-349.
- [25] , J.N. Al-Karaki and A.E. Kamal, “Routing techniques in wireless sensor networks: a survey. *Wireless Communications*”. IEEE, Decembre 2004.
- [26] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless sensor networks ”. In the Proceeding of the Hawaii International Conference System Sciences, Hawaii, January 2000.
- [27] S. Lindsey and C. Raghavendra, “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”. *Proceedings of the IEEE Aerospace Conference*, vol. 3, Big Sky, MT, USA, March 2002, pp. 1125-1130.
- [28] A. Manjeshwar and D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks”. *1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.
- [29] N. V. Subramanian, “Survey on Energy-Aware Routing and Routing Protocols for Sensor Networks”. *Technical Report, Computer Science, University of North Carolina, Charlotte*. 2004.
- [30] B. Karp, H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”. In *Mobile Computing and Networking (MOBICOM'2000)*, 2000, pp. 243–254.
- [31] W. Ye, J. Heidemann, and D. Estrin. “An Energy-Efficient MAC Protocol for Wireless Sensor Networks”. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*, vol.3, New York, NY, USA, June 2002, pp.1567-1576.
- [32] V. Rodoplu and T. H. Ming, “Minimum energy mobile wireless networks”. *IEEE Journal of Selected Areas in Communications*, Vol. 17, No. 8, 1999, pp. 1333-1344.

- [33] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad hoc Routing". In Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'01), Rome, Italy, July 2001, pp.70-84.
- [34] L. Doherty, K. S. J. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks". In Proceedings of the IEEE INFOCOM, vol.3, Alaska, 2001, pp.1655-1663.
- [35] Y. Yu, D. Estrin, and R. Govindan, "Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks," UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May 2001.
- [36] S. Singh and C. Raghavendra, "PAMAS: Power aware multi-access protocol with signaling for ad hoc networks". ACM Computer Communication Review, July 1998, pp. 5-26.
- [37] H. Qi, P. T. Kuruganti and Y. Xu, "The Development of Localized Algorithms in Wireless Sensor Networks". Published on 202 SENSORS ISSN, 22 July, pp. 1424- 8220.
- [38] C. B. Abbas, R. González, N. Cardenas and L. J. G. Villalba, "A proposal of a wireless sensor network routing protocol". Springer Science and Business Media .Telecommunication Systems. March 2008, pp. 61–68.
- [39] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks". In the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, August 2000.
- [40] D. Estrin, R. Govindan, J. Heidemann, and Satish Kumar, "Next century challenges: Scalable Coordination in Sensor Networks". In the Proceedings of the 5th annual ACM/IEEE international conference on Mobile Computing and Networking (MobiCom'99), Seattle, WA, August 1999.

- [41] E. M. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks". *IEEE Personal Communications*, Vol. 6, No. 2, April 1999, pp. 46-55.
- [42] B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling Data Centric Routing in Wireless Sensor Networks". In the Proceedings of IEEE INFOCOM, New York, NY, June 2002.
- [43] S. Ziane and A. Mellouk, "A swarm intelligent scheme for routing in mobile ad networks". *Systems Communications, IEEE*, Aug 2005.
- [44] Paolo Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks". Hardcover, july 2005.
- [45] S. Kumar, D. Shepherd, and F. Zhao, "Collaborative signal and information processing in micro-sensor networks". *IEEE Signal Processing Magazine*, March 2002.
- [46] M. Kubisch, H. Karl, A. Wolisz, L. C. Zhong and J. Rabaey, "Distributed Algorithms for Transmission Power Control in Wireless Sensor Networks". In Proc. IEEE INFOCOM, 2003. pp.558-563.
- [47] N. Li and J. C. Hou, "BLMST: a decentralized, power-efficient broadcast algorithm for wireless sensor networks". Accepted for publication in ACM Baltzer Wireless Networks (WINET), 2005.
- [48] M. Younis, M. Youssef, and K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks". In Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002), (Forth Worth, TX), 2002.
- [49] R. Shah and J. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks". In the Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Orlando, FL, March 2002.

- [50] K. Zeng, K. Ren, W. Lou and P. J. Moran, “Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply”. Kluwer Academic Publishers Hingham, MA, USA, 2009.
- [51] K. Arisha, M. Youssef, M. Younis, “Energy-Aware TDMA-Based MAC for Sensor Networks”. IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), May 2002.
- [52] A. Cerpa and D. Estrin, “ASCENT: Adaptive Self-Configuring Sensor Networks Topologies”. In the Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), New York, NY, USA, June 2002.
- [53] C. Schurgers and M. B. Srivastava, “Energy efficient routing in wireless sensor networks”. In the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force, McLean, VA, 2001.
- [54] L. Qin and T. Kunz, “Survey on Mobile Ad Hoc Network Routing Protocols”. Carleton University, Systems and Computer Engineering, Technical Report SCE-04-14, August 2004.
- [55] M. I. Razzak, S. A. Hussain, A. A. Minhas and M. Sher, “Collaborative Image Compression in Wireless Sensor Networks”. International journal of computational cognition (<http://www.ijcc.us>), vol. 8, no. 1, march 2010.
- [56] L. Zhang , Y. Lu, L. Chen, and D. Dong, “Game Theoretical Algorithm for Coverage Optimization in Wireless Sensor Networks”. Proceedings of the World Congress on Engineering 2008 Vol I WCE 2008, London, U.K, July 2008.
- [57] S. Dhar, K. Roy, and R. Kannan, “Node Energy Based Clustering Protocol for Wireless Sensor Networks with Guaranteed Connectivity”. Springer Berlin / Heidelberg, November 2004, pp. 75-84.

- [58] J. H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks”. In *IEEE/ACM Transactions on Networking (TON)*, vol. 12, August 2004, pp. 609-619.
- [59] M. Kalantari and M. Shayman, “Energy Efficient Routing in Wireless Sensor Networks”. In *Proc. Conference on Information Sciences and Systems*, 2004.
- [60] Y. Tian, F. Yu, Y. Choi, S. Park, E. Lee, M. Jin, and S. H. Kim, “Energy efficient data dissemination protocol for detouring routing holes in wireless sensor networks”. *ICC 2008 - IEEE International Conference on Communications*, vol. 31, no. 1, May 2008, pp. 2322-2326.
- [61] D. Kim, J. J. Garcia-Luna-Aceves, K. Obraczka, J. C. Cano, and P. Manzoni, “Routing Mechanisms for Mobile Ad Hoc Networks Based on the Energy Drain Rate”. In *IEEE Trans. Mob. Comput*, 2003, pp. 161-173.
- [62] R. Lin, Z. Wang, and Y. Sun, “Energy efficient medium access control protocols for wireless sensor networks and its state-of-art”. In *Industrial Electronics, 2004 IEEE International Symposium*, May 2004, pp 669 – 674.
- [63] K. Sohrabi and G.J. Pottie, “Performance of a novel self-organization protocol for wireless ad hoc sensor networks”. In *Proceeding of the 50th IEEE Vehicular Technology Conference (VTC’99)*, vol.2, Amsterdam, September 1999, pp.1222-1226.
- [64] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo, “An implicit prioritized access protocol for wireless sensor networks“. *IEEE Real-Time System Symposium (RTSS’02)*, December 2002.
- [65] H. Wu, A. A. Abouzeid, “Energy efficient distributed JPEG2000 image compression in multihop wireless networks”. *ASWN 2004*. pp 152-160.
- [66] D. Gao, Y. Niu, X. Xu and H. Zhang, “A hybrid localization system in wireless sensor networks”. In *Networking, Sensing and Control, 2009. ICNSC '09. International Conference on*, March 2009, pp. 576 – 580.

- [67] R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui and M. Jordan, “Learning the Kernel Matrix with Semidefinite Programming”. *JMLR.org*, Volume 5, December 2004, pp27 - 72 .
- [68] M. Gilg, Y. Yousef, and P. Lorenz, “Using image processing algorithms for energy efficient routing algorithm in sensor networks”. *ADAPTIVE 2009*. Athens/Glyfada, Greece, November 2009, pp. 132-136.
- [69] Y. Yousef, M. Gilg, and P. Lorenz, “Using Matrix convolutions and Clustering for Energy Efficient Routing Algorithm in Sensor Networks”. *AICT 2010 :Sixth Advanced International Conference on Telecommunications*, Spain/Barcelona, May 2010, pp. 275-279.
- [70] O. R. Vincent and O. Folorunso, “A Descriptive Algorithm for Sobel Image Edge Detection”. *Proceedings of Informing Science & IT Education Conference (InSITE)*, 2009.
- [71] W. Drytkiewicz, S. Sroka, V. Handziski, A. Koepke, and H. Karl, “A mobility framework for OMNeT++”. In *Proceedings of the 3rd International OMNeT++ Workshop*, Budapest, Hungary, January 2003.
- [72] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks”. *Ad Hoc Networks*, Elsevier, Vol. 2, Issue 1, 2004, pp. 1-22.
- [73] Z. Qiang and Z. Hongbo, “An Optimized AODV Protocol in Mobile Ad Hoc Network”. In *IEEE Wireless Communications and Networking and Mobile Computing*, October 2008, pp. 1-4.
- [74] C. Lin, “AODV Routing Implementation for Scalable Wireless Ad Hoc Network Simulation (SWANS) ”. April 2004.
- [75] I. D. Chakeres and E. M. Belding-Royer, “AODV routing protocol implementation design”. In *IEEE Distributed Computing Systems Workshops*. 23-24 March 2004, pp. 698-703.

- [76] X. Jing and M. Lee, “Energy-Aware Algorithms for AODV in Ad Hoc Networks”. Department of Electrical Engineering. In Proceedings of Mobile Computing and Ubiquitous Networking (ICMU 2004), Yokosuka, Japan, January 2004.
- [77] M. Macedo, A. Grilo, and M. Nunes, “Distributed Latency-Energy Minimization and interference avoidance in TDMA Wireless Sensor Networks”. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 2009, pp. 569-582.
- [78] M. Tubaishat, J. Yin, B. Panja, and S. Madria, “A Secure Hierarchical Model for Sensor Network”. In IEEE, special issue: Special section on sensor network technology & sensor data management (Part II), March 2004. pp. 7 – 13.
- [79] J. Glaser, D. Weber, S. A. Madani, and S. Mahlke, “Power Aware Simulation Framework for Wireless Sensor Networks and Nodes”. In *EURASIP Journal on Embedded Systems*, Article ID 369178, May 2008.
- [80] OMNeT++ Community - [www.omnetpp.org](http://www.omnetpp.org). 2001-2009.
- [81] Nano WiReach™ “Miniature embedded secure WiFi module”. In Connect One Ltd. Publication Number 17-3520-11, September 2008.
- [82] A. Rowe, R. Mangharam, and R. Rajkumar, “RT-Link, A Global Time- Synchronized Link Protocol for Sensor Networks”. Elsevier *Ad hoc Networks*, Special Issue on Energy Efficient Design in Wireless Ad Hoc and Sensor Networks, 2007.
- [83] B. Fisher, Y. Chen, “Generalizations of Non-Commutative Neutrix Convolution Products of Functions”. *Georgian Math. J.* 4 (1997), No. 5, pp. 413-419.
- [84] M. Nunes, A. Grilo, M. Macedo, “Interference-free TDMA slot allocation in Wireless Sensor Networks”. In: Proceedings of the 32nd IEEE Conference on Local Computer Networks (IEEE LCN’2007), Dublin, Ireland, October 2007.
- [85] M. Zhang, Y. Lu, C. Gonh, Y. Feng, “Energy-Efficient Maximum Lifetime Algorithm in Wireless Sensor Networks”. 2008 International Conference on Intelligent Computation Technology and Automation (ICICTA), October 2008.

- [86] M. Wu and C. W. Chen, “Collaborative Image Coding and Transmission Over Wireless Sensor Networks”. *EURASIP Journal on Advances in Signal Processing* , Article ID 70481, Volume 2007.
- [87] A. Celisse and S. Robin, “Nonparametric density estimation by exact leave-p-out crossvalidation”. *Computational Statistics and Data Analysis* 52, 2008, pp.2350-2368.
- [88] D. Keren, M. Osadchy, and C. Gotsman, “A novel, fast method for image detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2001, pp.747-761.
- [89] S. Konishi, A. Yuille, J. Coughlin, and S. C. Zhu, “Statistical edge detection: Learning and evaluating edge cues”. *IEEE Transactions on PAMI*, 2003, pp. 57–74.
- [90] F. Campillo and V. Rossi, “Convolution Particle Filter for Parameter Estimation in General State-Space Models ” . *IEEE Transactions on Aerospace and Electronic Systems*, Septembre 2009, pp. 1063 – 1072.
- [91] J. J. Yin , Q. Lin, and J. Q. Zhang, “The Gaussian convolution filter and its application to navigation”. *9th International Conference, Signal Processing ICSP 2008*, October 2008, pp. 2829 – 2832.
- [92] D. Barash and D. Comaniciu, “A Common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift”. *Image Vis. Comp.* 2004, pp.73-81.
- [93] M. Mahmoudi and G. Sapiro, “Fast Image and Video Denoising via Nonlocal Means of Similar Neighborhoods”. *IEEE Signal Processing Letters*, vol. 12, No. 12, December 2005.
- [94] I. Vanhamel, I. Pratikakis, and H. Sahli, “Multiscale Gradient Watersheds of Color Images”. *IEEE Transactions on Image Processing*, vol. 12, No. 6, June 2003.
- [95] B. Krishnamachari, D. Estrin and S. Wicker, “Modelling Data-Centric Routing in Wireless Sensor Networks”. *USC Computer Engineering Technical Report CENG 02-14*, 2002.

- [96] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, “Energy-aware wireless microsensor networks”. *IEEE Signal Processing Magazine*, Vol. 19, No. 2, March 2002, pp.40-50.
- [97] R. Bellman, “On a Routing Problem, in *Quarterly of Applied Mathematics*”. 16(1), 1958, pp.87-90.
- [98] A. Munaretto, H. Badis, K. Al Agha, and G. Pujolle, “QoS-enhanced OLSR Protocol for Mobile Ad Hoc Networks”. In the Proceedings of The ANWIRE 1st International Workshop, Glasgow, Ecosse, April 2003, pp 171–183.
- [99] D. B. Johnson, D. A. Maltz, and J. Broch, “DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks”. In *Ad Hoc Networking*, edited by Charles E. Perkins, chapter 5, Addison-Wesley, 2001, pp. 139–172.
- [100] C. E. Perkins, E. M. Royer, S. R. Das, “Ad hoc on demand distance vector (aodv ) routing”. In IETF, Internet Draft, draft-ietf-manet-aodv-05.txt. [En ligne] 2000.
- [101] E. Arias-Castro and D.L. Donoho, “Does median filtering truly preserve edges better than linear filtering?”. *Annals of Statistics*, vol. 37, no. 3, 2009, pp. 1172-1206.
- [102] A. U. Shankar, C. Alaettinoglu, I. Matta, and K. Dussa-Zieger, “Performance comparison of routing protocols using mars: Distance-vector versus linkstate”. In *ACM SIGMETRICS Conference on Performance*, June 1992.
- [103] K. Yedavalli, B. Krishnamachari, “Enhancement of the IEEE 802.15.4 MAC Protocol for Scalable Data Collection in Dense Sensor Networks”. *Proc. WiOpt08*, Berlin, Germany, April 2008.
- [104] L. Saraswat, P. S. Yadav, and R. Rani, “Adaptability of IEEE 802.15.4(Zigbee) Protocol for Wireless Sensor network”. (*IJCSE*) *International Journal on Computer Science and Engineering* Vol. 02, No. 03, 2010, pp. 554-559.

## **Liste de mes publications**

- M. Gilg, Y. Yousef, P. Lorenz, "Using Image Processing Algorithms for Energy Efficient Routing Algorithm in Sensor Networks", First International Conference on Adaptive and Self-adaptive Systems and Applications, ADAPTIVE'09, November 15-20, 2009, Athens, Greece, pp. 132-136.
- Y. Yousef, M. Gilg, and P. Lorenz, "Using Matrix convolutions and Clustering for Energy Efficient Routing Algorithm in Sensor Networks", AICT 2010, Sixth Advanced International Conference on Telecommunications, AICT'10, May 9-15, 2010, Barcelona, Spain, pp. 275-279.
- Y. Yousef, M. Gilg, and P. Lorenz, "Using Convolution Filters for Energy Efficient Routing Algorithm in Sensor Networks". International Journal On Advances in Telecommunications, vol 3, no 1&2, 2010, ISSN: 1942-2601.
- Y. Yousef, "Utilisation de matrice de convolution dans le routage pour les réseaux de capteurs sans fil". Besançon, Réunion RGE, France, Février 2010.

# Résumé

## Protocoles de routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil

Avec l'émergence des nouvelles technologies, les communications sans fil n'ont cessé de croître afin de permettre aux utilisateurs un accès à l'information et aux services électroniques, et ceci indépendamment de leur position géographique. Les réseaux sans fil ont aussi trouvé leur place pour des applications spécifiques telles que les transmissions radio utilisées pour l'interconnexion de capteurs. Ce type de réseau peut être considéré comme un sous-ensemble des réseaux ad hoc. Des contraintes spécifiques s'appliquent alors aux utilisateurs de ces réseaux, telles que la difficulté d'accès pour la maintenance, les problèmes liés à la miniaturisation et au nombre élevé de capteurs.

L'objectif de cette thèse est d'étudier les contraintes énergétiques liées à l'utilisation des batteries à capacité limitée pour l'alimentation des capteurs. Ainsi dans notre travail, nous proposons des protocoles de routage pour acheminer l'information tout en contrôlant la consommation d'énergie.

Pour atteindre cet objectif, nous avons proposé de représenter les réseaux de capteurs à travers une image à échelle de gris : les zones claires correspondant aux zones riches en énergie, alors que les zones sombres représentent des régions avec une capacité énergétique faible. Des filtres issus du monde de traitement d'image sont alors appliqués à cette image représentant l'énergie. Ainsi, nous proposons des filtres de convolution de type Sobel ou de type filtre moyen pour nos algorithmes de routage et nous construisons une matrice énergétique pour chaque capteur. Cette matrice est alors utilisée avec le produit de convolution pour guider le routage. Les différents algorithmes proposés font ensuite l'objet de simulations avec le simulateur de réseaux OMNeT++.

**Mots-clés :** Réseau de capteurs sans fil, Routage, Energie, Traitement d'image, Matrices de convolution

# Abstract

## **Routing protocols for energy management in wireless sensor networks**

With the emergence of new technologies, wireless communications have been developed in order to allow users an access to information and to electronic services, independently of their geographical position. Wireless networks have also been developed for specific applications such as radio transmissions used for interconnection of sensors. This type of network can be considered as a subset of ad hoc networks. On other side, this implies specific constraints on users, such as the problem of the access for maintenance, the problems of miniaturization, and the large number of sensors.

The objective of this thesis is to focus on energy constraints related to the use of batteries with limited capacity for the supply of sensors. In our work, we propose routing algorithms to route information while controlling energy consumption.

To achieve this goal, we have represented the sensor network as a grayscale image: light areas represent regions rich in energy, whereas dark areas represent regions with low energy capacity. Filters used in image processing are then applied to the image representing the energy. Thus, we propose convolution filters like Sobel or mean filter in our routing algorithms and we construct an energy matrix for each sensor. This matrix will be used with the convolution to find the best path. The proposed algorithms are verified by simulations performed with the network simulator OMNeT++.

**Keywords:** wireless sensor network, routing, energy, clustering, image processing, matrix convolution, kernel filter.